D

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; Photogrammetry; Scene analysis

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

S. Shekhar et al. (eds.), *Encyclopedia of GIS*,
DOI 10.1007/978-3-319-17885-1

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 (Hartley and Zisserman 2000). 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 (Schwarz et al. 1993). 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.

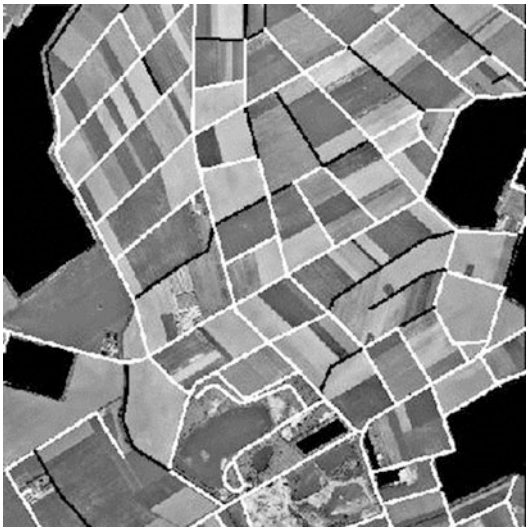
Orthoprojection, the projection of a central perspective image to a reference surface, mostly a horizontal plane, is a standard task in photogrammetry: Recently, automatic solutions for so-called true orthos have become available. True orthos are orthophotos for which a high quality DSM has been used 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 (Rosenfeld 1982). 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.



Data Acquisition, Automation, Fig. 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

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 not recognized automatically and need to be checked by a human operator (Gerke et al. 2004). 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 (Baltsavias 2004):

- 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 singleobject 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 selfdiagnosis;
- 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 (Gülch and Müller 2001) 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

References

Baltsavias E (2004) Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS J Photogramm Remote Sens 58:129–151

Gerke M, Butenuth M, Heipke C, Willrich F (2004) Graphsupported verification of road databases. ISPRS J Photogramm Remote Sens 58:152–165

Gülch E, Müller H (2001) New application of semiautomatic building acquisition. In: Baltsavias E, Grün A, van Gool L (eds) Automatic extraction of man-made objects from aerial and space images (III). Balkema, Lisse, pp 103–114

Hartley R, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge University Press, Cambridge

Rosenfeld A (1982) Computer image analysis: an emerging technology in the service of society. Computer science technical report TR-1177, MCS-79-23422, University of Maryland
Schwarz K-P, Chapman ME, Cannon E, Gong P (1993) An integrated INS/GPS approach to the georeferencing of remotely sensed data. Photogramm Eng Remote Sens 59:1667–1674

Data Analysis, Spatial

Michael F. Goodchild
Department of Geography, University of California, Santa Barbara, CA, USA

Synonyms

Anamolies; GeoDa; Geospatial Analysis; Geographically Weighted Regression; Geographical Analysis; Geostatistics; Patterns; Point Patterns; Spatial Analysis; Spatial Interaction

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 $<\mathbf{x},\mathbf{z}>$ where $\mathbf{x}$ denotes a point in space-time and $\mathbf{z}$ 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 et al. 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.sourceforge.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

References

Bailey TC, Gatrell AC (1995) Interactive spatial data analysis. Longman Scientific and Technical, Harlow

Berry BJL, Marble DF (eds) Spatial analysis: a reader in statistical geography. Prentice Hall, Englewood Cliffs (1968)

Bunge W (1966) Theoretical geography. Gleerup, Lund

Cliff AD, Ord JK (1973) Spatial autocorrelation. Pion, London

Fotheringham AS, O'Kelly ME (1989) Spatial interaction models: formulations and applications. Kluwer Academic, Boston

Fotheringham AS, Brunsdon C, Charlton M (2002) Geographically weighted regression: the analysis of spatially varying relationships. Wiley, Hoboken

Ghosh A, Rushton G (eds) Spatial analysis and location allocation models. Van Nostrand Reinhold, New York (1987)

Goodchild MF, Janelle DG (2004) Spatially integrated social science. Oxford University Press, New York

Haggett P (1966) Locational analysis in human geography. St Martin's Press, New York

Haggett P, Chorley RJ (1969) Network analysis in geography. Edward Arnold, London

Johnson S (2006) The ghost map: the story of London's most terrifying epidemic-and how it changed science, cities, and the modern World. Riverhead, New York

King LJ (1969) Statistical analysis in geography. Prentice Hall, Englewood Cliffs

Longley PA, Goodchild MF, Maguire DJ, Rhind DW (2005) Geographic information systems and science, 2nd edn. Wiley, New York

O'Sullivan D, Unwin DJ (2003) Geographic information analysis. Wiley, Hoboken

Taylor PJ (1977) Quantitative methods in geography: an introduction to spatial analysis. Houghton Mifflin, Boston

Tomlin CD (1990) Geographic information systems and cartographic modeling. Prentice Hall, Englewood Cliffs

Data Approximation

▸ Constraint Databases and Data Interpolation

Data Collection, Reliable Real-Time

Vana Kalogeraki[1] and Amir Soheili[2]
[1]Department of Computer Science and Engineering, University of CA, Riverside, CA, USA
[2]ESRI, Redlands, CA, USA

Synonyms

Energy-Aware; Energy Optimization; Monitoring; Peer-Tree (spatial Index); Reliable Real–Time Data Collection; Routing; Spatial Index; Spatial Queries; SPIX; Surveillance

Definition

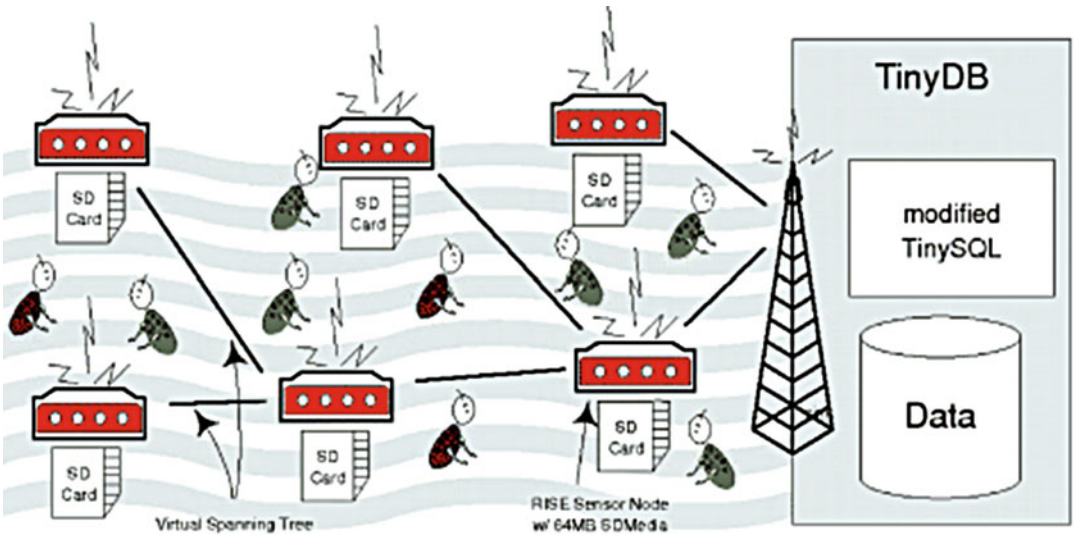
Recent technological advances in wireless technologies and microelectronics have led to the emergence of wireless sensor networks, consisting of large numbers of small, low-power, inexpensive wireless sensor devices that are embedded in the physical world and are able to monitor it in a non-intrusive 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



Data Collection, Reliable Real-Time, Fig. 1 Example of spatial data collection

and optimizing the distributed spatial index to efficiently and reliably process the spatial queries while reducing energy consumption.

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 (Beckman et al. 1990; Gutman 1984; Sellis et al. 1987) are used to execute a spatial query. R-Tree (Gutman 1984) 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 (Sellis et al. 1987) and R*Tree (Beckman et al. 1990) 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 (Tao and Papadias 2003). 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. Demirbas and Ferhatosmanoglu (2003) 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 Demirbas and Ferhatosmanoglu (2003), 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 (Heinzelman and Chandrakassan 2000) proposes an energy adaptive efficient clustering to distribute energy load evenly among the sensors in the network. Hu et al. (2005) 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 (Intanagonwiwat et al. 2003) forwards the request based on the sender's interest such as location. Geographic based routing (Ko and Vaidya 2000) 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 (Bonnet et al. 2001) and TinyDB (Madden et al. 2003) 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. (2003) 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 communication 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 (Bonnet et al. 2001; Madden et al. 2003), 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 1 (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.

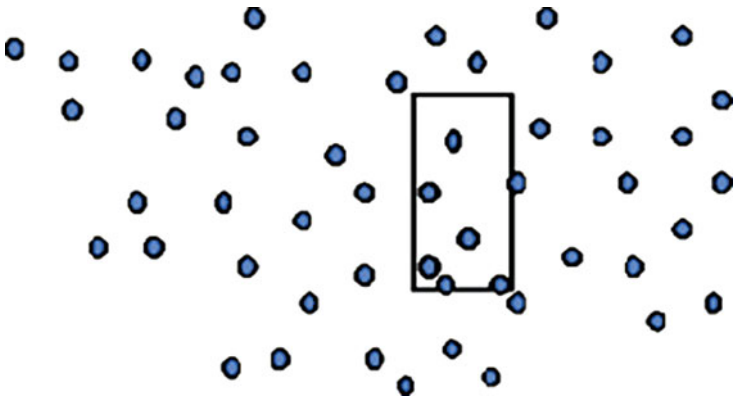
Figure 2 gives an example of a spatial query on a sensor network. Although the main interest is in techniques to efficiently evaluate aggregates such as SUM, MAX or MIN, the techniques described are general techniques tare general and can be used for other hierarchically decomposable functions. Alternative 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 (Beckman et al. 1990; Gutman 1984) 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 (Soheili et al. 2005) is described, a distributed index structure that allows each sensor to efficiently determine if it needs to participate



Data Collection, Reliable Real-Time, Fig. 2 A range query example: the use is interested only in the values of the sensors falling in the query *rectangle*

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.

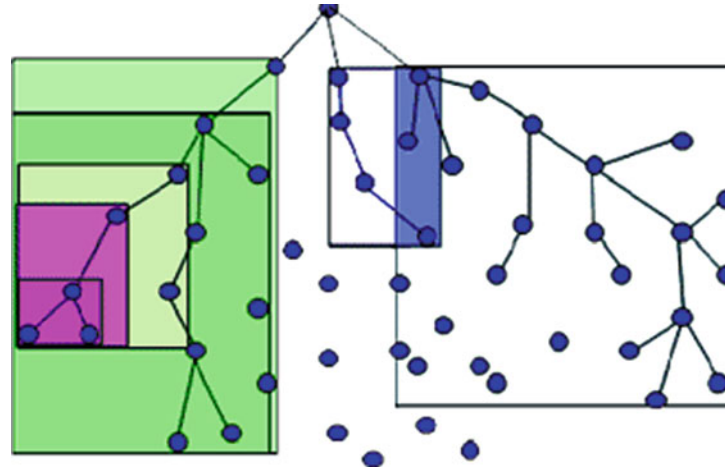
SPIX imposes a hierarchical structure in the network. It makes the assumption that spatial queries will always be disseminated into the sensor network from a base station. The base station is responsible for preparing the query, submitting it into the sensor network and getting the result back. The spatial 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.

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



Data Collection, Reliable Real-Time, Fig. 3 A SPIX index structure example

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.

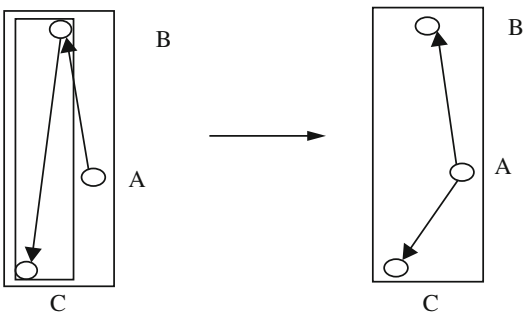
In order to avoid disconnections or cycles in the network, the base station submits its location to the sensors in the advertisement phase. Each sensor reviews its candidate parents before starting the parent selection phase and if there is at least one candidate closer from this sensor to the base station, 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 (Beckman et al. 1990).

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 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.

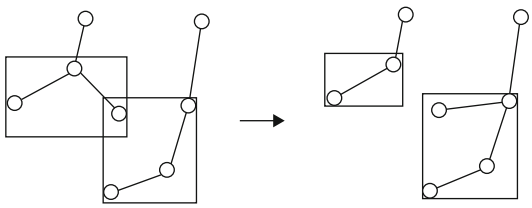


Data Collection, Reliable Real-Time, Fig. 4 Effect of eliminating thin *rectangles*

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. Figure 5 shows how the energy optimization phase reduces the MBR size of the higher level sensors.



Data Collection, Reliable Real-Time, Fig. 5 Effect of energy optimization phase

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.
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

References

Beckman N, Kriegel H, Schneider R, Seeger B (1990) The R*tree: an efficient and robust access method for point and rectangles. ACM SIGMOD Rec 19(2):322–331

Bonnet P, Gehrke J, Seshardi P (2001) Toward sensor database systems. Paper presented at the 2nd international conference on mobile data management (MDM 2001), Hong Kong, pp 3–14

Demirbas M, Ferhatosmanoglu H (2003) Peer-to-peer spatial queries in sensor networks. Paper presented at the 3rd IEEE international conference on peer-to-peer computing (P2P'03), Linkoping, pp 32–39

Gutman A (1984) RTree – a dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM SIGMOD international conference on management of data (SIGMOD 1984), Boston

Heinzelman WR, Chandrakassan A (2000) EnergyEfficient communication protocol for wireless microsensor networks. Paper presented at the 33rd Hawaii international conference of system sciences, Maui

Hu H, Xu J, Wang WS, Zheng B, Lee DL, Lee W (2005) Proactive caching for spatial queries in mobile environments. In: Proceedings of the 21st international conference on data engineering (ICDE 2005), Tokyo

Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. IEEE/ACM Trans Netw 11(1):2–16

Ko Y-B, Vaidya NH (2000) Location-aided routing (LAR) in mobile ad hoc networks. Wirel Netw 6(4):307–321
Madden S, Franklin MJ, Hellerstein JM, Hong W (2003) The design of an acquisitional query processor for sensor networks. In: Proceedings of the 2003 ACM SIGMOD international conference on management of data (SIGMOD), San Diego, pp 491–502
Sellis TK, Roussopoulos N, Faloutsos C (1987) The R+tree: a dynamic index for multidimensional objects. In: Proceedings of 13th international conference on very large data bases (VLDB 1987), Brighton
Soheili A, Kalogeraki V, Gunopulos D (2005) Spatial queries in sensor networks. Paper presented at the 13th international symposium on advances in geographic information systems (GIS 2005), Bremen
Tao Y, Papadias D (2003) Spatial queries in dynamic environments. ACM Trans Database Syst 28(2): 101–139

Data Compression

▸ Image Compression

Data Compression for Network GIS

Haijun Zhu[1] and Chaowei (Phil) Yang[2]
[1]Joint Center for Intelligent Spatial Computing (CISC), College of Science, George Mason University, Fairfax, VA, USA
[2]Joint Center for Intelligent Spatial Computing, College of Sciences, George Mason University, Fairfax, VA, USA

Synonyms

Information theory; Non-raster data compression; 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 (Egger et al. 1999; Jayant and Noll 1984), and 10 ∼ 20 % for vector (Shekhar et al. 2002) and textual data (Bell et al. 1989). Such compression ratios result in significant performance improvement.

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
…	…

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 (Information Technology 1993) 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 (Shapiro 1993; Said and Pearlman 1996;

Taubman 2000). 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.

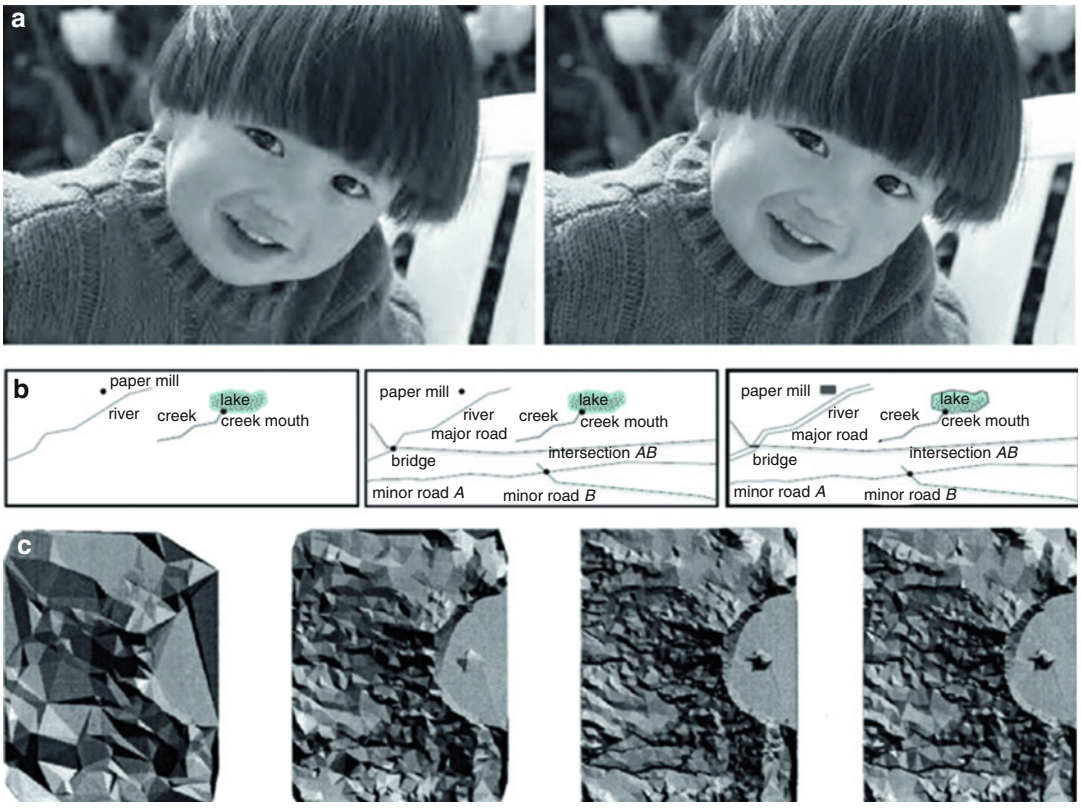
PT is based on multiresolution data decomposition (Shapiro 1993; Said and Pearlman 1996; Taubman 2000). For raster data, many effective algorithms can be used to generate such decomposition. For non-raster data, it is quite hard to construct progressive bitstreams effectively because these data are not defined in a regular spatial grid and commonly used multi-resolution decomposition algorithms (e.g., wavelet) are difficult to apply (Bertolotto and Egenhofer 2001, 1999; Buttenfield 2002). Therefore, other methods (e.g., cartographical-principle based decimation, Fig. 1b, c may be adopted.

Historical Background

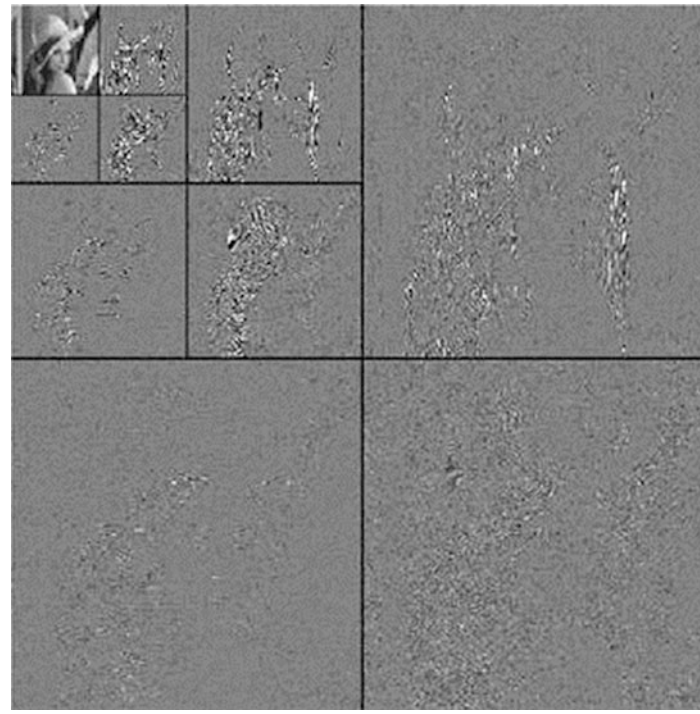
Data compression of network GIS is similar to other data compression algorithms on distributed computing platforms. Image compression algorithms such as JPEG had been applied since the first Web-based GIS emerged in 1993 (Plewe 1997). However, the compression of vector data was introduced much later, such as the Douglas-Peuker algorithm (1973) and the work done in 2001 by Bertolotto and Egenhofer (2001).

Scientific Fundamentals

Data compression originates from information theory (Shannon 1948), 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 sig-



Data Compression for Network GIS, Fig. 1 Progressive transmission of different types of data in networked GIS. **(a)** Image. **(b)** Cartographic principle based progressive vector. **(c)** TIN



Data Compression for Network GIS, Fig. 2
Wavelet decomposition of image

nificant theoretical results in information theory is the so-called source coding theorem (Shannon 1948), 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 (Jayant and Noll 1984; Cover and Thomas 1991) 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 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 (Shapiro 1993; Said and Pearlman 1996) or optimally add more truncation points (Taubman 2000).

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 (Yang et al. 2005) to deal with large volumes of geospatial raster data,
- Integration of other types of datasets (e.g., vector and 3-D data).

WebGIS:

- TerraServer (Barclay et al. 1999) 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 the response time will be too long. Yang, et al. (2005) developed a method to manage a permanent pyramid so that performance can be improved.
- Google Earth (2006) 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 Heckbert and Garland (1997). 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 (Shekhar et al. 2002; Lu and Dunham 1991) 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 (Bertolotto and Egenhofer 2001, 1999; Buttenfield 2002). 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 (Hoppe 1996) 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 (Park et al. 2001).

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 (OGC 2006). 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. (2006) and the OGC (Open GIS Consortium) also provides promising results (OGC 2005).

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

References

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

Bell T, Witten IH, Cleary JG (1989) Modeling for text compression. ACM Comput Surv 21:557–591

Bertolotto M, Egenhofer MJ (1999) Progressive vector transmission. In: Proceedings of the 7th international symposium on advances in GIS, Kansas City

Bertolotto M, Egenhofer MJ (2001) Progressive transmission of vector map data over the World Wide Web. GeoInformatica 5(4):345–373

Buttenfield BP (2002) Transmitting vector geospatial data across the internet. In: Proceedings of the second international conference on geographic information science, Boulder, 25–28 Sept 2002. Lecture notes in computer science, vol 2478. Springer, Berlin, pp 51–64

Cover TM, Thomas JA (1991) Elements of information theory. Wiley, Chichester

CubeWerx Inc (2006) http://www.cubewrex.com/. Accessed 20 Aug 2006

Douglas D, Peucker T (1973) An algorithm for the reduction of the number of points required to represent a digitized line or its character. Can Cartogr 10(2): 112–122

Egger O, Fleury P, Ebrahimi T, Kunt M (1999) Highperformance compression of visual information-a tutorial review-part I: still pictures. Proc IEEE 87:976–1011

Google Earth (2006) http://earth.google.com. Accessed 20 Dec 2006

Heckbert PS, Garland M (1997) Survey of polygonal surface simplification algorithms. Carnegie Mellon University, Pittsburg

Hoppe H (1996) Progressive meshes. In: Proceedings SIGGRAPH'96, New Orleans, 4–9 Aug 1996, pp 99–108

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)

Jayant NS, Noll P (1984) Digital coding of waveforms. Prentice-Hall, Englewood Cliffs

Lu CC, Dunham JG (1991) Highly efficient coding schemes for contour lines based on chain code representations. IEEE Trans Commun 39(10):1511–1514

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 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

Park D, Cho H, Kim Y (2001) A TIN compression method using Delaunay triangulation. Int J Geogr Inf Sci 15:255–269

Plewe B (1997) GIS online. Information retrieval, mapping and the Internet. OnWord Press, Albany

Said A, Pearlman WA (1996) A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans Circuits Syst Video Technol 6:243–250

Shapiro JM (1993) Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans Signal Process 41:3445–3462

Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27: 379–423; 623–656

Shekhar S, Huang Y, Djugash J, Zhou C (2002) Vector map compression: a clustering approach. In: Proceedings of the 10th ACM international symposium on advances in geographic information systems, McLean, 8–9 Nov 2002

Taubman D (2000) High performance scalable image compression with EBCOT. IEEE Trans Image Process 9:1158–1170

Yang C, Wong D, Yang R, Kafatos M, Li Q (2005) Performance improving techniques in WebGIS. Int J Geogr Inf Sci 19(3):319–342

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

Digital divide; Map distribution

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

Aanestad M, Monteiro E, Nielsen P (2007) Information infrastructures and public goods: analytical and practical implications for SDI. Inf Technol Dev 13:7–25

Câmara G, Fonseca F, Monteiro AM, Onsrud H (2006) Networks of innovation and the establishment of a spatial data infrastructure in Brazil. Inf Technol Dev 12:255–272

Davis CA, Fonseca F (2006) Considerations from the development of a local spatial data infrastructure. Inf Technol Dev 12:273–290

Georgiadou Y, Bernard L, Sahay S (2006) Implementation of spatial data infrastructures in transitional economies. Inf Technol Dev 12:247–253

Lance K, Bassolé A (2006) SDI and national information and communication infrastructure (NICI) integration in Africa. Inf Technol Dev 12:333–338
Onsrud H, Poore B, Rugg R, Taupier R, Wiggins L (2004) The future of the spatial information infrastructure. In: McMaster RB, Usery EL (eds) A research Agenda for geographic information science. CRC, Boca Raton, pp 225–255
Satish KP (2006) Technological frames of stakeholders shaping the SDI implementation: a case study from India. Inf Technol Dev 12:311–331
Sorensen M, Sayegh F (2007) The initiation, growth, sustainment of SDI in the Middle East-notes from the trenches. Inf Technol Dev 13:95–100

Data Integration

▸ Conflation of Features

Data Mining

▸ Urban Data Science: An Introduction

Data Mining of Constraint Databases

Peter Revesz
University of Nebraska-Lincoln, Lincoln, NE, USA

Definition

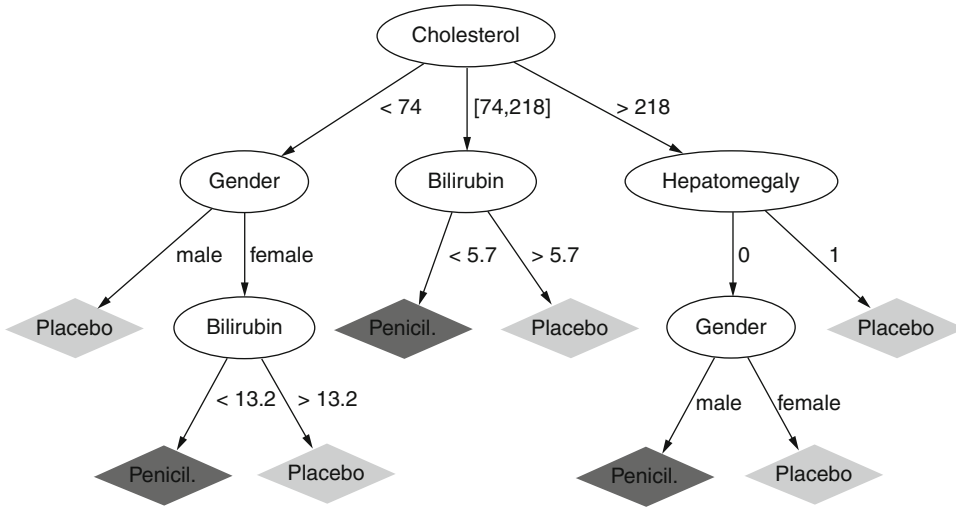
Many data mining algorithms can be extended and applied to constraint databases (Lakshmanan et al. 2003; Mohan and Revesz 2014; Revesz 2010; Turmeaux and Vrain 1999). Constraint databases are used in data mining because data mining algorithms such as decision trees (Quinlan 1986) and support vector machines (Vapnik 1995) generate classifications that can be naturally represented by constraint databases (Geist 2002; Johnson et al. 2000; Lakshmanan et al. 2003; Turmeaux and Vrain 1999). The constraint database representation enables querying the classification data and to further enhance the data mining results.

Main Text

Constraint databases are convenient in representing and further querying the results of data mining classifications (Lakshmanan et al. 2003; Mohan and Revesz 2014; Revesz 2010; Turmeaux and Vrain 1999). Fig. 1 shows an example from Chapter 17 of (Revesz 2010) is a decision tree that classifies *primary biliary cirrhosis* patients according to the drug that is assigned to patients in a hospital. In that decision tree, the feature space is $X_1 = \{B, C, G, H\}$, and the set of labels is $Y_1 = \{\text{Penicillamine, Placebo}\}$, where B is the serum bilirubin measured in mg/dl; C is the serum cholesterol in mg/dl; G, meaning gender, is 0 for female and 1 for male; and H, meaning hepatomegaly, is 1 if the liver is enlarged and 0 otherwise. The label Penicillamine means that the patient was prescribed penicillamine, and the label Placebo means that the patient was given only a sugar pill. The decision tree can be conveniently represented in a constraint database table as follows:

				Drug
B	**C**	**G**	**H**	**D**
b	c	g	h	d	c < 74, g = 1, d = "Placebo"
b	c	g	h	d	c < 74, g = 0, b < 13.2, d = "Penicil"
b	c	g	h	d	c < 74, g = 0, b > 13.2, d = "Placebo"
b	c	g	h	d	c >= 74, c <= 218, b < 5.7, d = "Penici"
b	c	g	h	d	c >= 74, c <= 218, b > 5.7, d = "Placebo"
b	c	g	h	d	c > 218, h = 0, g = 1, d = "Penicil"
b	c	g	h	d	c > 218, h = 0, g = 0, d = "Placebo"
b	c	g	h	d	c > 218, h = 1, d = "Placebo"

For the same set of patients, several different decision tree-based data mining results could be similarly represented using constraint database tables. Constraint database systems then allow the querying of these different representations. For example, suppose that another decision tree classifies the patients using the feature space $X_2 = \{C, G, H, T\}$, where C, G, and H are as before and T is triglycerides level, and the labels $Y_2 = \{\text{Alive, Dead, Transplanted}\}$. If the



Data Mining of Constraint Databases, Fig. 1 A decision tree for PCB patients, Fig. 1.

result of this decision tree is also represented by a constraint database table with scheme Status(C, G, H, T, S), then the constraint database tables representing the two decision trees can be combined by a simple SQL query to result in a *reclassification* (Revesz and Triplet 2010, 2011), which is represented by the constraint database table PBC patient(B, C, G, H, T, D, S). The PBC patient relation is then easily queried in novel ways that are not possible for the individual results and allow the discovery of new relationships between drug prescribed and patient health outcomes (Revesz and Triplet 2010).

Data mining algorithms, such as decision trees (Quinlan 1986) and support vector machines (Vapnik 1995), can be extended to cases when the constraint database represents temporal data (Revesz 2014; Revesz and Triplet 2011) and spatiotemporal data (Mohan and Revesz 2014). An example of data mining with temporal data occurs in mining the history of citations to predict the citation curve of individual researchers (Revesz 2014), and an example of spatiotemporal data mining is the generation of a set of rules for the optimal control of a set of dams on a river reservoir system (Mohan and Revesz 2014).

Cross-References

- Constraint Databases, Spatial
- Constraint Databases and Data Interpolation
- Constraint Databases and Moving Objects
- Linear Versus Polynomial Constraint Databases

References

Geist I (2002) A framework for data mining and KDD. In: Haddad H, Papadopoulos G (eds) Proceedings of the ACM symposium on applied computing. ACM, New York, pp 508–13

Johnson T, Lakshmanan LVS, Ng RT (2000) The 3W model and algebra for unified data mining. In: Proceedings of the IEEE international conference on very large databases. Morgan Kaufmann, pp 21–32

Lakshmanan LVS, Leung CKS, Ng RT (2003) Efficient dynamic mining of constrained frequent sets. ACM Trans Database Syst. Cairo, Egypt, 28(4):337–389

Mohan A, Revesz PZ (2014) Applications of spatiotemporal data mining to North Platte River reservoirs. In: Proceedings of the 18th international database engineering and applications symposium, Porto. ACM, pp 306–309

Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106

Revesz PZ (2010) Introduction to databases: from biological to spatio-temporal. Springer, New York

Revesz PZ (2014) A method for predicting the citations to the scientic publications of individual researchers. In: Proceedings of the 18th international database en-

gineering and applications symposium, Porto. ACM, pp 9–18
Revesz PZ, Triplet T (2010) Classification integration and reclassification using constraint databases. Artif Intell Med 42(3):79–91
Revesz PZ, Triplet T (2011) Temporal data classication using linear classiers. Inf Syst 36(1):30–41
Turmeaux T, Vrain C (1999) Learning in constraint databases. Lecture Notes in Computer Science, vol 1721. Springer, Berlin/New York, pp 196–207
Vapnik V (1995) The nature of statistical learning theory. Springer, New York

Recommended Reading

Gomez-Lopez MT, Ceballos R, Gasca RM, Del Valle C (2009) Developing a labelled object-relational constraint database architecture for the projection operator. Data Knowl Eng 68(1):146–172

Data Mining Techniques for the Characterization of Dynamic Regions in Spatiotemporal Data

Michael P. McGuire
Department of Computer and Information Sciences, Towson University, Towson, MD, USA

Synonyms

Spatiotemporal Change Detection; Spatiotemporal Data Mining; Spatiotemporal Dynamics;

Definition

Spatiotemporal sensor data is prevalent in many domains and, at its most basic level, consists of a sensor location defined by coordinates in two-dimensional or three-dimensional space and an attribute or set of attributes being measured at that location. In the context of this entry, sensors are typically represented in the form of a point location where the objective is to measure a spatial process that is moving over time. For example, a precipitation gage or a pixel in a satellite image could be modeled as a stationary sensor. Moving sensors such as depth sensors on boats or a drifting temperature probe in the ocean also attempt to measure a moving phenomenon except the sensors are also moving in space. The resulting dataset includes a set of spatial coordinates representing either a sensor or the center of a grid cell, a time stamp, and the attributes being measured at that location. Dynamic regions are then the locations and time periods that are experiencing constant change. For example, given a spatiotemporal field of climate variables, a dynamic spatiotemporal region for a particular variable is composed of the time periods and spatial locations that have a significant amount of change. For example, in this dataset, the path of a tropical cyclone would create a dynamic spatiotemporal region over a given time period. This entry gives an overview of data mining techniques for finding dynamic regions in spatiotemporal sensor data and gives a number of applied examples in Earth and environmental sciences.

Historical Background

Over the last few decades, advances in sensor networks and remote sensing platforms have resulted in a massive amount of spatiotemporal data. Most often the purpose of this data is to monitor natural phenomena related to Earth's environmental systems. These advances in automated measurement have lead to major discoveries uncovering spatial processes related climate science, hydrology, atmospheric science, and numerous other Earth science disciplines. Furthermore, the use of remote sensing platforms has resulted in the ability to use spatiotemporal datasets to precisely measure anthropogenic impacts including urbanization, deforestation, and land conversion.

As a motivational example for this entry, in the field of climatology, large-scale gridded models are typically integrated with sensor data resulting in a massive spatiotemporal dataset comprised of a number of variables measuring climate conditions such as air temperature, geopotential height,

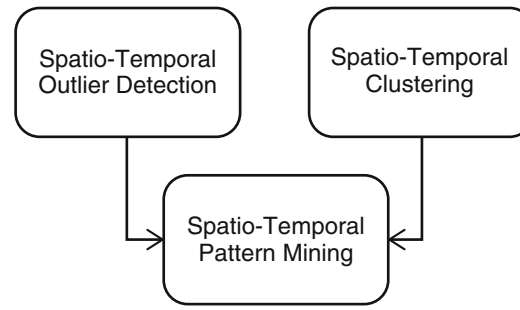
relative humidity, specific humidity, Omega (vertical velocity), U-wind, and V-wind. The spatiotemporal pattern of any one of these variables is extremely complex and therefore very difficult to characterize in a discrete manner. With this in mind, finding areas in space and time that are most dynamic can lead to the discovery of interesting climate patterns and events. For example, finding the locations of dynamic regions in space and time might allow scientists to better characterize and, ultimately, predict global climate phenomena such as El Niño, La Niña, regional drought conditions, monsoon seasons, tropical cyclone development, and climate change in general.

This motivational example requires the analysis of a very large multidimensional spatiotemporal dataset. Historically, the analysis of spatiotemporal patterns is rooted in the field of physics where partial differential equations and spatiotemporal covariance models are used to model physical processes and spatiotemporal statistics where statistical models are created to characterize spatiotemporal point processes (Cressie and Wikle 2011). Exploratory data analysis approaches have traditionally used empirical orthogonal functions (EOF) and singular value decomposition (SVD) to find spatiotemporal patterns in these types of data (Von Storch and Zwiers 2002). Limitations of physical and statistical models include the masking of nondominant patterns which might be of interest to a scientist. Also, using these approaches to find local spatiotemporal regions in the data where interesting patterns emerge requires a great deal of prior information about the process under investigation. More recently, the analysis of spatial and spatiotemporal data has gained popularity in the data mining community. From a data mining perspective, a number of techniques can be used to find and analyze dynamic regions in spatiotemporal sensor data. First, spatial and spatiotemporal outlier detection and spatiotemporal clustering can be used to find dynamic spatiotemporal regions. Then spatiotemporal pattern mining techniques can be used to analyze how the dynamic regions change over time. A number of key applications of these techniques are also described in this entry including (1) the analysis of precipitation data, (2) the analysis of global climate model results, (3) pattern mining in sea surface temperature data, and (4) finding submerged debris in crowdsourced bathymetry data.

Scientific Fundamentals

This section reviews a number of data mining techniques that can be used in combination as a data mining workflow to find and analyze dynamic regions in spatiotemporal sensor data. An overview of this workflow is shown in Fig. 1. The first two sets of approaches, spatial and spatiotemporal outlier detection and spatiotemporal clustering, can be used to find the dynamic spatiotemporal regions where they could be expressed as either outlying regions or regions that are clustered. Once the regions are found, then spatiotemporal pattern mining techniques can be used to analyze the regions as they evolve over time.

The remainder of this section provides a review of data mining techniques for spatial and spatiotemporal outlier detection, spatiotemporal clustering, and spatiotemporal pattern mining. Throughout this section, a motivational of a tropical cyclone moving through a spatiotemporal field is used to illustrate how the approaches



Data Mining Techniques for the Characterization of Dynamic Regions in Spatiotemporal Data, Fig. 1 Overview of approaches to find dynamic spatiotemporal regions

could be used to find dynamic spatiotemporal regions.

Spatial and Spatiotemporal Outlier Detection

Spatial and spatiotemporal outlier detection techniques can be used to find dynamic regions in spatiotemporal sensor datasets. In this case, the dynamic spatiotemporal regions can be expressed as a combination of the spatial locations and time periods where outliers occur and the spatial region that contains the outliers over time. The amount of change in the location of the spatial outliers over time increases the magnitude of the dynamic spatiotemporal region. For example, consider the path and areal extent of a tropical cyclone as a dynamic spatiotemporal region. In a gridded climate dataset, this event would produce spatial outliers for variables such as air pressure, precipitation, and wind speed. The locations of these outliers over any duration of time will result in a dynamic spatiotemporal region.

Spatial outlier detection techniques are related to distance-based multivariate outlier detection techniques where the spatial dimensions are grouped along with the other dimensions in the dataset (Knorr et al. 2000). Spatial outlier detection methods, on the other hand, typically consist of an initial step to determine the neighborhood of a spatial object. Then once the neighborhood is created, statistical outlier detection methods are then applied to determine outliers within the neighborhood based on a nonspatial measurement value. In Shekhar et al. (2003), a global aggregate function is applied to a single nonspatial attribute. In this scheme, a spatial outlier is defined as a spatially referenced point whose nonspatial attribute values are different from those of other spatially referenced points in its spatial neighborhood. Another approach uses an iterative scheme where for each point and its k-nearest neighbors μ and σ, points are added to the neighborhood if their values are below a certain threshold; spatial points that are beyond the threshold are considered to be spatial outliers (Changtien et al. 2003). In Kou et al. (2007), a graph is constructed based on the k-nearest neighbors for spatial points. Differences between nonspatial attributes are then assigned as edge weights where a graph cut is then performed to identify isolated points that are dissimilar from their neighbors. There have also been a number of studies which focus on finding local outliers in spatial data (Breunig et al. 1999). Local outliers are assigned a degree of outlierness according to their spatial position and attribute values. In Sun and Chawla (2004), a method is presented that takes into account spatial autocorrelation and heteroscedasticity to reduce the effects of outliers on their neighbors. This approach gives a higher weight to outliers that are in spatially stable areas over outliers that are in unstable areas.

Outlier detection methods have also been developed to find spatiotemporal outliers where outlying values are found in space and time. In Cheng and Li (2006), a spatiotemporal outlier is defined to be a "spatiotemporal object whose thematic attribute values are significantly different from those of other spatially and temporally referenced objects in its spatiotemporal neighborhood." A number of spatiotemporal outlier detection techniques use spatial clustering as a first step in the process. For example, in Birant and Kut (2006) density-based clustering is used to first find clusters and outliers in the data. Then spatial outliers are identified with respect to their spatial neighbors. In this approach, temporal outliers are then identified with relation to a temporal neighborhood. Then in this scheme, an outlier is a point that remains anomalous based on checking its spatial and temporal neighborhood. In Das and Parthasarathy (2009), a similar scheme is used where distance-based clustering is used to identify outliers and spatial and temporal neighborhood-based outlier detection is used to find spatiotemporal anomalies in global climate data.

Scan statistics are popular tools for the spatial and spatiotemporal analysis of epidemiological data and are another set of approaches that can be used to find spatial and spatiotemporal outliers (Kulldorff 1997; Agarwal et al. 2006; Janeja and Atluri 2008). Scan statistics use a scan window which is generally a spatial extent. Then a statistical measure is calculated based on the locations falling within the window, and the

window is then progressively moved to cover the entire dataset. The resulting outlying windows represent regions in the data that are statistically different. Scan statistics can then be used to find outlying regions in the data. In Wu et al. (2010), scan statistics are used to discover top-k outliers in gridded precipitation datasets. This is a spatial scan statistic approach where sequences of outliers over time are stored in a tree and a recursive algorithm is used to extract all possible outlier sequences.

Spatiotemporal Clustering

Similar to outlier detection, when mining spatiotemporal data, the places in space and time where the data is clustered often can represent dynamic spatiotemporal regions. The main objective of clustering is to find groupings in a dataset. In particular, spatiotemporal clustering finds spatial locations that are grouped for a specific period of time. To use spatiotemporal clustering to find dynamic regions, the approach needs to find places that are clustered in space and time but analyze the change in the clustering results over time. Considering the motivational example of a tropical cyclone, these clustered areas could be expressed as the boundaries of the storm as it moves through space. Areas experiencing elevated pressure, precipitation, and wind speed, for example, would be clustered around the center of the storm and therefore delineate the boundaries of the area affected by the storm. As the storm moves, these clustered areas change significantly and the change in this clustering over time would be considered to be dynamic regions where the magnitude of the dynamic region would be characterized by the amount of change in the clustering.

One of the first methods proposed for clustering spatiotemporal data is ST-DBSCAN (Birant and Kut 2007). This approach extends the DBSCAN algorithm to include nonspatial and temporal dimensions. Another study presents an approach to improve spatiotemporal clustering by extending the distance measure traditionally used in most clustering algorithms to be a function of the position history of the spatiotemporal objects in the dataset (Rosswog 2008). In Sap (2005), a weighted kernel k-means algorithm is proposed to account for problems with nonlinear separability in spatiotemporal data. In particular, a penalty term containing spatial neighborhood information is used, and time is included as an additional variable to the clustering. In Lin et al. (2009), a tight clustering algorithm is presented where the clustering is based on a measure of process similarity. The process similarity uses a combination of principal components analysis to determine a single indicator to account for multiple attributes and Pearson product-moment correlation to measure the correlation between the time series for two spatial points. Clustering is then performed on the resulting similarity graph. While this approach does account for spatiotemporal aspects of the data, it provides a global clustering for an entire time series and, therefore, does not uncover changing patterns over time. In Günnemann et al. (2012), an approach to finding traces of subspace clusters in temporal data is presented. This approach uses a distance function based on subspace similarity to trace evolution in temporal clusters. Another approach (Steinhaeuser et al. 2012) models spatiotemporal dependence in climate data using complex networks.

Spatiotemporal Pattern Mining

Another set of approaches that is relevant to this entry are approaches that are focused on mining spatiotemporal patterns. These approaches find patterns that are specific to spatiotemporal data by extracting the spatiotemporal pattern represented in the data as it evolves over time. Take the example of a tropical cyclone that has been used throughout this entry. This assumes that some outlier detection or clustering approach is used to identify a set of dynamic spatiotemporal regions in the data. Spatiotemporal pattern mining can then be used to analyze the movement of the dynamic spatiotemporal regions. This includes spatial attributes of the evolution of the dynamic spatiotemporal regions such as change in extent, trajectory, merging, and splitting.

Directly related to spatiotemporal clustering are approaches that analyze cluster transitions. These approaches take the evolution of cluster

formation over time and provide a framework to analyze such transitions as single clusters splitting into multiple clusters, many clusters merging to form a single cluster, cluster persistence, and cluster disappearance. The MONIC framework (Spiliopoulou et al. 2006) introduces the concept of cluster transitions. In this research, cluster transitions are considered in one of two categories – internal transitions which include transitions between a cluster in the general space such as relationships with other clusters including cluster survival, cluster splits, cluster absorption, cluster disappearance, and cluster emergence and external transitions such as changes in size, compactness, and location transition. In Oliveira and Gama (2010) cluster transitions are also studied where clusters are modeled using a graph structure where the nodes in the graph represent cluster centers and the edges connecting cluster centers are weighted based on conditional probabilities that each node is a member of the same cluster. In this approach, cluster transitions are characterized by mining the changes in the graph structure. In Chan et al. (2008), clustering for spatial-temporal analysis of graphs (cSTAG) is used to mine spatiotemporal patterns in emerging graphs. This method begins by partitioning the time series into a number of overlapping fixed windows. For each window, regions of correlated spatiotemporal change are discovered. The graph changes are represented as waveforms. The cSTAG algorithm then clusters these waveforms to find regions of correlated change. In McGuire et al. (2011), a measure of spatial change over time based on a graph cut algorithm is used to find interesting areas in large spatiotemporal datasets.

A general approach to model spatiotemporal features and describe their evolution over time is presented in Yang et al. (2005). Dynamic spatiotemporal regions can be used to detect events in spatiotemporal datasets. In Huang et al. (2008), a spatial sequence index and a temporal-slicing-based algorithm are used to find sequential spatiotemporal events. The approach uses a spatial sequence index and a temporal-slicing-based algorithm to find sequential spatiotemporal events. In Worboys and Duckham (2006), combinatorial maps are used as a framework to model spatiotemporal change in geosensor networks where a number of transition rules are defined to represent evolving regions in the data.

Once the dynamic regions are identified, additional moving objects and trajectory data mining approaches can be used to find the movement patterns of the regions over time. For example, McGuire et al. (2014) focuses on the analysis of trajectories and extents of dynamic spatiotemporal regions. In this approach, local spatial autocorrelation is used to identify dynamic spatiotemporal regions in the form of globally and locally dynamic spatial locations and time periods in large sensor datasets by measuring the spatial dependence of a variable between neighboring spatial locations over time. A globally dynamic spatial location is a sensor that is most different from neighboring sensors in terms of measurements taken at the sensor across all time periods. A globally dynamic time period is the point in time where these differences are most pronounced across all sensors. Similarly a locally dynamic spatial location is most different from its neighbors within a single time period, and a locally dynamic time period is the location in time where these differences exist with respect to a specific spatial location. The trajectories and extents of moving dynamic spatiotemporal regions are then analyzed in order to be able to map a dynamic spatiotemporal phenomenon over time.

Key Applications

Finding dynamic regions in spatiotemporal sensor data can be used in a number of situations to enhance the understanding and prediction of natural phenomena. In this section, a number of key applications are discussed including precipitation data, global climate models, sea surface temperature, and crowdsourced bathymetry.

Precipitation Data

The analysis of precipitation data is a critical step when forecasting floods in a watershed. Given a spatiotemporal field of precipitation data such as that provided by NEXRAD radar

sensors (Lin and Mitchell 2005), the objective is to find the trajectory and extent of high-intensity precipitation cells. In this case, the dataset is a time series of gridded rasters. In this dataset, dynamic spatiotemporal regions could be found by using a spatiotemporal clustering algorithm. In Reljin et al. (2003) a self-organizing map (SOM) neural network is used to find spatiotemporal regions of precipitation data. A major benefit of this approach is the nonlinearity of neural networks as applied to precipitation data which is also not linear in nature. In McGuire et al. (2014) a graph-based clustering algorithm is used to find dynamic spatiotemporal regions where nodes are represented by the grid cell centroid and edges are formed between adjacent grid cells to form local neighborhoods. Then an edge cut could be performed based on a threshold resulting in groupings of similar grid cells. Then another threshold could be applied to this result to isolate locations with high precipitation in space and time. After this step, spatiotemporal pattern mining techniques could be used to analyze how the dynamic regions evolve over time. This would include the trajectory and extent of the high precipitation areas. This could then be extended to a real-time scenario where the trajectories and extents are then used to predict areas that may experience severe flooding events. The result would then be used in a flood warning system to predict heavy areas of precipitation over watersheds.

Global Climate Models

Over the last two decades, climate change has been a popular topic of both scientific and political discussion. Because of this, there is a wealth of data available from climate models. This results in a massive spatiotemporal dataset generated for a long time period. Given such a massive spatiotemporal dataset, it becomes difficult to characterize the spatiotemporal pattern in terms of finding the areas that are changing the most over time. Furthermore, methods to track this change are needed. The results from climate models are typically produced as a raster time series. With this in mind, spatiotemporal clustering and outlier detection approaches could be used to first find the natural groupings in the data. For example, given a raster time series of global air temperature, clustering would find regions with similar temperatures. Then, the clustering results could be analyzed over time to find the locations where the clustering changes the most. This would result in the detection of regions where the air temperature data is most dynamic. Then, trajectory mining would be used to analyze the movement and change in areal extent of the dynamic regions. The characterization of the dynamic spatiotemporal regions could then be used to predict known climate phenomena such as dangerously hot or cold weather patterns.

Sea Surface Temperature

The impact of sea surface temperature on weather around the globe is significant. Consider in recent years the effect of El Niño and La Niña events contributing to extreme precipitation and drought scenarios in various places in the world. Because of this, a better understanding of the dynamics of the spatiotemporal distribution of sea surface temperature is needed. Take, for example, the tropical atmosphere ocean (TAO) array of sea surface temperature sensors in the Pacific Ocean (NOAA 2000). Dynamic regions in this data were found and analyzed in McGuire et al. (2014) as was mentioned in the precipitation data example. Because the sensors in the TAO array are not distributed in a regular grid, a Delaunay triangulation was used to form neighborhoods for each sensor where the neighborhood was made up of directly adjacent sensors. The trajectories and extents of dynamic sea surface temperature regions in the Pacific Ocean were then analyzed. The results showed significant differences in the trajectories and locations of dynamic regions under normal, El Niño, and La Niña conditions. The results of this approach could be used to predict and determine the affects of El Niño and Laña events on global climate patterns.

Detection of Submerged Debris in Crowdsourced Bathymetry Data

Recently, sensors have been placed on commercial and recreational marine vessels to collect crowdsourced bathymetry data. This has enabled the near real-time assessment of the navigability

of channels in shallow waters such as that found in the Chesapeake Bay and US Intercoastal Waterway. One application of dynamic spatiotemporal regions in this data allows the identification of submerged debris which can result in damage to vessels and ultimately the loss of property or life. This example application is based on research by Sedaghat et al. (2013) where outlier detection methods were used to identify submerged debris in crowdsourced bathymetry data. In particular, local outlier detection is used to find outlying regions that could potentially indicate the presence of submerged debris, and then density-based clustering is used to find clusters of outliers over time. This indicates significant changes in the bathymetry data resulting in dynamic spatiotemporal regions. This approach could be extended to include the analysis of the spatiotemporal patterns of submerged debris. Having this information could result in finding and remediating the source of the submerged debris and therefore improving the navigability and safety of shallow waterways.

Future Directions

There are a number of promising areas of research for finding dynamic spatiotemporal regions. First, dynamic regions can be found in sensor data using both spatiotemporal clustering and spatiotemporal outlier detection techniques. The relationships between these two techniques have yet to be studied. For example, density-based clustering algorithms such as ST-DBSCAN find both clusters and outliers in the data. It is conceivable that both groups of outliers and clusters that change a great deal result in dynamic spatiotemporal regions in the data. It would be of interest to know the difference between dynamic regions found with these two approaches. Furthermore, it would also be interesting to see whether there exist differences in the trajectories and extents in the dynamic spatiotemporal regions created with these two approaches for a specific dataset.

Nearly all of the approaches above deal with two dimensions of space, time, and a single attribute. There are a number of interesting future research directions when discussing the dimensions of the data. The first is to mine dynamic spatiotemporal regions in three-dimensional space, and the second is to extend the approach to include multiple attributes or even high-dimensional data to find subspaces in the data. For example, given the global climate model dataset as discussed in the above application, one could find dynamic regions in the spatial dimensions of latitude, longitude, and at multiple atmospheric levels. This would lead to the analysis of three-dimensional trajectories and extents of dynamic regions. Dealing with high-dimensional data would also be a challenge where one must take into consideration the correlation structure in the data which would include correlations in space, time, and between attributes. Also, another potentially promising area of research would be to find dynamic spatiotemporal subspaces in the data where subsets of attributes might be correlated and their dynamic spatiotemporal regions would overlap in space and time.

Finally, applying these techniques on real-world datasets could prove to be another ripe area of research. As is shown in a number of the key applications above, this can provide interesting insight to Earth's dynamic systems. Spatiotemporal sensor data is increasing at astronomical rates, and therefore, there are many opportunities where these methods could be applied to find new and interesting patterns.

References

Agarwal D, McGregor A, Phillips J, Venkatasubramanian S, Zhu Z (2006) Spatial scan statistics: approximations and performance study. In: Conference on knowledge discovery in data: proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, Pennsylvania, Citeseer, vol 20, pp 24–33

Birant D, Kut A (2006) Spatio-temporal outlier detection in large databases. J Comput Inf Technol 14(4):291

Birant D, Kut A (2007) St-dbscan: an algorithm for clustering spatial–temporal data. Data Knowl Eng 60(1):208–221

Breunig MM, Kriegel HP, Ng RT, Sander J (1999) Optics-of: identifying local outliers. In: Principles of data mining and knowledge discovery, Prague, Czech Republic, pp 262–270
Chan J, Bailey J, Leckie C (2008) Discovering correlated spatio-temporal changes in evolving graphs. Knowl Inf Syst 16(1):53–96
Changtien L, Dechang C, Yufeng K (2003) Algorithms for spatial outlier detection. In: Proceedings of 3rd IEEE international conference on data mining, Los Alamitos. IEEE Computer Society Press, pp 597–600
Cheng T, Li Z (2006) A multiscale approach for spatio-temporal outlier detection. Trans GIS 10(2):253–263
Cressie N, Wikle CK (2011) Statistics for spatio-temporal data. Wiley, Hoboken, New Jersey
Das M, Parthasarathy S (2009) Anomaly detection and spatio-temporal analysis of global climate system. In: SensorKDD '09: proceedings of the third international workshop on knowledge discovery from sensor data, New York, New York pp 142–150
Günnemann S, Kremer H, Laufkötter C, Seidl T (2012) Tracing evolving subspace clusters in temporal climate data. Data Min Knowl Discov 24(2):387–410
Huang Y, Zhang L, Zhang P (2008) A framework for mining sequential patterns from spatio-temporal event data sets. IEEE Trans Knowl Data Eng 20(4):433–448
Janeja V, Atluri V (2008) Random walks to identify anomalous free-form spatial scan windows. IEEE Trans Knowl Data Eng 20(10):1378–1392
Knorr EM, Ng RT, Tucakov V (2000) Distance-based outliers: algorithms and applications. VLDB J Int J Very Large Data Bases 8(3–4):237–253
Kou Y, Lu C, Santos R (2007) Spatial outlier detection: a graph-based approach. In: 2007 ICTAI 2007 19th IEEE international conference on tools with artificial intelligence, Patras, Greece, vol 1
Kulldorff M (1997) A spatial scan statistic. Commun Stat Theory Methods 26(6):1481–1496
Lin Y, Mitchell KE (2005) The NCEP stage II/IV hourly precipitation analyses: development and applications. In: 19th conference on hydrology. American Meteorological Society, San Diego, 9–13 Jan 2005
Lin F, Xie K, Song G, Wu T (2009) A novel spatio-temporal clustering approach by process similarity. In: 2009 FSKD '09 sixth international conference on Fuzzy systems and knowledge discovery, Tainjin, China, vol 5, pp 150–154
McGuire M, Janeja V, Gangopadhyay A (2011) Characterizing sensor datasets with multi-granular spatio-temporal intervals. In: Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems (GIS '11). ACM, New York
McGuire MP, Janeja VP, Gangopadhyay A (2014) Mining trajectories of moving dynamic spatio-temporal regions in sensor datasets. Data Min Knowl Discov 28(4): 961–1003
NOAA (2000) Tropical atmosphere ocean project. http://www.pmel.noaa.gov/tao/jsdisplay/
Oliveira M, Gama J (2010) Bipartite graphs for monitoring clusters transitions. In: Advances in intelligent data analysis IX, Tuscon, Arizona, pp 114–124
Reljin I, Reljin DB, Jovanović G (2003) Clustering and mapping spatial-temporal datasets using som neural networks. J Autom Control 13(1):55–60
Rosswog J, Ghose K (2008) Detecting and tracking spatio-temporal clusters with adaptive history filtering. In: 2008 ICDMW '08 IEEE international conference on data mining workshops, Pisa, Italy, pp 448–457
Sap MNM, Awan A (2005) Finding spatio-temporal patterns in climate data using clustering. In: 2005 international conference on cyberworlds, pp 8–164. doi:10.1109/CW.2005.45
Sedaghat L, Hersey J, McGuire MP (2013) Detecting spatio-temporal outliers in crowdsourced bathymetry data. In: Proceedings of the second ACM SIGSPATIAL international workshop on crowdsourced and volunteered geographic information. ACM, New York, pp 55–62
Shekhar S, Lu C, Zhang P (2003) A unified approach to detecting spatial outliers. GeoInformatica 7(2): 139–166
Spiliopoulou M, Ntoutsi I, Theodoridis Y, Schult R (2006) Monic: modeling and monitoring cluster transitions. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 706–711
Steinhaeuser K, Ganguly A, Chawla N (2012) Multivariate and multiscale dependence in the global climate system revealed through complex networks. Clim Dyn 39(3-4): 889–895
Sun P, Chawla S (2004) On local spatial outliers. In: 2004 ICDM '04 Fourth IEEE international conference on data mining, pp 209–216. doi:10.1109/ICDM.2004.10097
Von Storch H, Zwiers F (2002) Statistical analysis in climate research. Cambridge University Press, Cambridge
Worboys M, Duckham M (2006) Monitoring qualitative spatiotemporal change for geosensor networks. Int J Geogr Inf Sci 20(10):1087–1108
Wu E, Liu W, Chawla S (2010) Spatio-temporal outlier detection in precipitation data. In: Knowledge discovery from sensor data. CRC Press, Boca Raton, Florida, pp 115–133
Yang H, Parthasarathy S, Mehta S (2005) A generalized framework for mining spatio-temporal patterns in scientific data. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining. ACM, New York, pp 716–721

Data Modeling

▸ Modeling and Multiple Perceptions

Data Models

▸ Application Schema
▸ Data Models in Commercial GIS Systems

Data Models in Commercial GIS Systems

Erik Hoel
Environmental Systems Research Institute, Redlands, CA, USA

Synonyms

Data models; Data representations; Raster models; Vector 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 non-trivial (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 m. 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 hydrometeorological), 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.

Cross-References

- ▸ Oracle Spatial, Raster Data
- ▸ Spatial Data Transfer Standard (SDTS)
- ▸ Voronoi Diagram

Recommended Reading

Bernhardsen T (2002) Geographic information systems: an introduction, 3rd edn. John Wiley, New York

Chrisman N (2006) Charting the unknown: how computer mapping at Harvard became GIS. ESRI Press, Redlands

Cooke D, Maxfield W (1967) 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

DeMers M (2005) Fundamentals of geographic information systems, 3rd edn. John Wiley, New York

Hoel E, Menon S, Morehouse S (2003) Building a robust relational implementation of topology. In: Proceedings of symposium on advances in spatial and temporal databases (SSTD), Santorini Island, pp 508–524

Laurini R, Thompson D (1992) Fundamentals of spatial information systems. Academic, London

Longley P, Goodchild M, Maguire D, Rhind D (2005) Geographic information systems and science, 2nd edn. John Wiley, Chichester

Peucker T, Chrisman N (1975) Cartographic data structures. Am Cartogr 2(1):55–69

Rigaux P, Scholl M, Voisard A (2002) Spatial databases with application to GIS. Morgan Kaufmann, San Francisco

Samet H (2006) Foundations of multidimensional and metric data structures. Morgan Kaufmann, San Francisco

Schneider M (1997) Spatial data types for database systems: finite resolution geometry for geographic information systems. Lecture notes in computer science, vol 1288. Springer, Berlin
Shekhar S, Chawla S (2003) Spatial databases: a tour. Prentice Hall, Upper Saddle River
Thill J-C (2000) Geographic information systems in transportation research. Elsevier Science, Oxford
Tomlinson R (2003) Thinking about GIS: geographic information system planning for managers. ESRI Press, Redlands
Zeiler M (1999) Modeling our world: the ESRI guide to geodatabase design. ESRI Press, Redlands

Data Quality

▸ Spatial Data Transfer Standard (SDTS)

Data Representations

▸ Data Models in Commercial GIS Systems

Data Schema

▸ Application Schema

Data Semantics

▸ Semantic Kriging

Data Stream Systems, Empowering with Spatiotemporal Capabilities

Mohamed Ali
Center for Data Science, Institute of Technology, University of Washington, Tacoma, WA, USA

Synonyms

Complex event processing; Data streams; Geostreaming; StreamInsight

Definition

Spatiotemporal data streaming (or geostreaming) refers to the acquisition, processing, and analysis of stream data that has geographical locations and/or spatial extents such as point coordinates, lines, or polygons.

Real-time stream data acquisition through sensors and probes has been widely used in numerous applications. Hence, integrating spatial operators in commercial data-streaming engines has gained tremendous interest in recent years. In this entry, we consider the Microsoft StreamInsight (StreamInsight, for brevity) as our industrial case study. We highlight the background beyond its temporal model and discuss the various efforts that leverage its temporal model to the spatial domain.

Historical Background

Spatial queries and operations are common and essential for a variety of location-aware applications, e.g., find out the gas stations nearby a driver's location. During the last decade, accommodating spatial queries, e.g., *K Nearest Neighbor (KNN)* query, *Reverse Nearest Neighbor (RNN)* query, and *range* query, in data stream processing engines has attracted the database researchers' interest. Supporting spatiotemporal features in a *Data Stream Management System* (*DSMS*) requires the system to be equipped with especial indices, e.g., *R-tree*, and operators, e.g., *intersect* or *overlap*. *DSMSs* consolidate streams of data from multiple sources and with different formats and types (including the spatial types) and evaluate the issued queries in low response times. Consequently, the data stream query processor extracts interesting patterns and trends from the feeded spatial and nonspatial data in real time, Abadi et al. (2005), Chandrasekaran et al. (2003), Cranor et al. (2003), and StreamBase Inc.

Scientific Fundamentals

Fundamentally, a *Data Stream Management System* gives its connected applications the ability to issue *continuous queries* that digest and evaluate streams of data in real-time basis (Ali et al. 2009; Chandramouli et al. 2009; Barga et al. 2007). Moreover, a streaming engine is expected to include an extensibility mechanism to smoothly combine domain-specific rules and policies into the query pipeline. Here, we consider Microsoft StreamInsight as an example data streaming. StreamInsight has been designed to be an extensible system that is able to incorporate user-defined modules and functions and execute them as part of the continuous query processing plan (Ali et al. 2011). Furthermore, streaming applications and systems require the *continuous query processing* engine to guarantee the ability to digest input data with high rates and with incomplete and/or inaccurate values.

To these ends, the StreamInsight is engineered to handle imperfections in event delivery and also to assure the consistency of the returned final results. Consistency here can be interpreted as set of tests to confirm the correctness of the generated answers before being delivered to the query issuer. Consistency also means that obsolete and missed tuples should not significantly affect the validity of the output.

To guarantee the efficiency and consistency measurements when dealing with spatial data, StreamInsight is extended with Microsoft SQL Server Spatial Library SQL which provides a simple an easy to use, scalable, and highly efficient execution environment for spatial data analysis and processing. SQL Spatial Library provides data type support for point, line, and polygon objects. Also, various methods are provided to handle these spatial data types. SQL Spatial Library adheres to the *Open Geospatial Consortium Simple Feature Access* specification (Open Geospatial Consortium) and is provided as part of the SQL Server Types Library.

The following brief explanation of terms, features, and components is crucial for understanding the event stream model in Microsoft StreamInsight. For more details, the reader is referred to Barga et al. (2007). A *physical stream* is a sequence of events. An event $e_i = \langle p, c \rangle$ is a notification from the outside world that contains (1) a payload $p = \langle p_1, \ldots, p_k \rangle$ and (2) a control parameter c that provides metadata. The control parameter includes an event generation time and a duration that indicate the period of time over which an event can influence output. We capture this temporal information by defining $c =< LE, RE >$, where the interval $[LE, RE)$ specifies the period (or lifetime) over which the event contributes to output. The left endpoint (LE) of this interval, also called start time, is the application time of event generation. The event start time is also called the event timestamp. Assuming the event lasts for x time units, the right endpoint of an event, also called end time, is simply $RE = LE + x$.

StreamInsight allows users to issue *compensations* (or corrections) for earlier reported events, by the notion of retractions (Barga et al. 2007; Motwani et al. 2003; Ryvkina et al. 2006), which indicates a modification of the lifetime of an earlier event. This is supported by an optional third control parameter RE_{new}, that indicates the new right endpoint of the corresponding event. Event deletion (called a full retraction) is expressed by setting $RE_{new} = LE$ (i.e., zero lifetime).

A *Canonical History Table* (*CHT*) is the logical representation of a stream. Each entry in a CHT consists of a lifetime (LE and RE) and the payload. All times are application times, as opposed to system times. Thus, StreamInsight models a data stream as a time-varying relation, motivated by early work on temporal databases by Jensen and Snodgrass (1992).

Table 1 shows an example CHT. This CHT can be derived from the actual physical events (either new inserts or retractions) with control

Data Stream Systems, Empowering with Spatiotemporal Capabilities, Table 1 Canonical History Table

ID	LE	RE	Payload
e_0	1	5	P_1
e_1	4	9	P_2

Data Stream Systems, Empowering with Spatiotemporal Capabilities, Table 2 Physical stream corresponding to CHT

ID	Type	LE	RE	RE_{new}	Payload
e_0	Insertion	1	∞	–	P_1
e_0	Retraction	1	∞	10	P_1
e_0	Retraction	1	10	5	P_1
e_1	Insertion	4	9	–	P_2

parameter $c = \langle LE, RE, RE_{new} \rangle$. For example, Table 2 shows one possible physical stream with an associated logical CHT shown in Table 1. Note that a retraction event includes the new right endpoint of the modified event. The CHT (Table 1) is derived by matching each retraction in the physical stream (Table 2) with its corresponding insertion and adjusting RE of the event accordingly.

We need to ensure that an event is not arbitrarily out of order; this is realized using time-based punctuations (Barga et al. 2007; Srivastava and Widom 2004; Tucker et al. 2003). A time-based punctuation is a special event that is used to indicate time progress. These punctuations are called *Current Time Increments* (*CTIs*) in StreamInsight. A CTI is associated with a timestamp t and indicates that there will be no future event in the stream that modifies any part of the time axis that is earlier than t. Note that we could still see retractions for events with LE less than t, as long as both RE and RE_{new} are greater than or equal to t.

There are two approaches for the spatiotemporal stream processing within StreamInsight: an extensibility approach and a native support approach. The extensibility approach combines the values of the StreamInsight extensibility framework and the SQL Spatial Library by giving the *UDM* writers the ability to invoke the library methods within their code. Alternatively, the native support approach deals with spatial attributes as first-class citizens and reasons about the spatial properties of incoming events and, more interestingly, provides consistency guarantees over space as well as time. For details on these two approaches, the reader is referred to Ali et al. (2010) and Jeremiah et al. (2011).

Key Applications

Spatiotemporal stream engines such as Microsoft StreamInsight are beneficial in many real applications and systems. Here we give two brief examples of these applications.

Traffic Management Systems

In a traffic management scenario, the system answers queries about the past, current, and future road conditions. Further, it suggests the best driving directions for newly added vehicles by taking future road conditions into consideration. Note that as long as the vehicle is on track, i.e., following the route planned by the system according to the expected speed, there is no need for the vehicle to transmit any events to the system, which results in reducing transmission load over the wireless network. However, if the vehicle changes its route selection policy, makes an unexpected turn, or stops for some time, the vehicle generates retraction and insertion events to adjust its path. In response to the retraction event, the system updates the result of its CQs and possibly generates compensation events or new speculative output. Further, we could define a *spatiotemporal algebra* with new streaming operators that natively take location into consideration; for example, we may add a spatiotemporal *left-semi-join* operator that accepts a proximity metric and outputs events related to the left input object only when it overlaps in time as well as space (within the proximity metric) with a matching object on the right input. For a detailed discussion on this application scenario and a streaming approach to the solution, the reader is referred to Ali et al. (2010) and Jalal et al. (2010).

Criminal Activity Tracking and Monitoring Systems

Court orders may require supervising agencies to track and monitor a specific set of offenders using ankle bracelets. According to the decision of the criminal justice system, each offender with a tracking device is assigned a designated spatiotemporal curfew. This curfew typically consists of confinement zones to which the offender

is detained to and a set of restricted zones to which he is obliged to stay away from.

For example, an offender may be required to stay home at night during a court-ordered curfew. Also, a sex offender would be restricted from visiting school zones. Offenders are free to move around without the monitoring agencies being alerted as long as they remain within the designated confinement regions and as long as they do not enter restricted zones. A spatiotemporal DSMS helps (1) detect unauthorized activities in real time and provide alerts to a community corrections officer, a law enforcement dispatcher, or a control center and (2) mine for the offenders' suspicious behavior and predict probable future threats beforehand. Unauthorized activities include protecting geographically defined regions (e.g., school zones) in which the offender is not allowed to be present. Suspicious behaviors include the meeting of offenders with each other on a regular basis, possibly near restricted zone. For a detailed discussion on this application scenario and a streaming approach to the solution, the reader is referred to Daubal et al. (2013).

Future Directions

Future directions for spatiotemporal data stream management systems would focus on *big spatial data* processing and analysis. In this paradigm, the geospatial data streaming (or geostreaming) will serve a key role at the intersection of mobility and cloud computing (Shekhar et al. 2012). Geostreaming will establish the query processing pipeline between the mobile devices with their streams of location updates and the cloud storage.

References

Abadi D et al (2005) The design of the Borealis stream processing engine. In: CIDR. Asilomar, CA

Ali M et al (2009) Microsoft CEP server and online behavioral targeting. In: VLDB. Lyon, France

Ali M, Chandramouli B, Sethu Raman B, Katibah E (2010) Spatio-temporal stream processing in microsoft streaminsight. IEEE Data Eng Bull 33(2): 69–74

Ali M, Chandramouli B, Goldstein J, Schindlauer R (2011) The extensibility framework in microsoft streaminsight. In: ICDE. Hannover, Germany

Barga R et al (2007) Consistent streaming through time: a vision for event stream processing. In: CIDR. Asilomar, CA

Chandrasekaran S et al (2003) TelegraphCQ: continuous dataflow processing for an uncertain world. In: CIDR. Asilomar, CA

Chandramouli B, Goldstein J, Maier D (2009) On-the-fly progress detection in iterative stream queries. In: VLDB. Lyon, France

Cranor C et al (2003) Gigascope: a stream database for network applications. In: SIGMOD. San Diego, CA

Daubal M, Fajinmi O, Jangaard L, Simonson N, Yasutake B, Newell J, Ali M (2013) Safe step: a real-time gps tracking and analysis system for criminal activities using ankle bracelets. In: The ACM SIGSPATIAL conference on advances in geographic information systems, GIS. Orlando, FL

Jensen C, Snodgrass R (1992) Temporal specialization. In: ICDE. Tempe, AZ

Jeremiah M, Raymond M, Archer J, Adem S, Hansel L, Konda S, Luti M, Zhao Y, Teredesai A, Ali M (2011) An extensibility approach for spatio-temporal stream processing using microsoft streaminsight. In: The international symposium on spatial and temporal databases, SSTD. Minneapolis, MN

Kazemitabar SJ, Demiryurek U, Ali MH, Akdogan A, Shahabi C (2010) Geospatial stream query processing using microsoft sql server streaminsight. In: VLDB. Singapore

Motwani R et al (2003) Query processing, approximation, and resource management in a DSMS. In: CIDR. Asilomar, CA

Open Geospatial Consortium. http://www.opengeospatial.org/standards/sfa (Last Accessed March 2016)

Ryvkina E et al (2006) Revision processing in a stream processing engine: a high-level design. In: ICDE. Atlanta, GA

Shekhar S, Evans MR, Gunturi V, Yang K (2012) Spatial big-data challenges intersecting mobility and cloud computing. In: The NSF workshop on social networks and mobility in the cloud. Washington DC

SQL Server Spatial Libraries. http://www.microsoft.com/sqlserver/2008/en/us/spatial-data.aspx (Last Accessed March 2016)

Srivastava U, Widom J (2004) Flexible time management in data stream systems. In: PODS. Paris, France

StreamBase Inc. http://www.streambase.com/ (Last Accessed March 2016)

Tucker P et al (2003) Exploiting punctuation semantics in continuous data streams. In: IEEE TKDE

Data Streams

▸ Data Stream Systems, Empowering with Spatiotemporal Capabilities

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

- ▸ Spatiotemporal 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 (Gray 2006). By 1990, the concepts of data warehouse had developed enough that the first major data warehouse textbook appeared (Inmon 1990). 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 (1995). The growth in the markets for data warehousing was driven by the expanding data storage and its analytical uses in organizations. During the past 15 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 (Gray 2006).

Early geographic information systems were associated with databases, but not until much later with data warehouses. In the past 5 years, some data warehouse products such as Oracle (Rittman 2006a, b) 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.

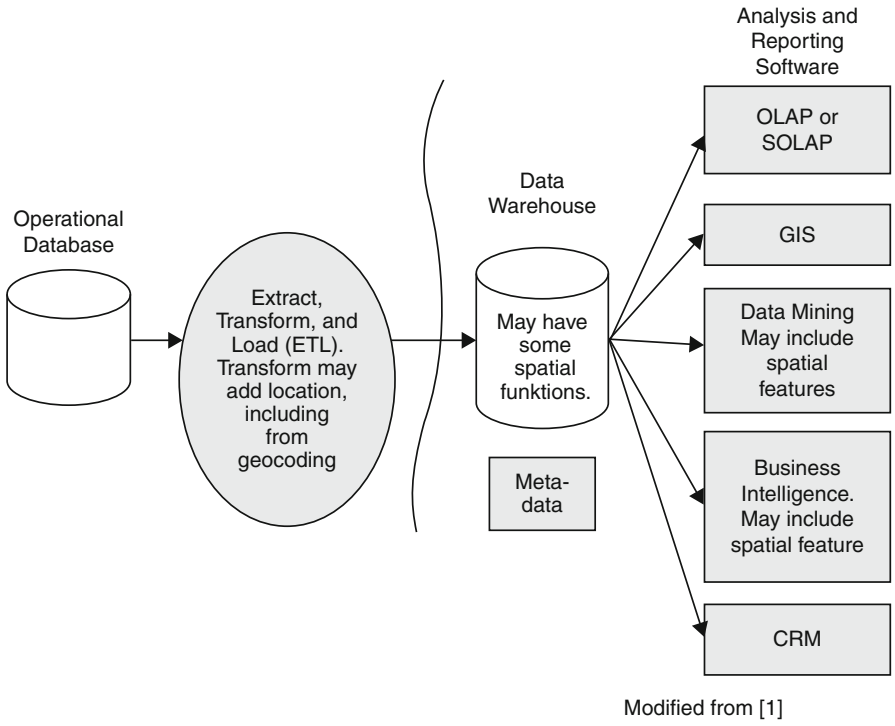
A data warehouse differs from a relational database in the following key characteristics: data warehouses are subject-oriented, time-variant, non-volatile, integrated, and oriented towards users who are decision-makers (Gray 2006; Gray and Watson 1998). *Subject-oriented* means that the user accesses 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 (Inmon 1990). 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 (Gray 2006). 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 (Jukic 2006), 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* (Gray 2006).



Data Warehouses and GIS, Fig. 1 The data warehouse and its data flows, spatial functions, and GIS components

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 (Gray 2006; Gray and Watson 1998). 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 (Gray 2006; Jukic 2006). 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 (Gray 2006; Jukic 2006). 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* (Gray and Watson 1998). 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) 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 data-bases 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 (SQL Server Magazine 2002; Reid 2006). 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.

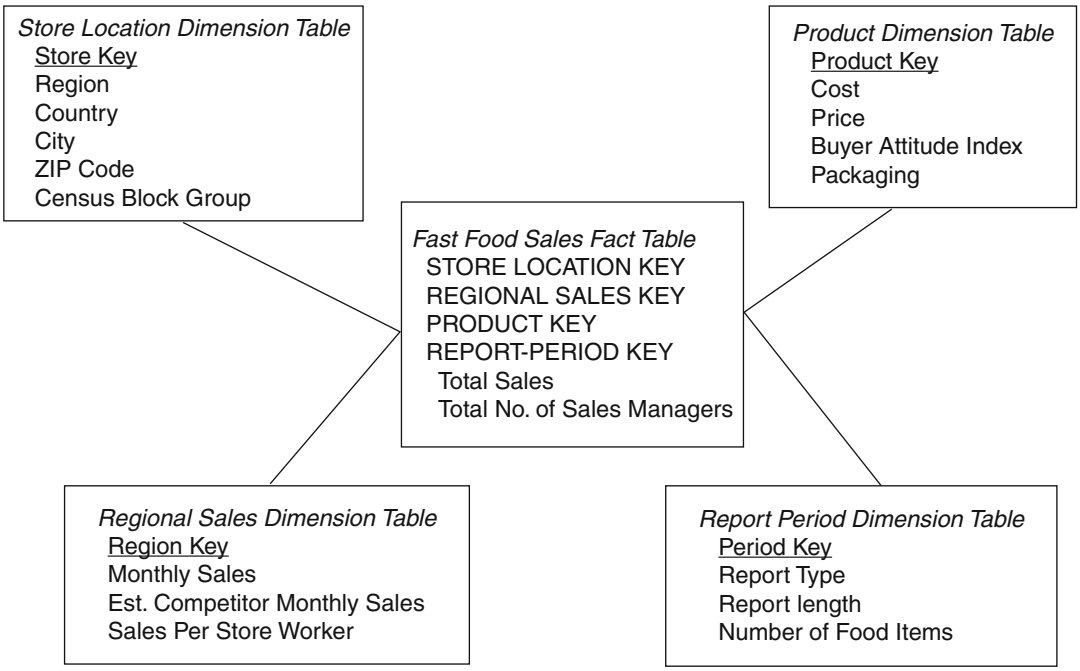
Online analytical processing (OLAP) is a set of rules for accessing and processing multidimensional data in the data warehouse. OLAP rules are focused on simple business decision-making that directly accesses the dimensions of data in the warehouse rather than on complex models. Among the main types 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 (Gray 2006). OLAP is good at answering "why" and "what if" questions.

Specifically, OLAP refers the following characteristics of the information in the data warehouse (Codd 1995): (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* (Bimonte et al. 2005; Malinowski and Zimanyi 2003; Marchand et al. 2003). 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" (Bimonte et al. 2005).

SOLAP models are still under development, in particular to formulate improved SOLAP-based operators for spatial analysis, and more elaborate working prototypes (Bimonte et al. 2005). 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 (Gray 2006). In the context of data warehouses, BI can conduct modeling based on information from the data warehouse for forecasting, simulations,



Data Warehouses and GIS, Fig. 2 Data warehouse star schema, with location included

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 (Gray 2006). The variety of data mining methods include intelligent agents, expert systems, fuzzy logic, neural networks, exploratory data analysis, and data visualization (Gray 2006; Codd 1995). 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 (Viswanathan et al. 2005). 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 databases, 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 (Oracle 2006).

Large data warehouses may store many terabytes of information, cost several million dollars, and take up to 2 or 3 years to implement (Gray 2006). Their pluses include better decision-making capability, faster access, retention of data for longer time periods, and enhanced data quality (Oracle 2006). 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 (Rittman 2006a). For viewing, Oracle Spatial 10g's low-level "Map Viewer" provides simple maps of the dimensional attributes and summary data in the data warehouse (Rittman 2006b). 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 (Gray 2006; SQL Server Magazine 2002; Reid 2006; Pick et al. 2000). In this section, two examples are given of real-world applications: (1) auto insurance applications (Bimonte et al. 2005) and (2) traffic patterns for the city area of Portland, Oregon, over a time span of almost two decades (SQL Server Magazine 2002).

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 (Bimonte et al. 2005). 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 (Bimonte et al. 2005).

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 pre-defined 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 15 years of data and 14 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 (SQL Server Magazine 2002). 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 (SQL Server Magazine 2002). 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 (SQL Server Magazine 2002). Because of the scope of the programs, CGIS limited the types of questions users can ask to a pre-defined 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 (SQL Server Magazine 2002).

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

References

Bimonte S, Tchounikine A, Miquel M (2005) Towards a spatial multidimensional model. In: Proceedings of the 8th ACM international workshop on data warehousing and OLAP, Bremen, 4–5 Nov, pp 39–46

Codd EF (1995) Twelve rules for on-line analytic processing. Computerworld 29:84–87

Gray P (2006) Manager's guide to making decisions about information systems. John Wiley, New York

Gray P, Watson H (1998) Decision support in the data warehouse. Prentice Hall, Upper Saddle River
Inmon WH (1990) Building the data warehouse. John Wiley, New York
Jukic N (2006) Modeling strategies and alternatives for data warehousing projects. Commun ACM 49(4):83–88
Malinowski E, Zimanyi E (2003) Representing spatiality in a conceptual multidimensional model. In: Proceedings of the 12th annual ACM international workshop on geographic information systems. ACM, Washington, DC, pp 12–22
Marchand P, Brisebois A, Bedard Y, Edwards G (2003) Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis. J Int Soc Photogramm Remote Sens 59:6–20
Oracle (2006) Oracle data warehousing. Available at www.oracle.com
Pick JB, Viswanathan N, Hettrick WJ (2000) A dual census geographical information system in the context of data warehousing. In: Proceedings of the Americas conference on information systems. Association for Information Systems, Atlanta, pp 265–278
Reid H (2006) Applying Oracle spatial to a very large insurance problem. Location Intelligence, 20 June. Available at www.locationintelligence.net
Rittman M (2006) An introduction to Oracle warehouse builder 10g. DBAzine.com. 19 Aug. Available at www.dbazine.com
Rittman M (2006) GIS-enabling your Oracle data warehouse. DBAzine.com. 18 Apr. Available at www.dbazine.com
SQL Server Magazine (2002) City of Portland Tames massive SQL server data warehouse. 27 June. Available at www.sqlmag.com
Viswanathan N, Pick JB, Hettrick WJ, Ellsworth E (2005) An analysis of commonality in the twin metropolitan areas of San Diego, California and Tijuana, Mexico. J Soc Plan Sci 39:57–79

Recommended Reading

Tan P-N, Steinbach M, Kumar V (2005) Introduction to data mining. Addison Wesley, Upper Saddle River

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

Data warehousing; Peer data management; Resolving semantic schema heterogeneity; Schema mapping

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.

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 Shu et al. (1975).

Scientific Fundamentals

Schema Mapping Creation

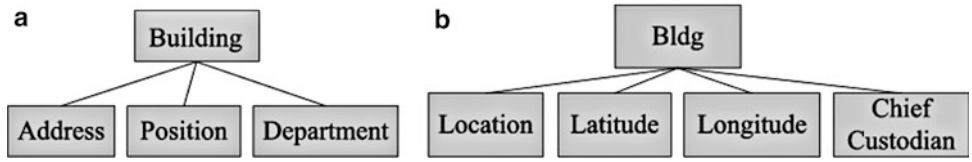
Before the schemas can be integrated, the similarities between attributes must be determined so that a mapping can be created between the schemas. For example, given the relations in Tables 1 and 2, it is obvious to a human that "Position" in Building (Table 1) corresponds to the concatenation of "Latitude" 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 only present

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



Database Schema Integration, Fig. 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)

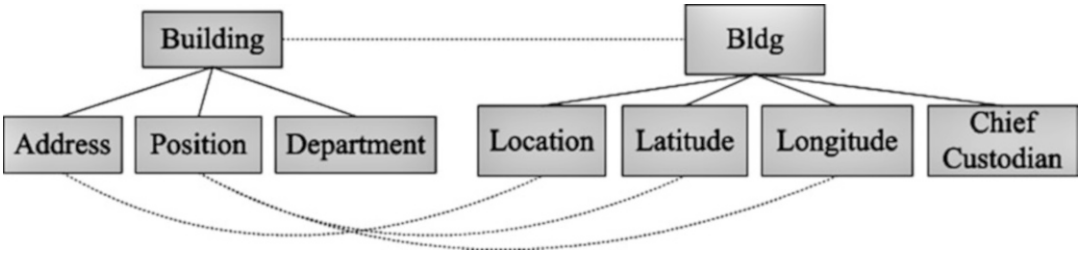
in the designers' heads, any semiautomatic approach must defer the ultimate decision about when two concepts are equal to a human being. Thus, the goal of *schema matching* is to create computer tools to leverage a person's decisions to make the best possible mappings. A survey of the techniques can be found in Rahm and Bernstein (2001). 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, 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 obvi-



Database Schema Integration, Fig. 2 A likely mapping between the elements in Fig. 1

ously 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) Doan et al. (2001) is a system that creates mappings using machine learning methods (see Mitchell (1997) 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 matchers) 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 (Melnik et al. 2002); 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 (Miller et al. 2000) 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 semistructured 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 Tables 1 and 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 et al. (1986) provide a thorough explanation of what considerations have to be made, and also provide a survey of some early work. Additional work on this problem can be found in other sources, including (Buneman et al. 1992; Melnik et al. 2003; Pottinger and Bernstein 2000).

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

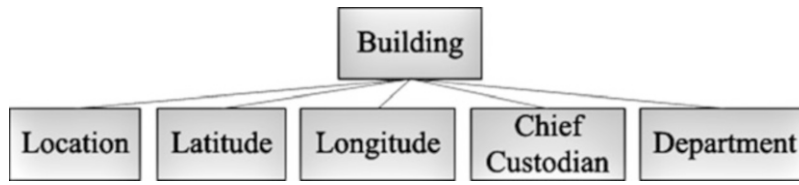
In data warehousing (Fig. 4a), the data is imported from the sources. For this to happen, a global schema needs to be created for the source data to be imported into. This schema can be created through a process such as the one described to create the schema in Fig. 3. An example of such a system can be found in Calvanese et al. (1998). 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 (Gray et al. 1997) 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

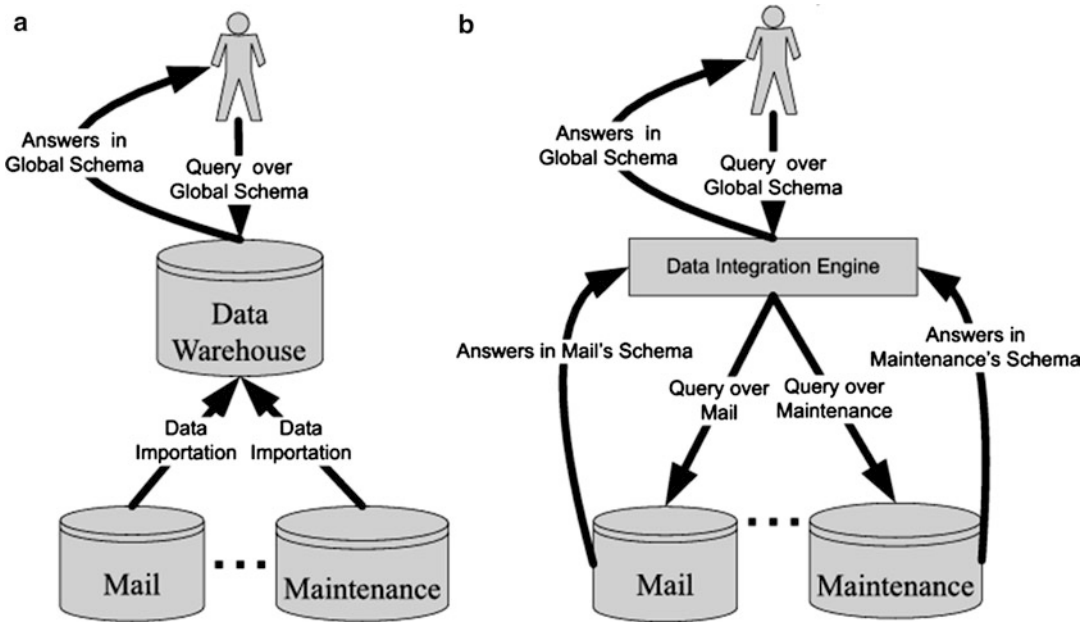
In a database schema integration system, as shown in Fig. 4b, the data are maintained separately at each source. At query time each query over the global schema has to be reformulated into a query over a local source (see Ullman (1997) for a survey).

Peer Data Management Systems

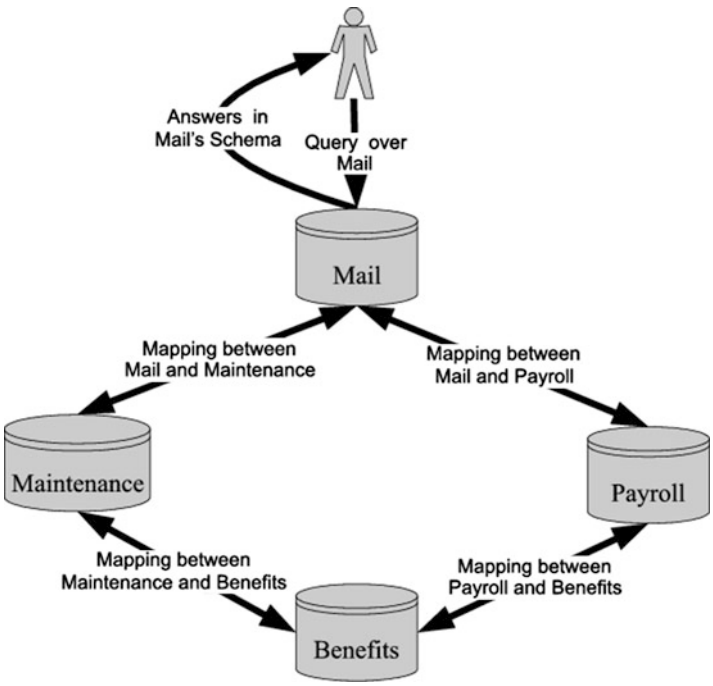
A peer-to-peer (P2P) system is a loose confederation of sources, such as the one shown in Fig. 5, where each peer may contain different data, and is free to come and go at any time. Typical P2P networks such as Napster have no



Database Schema Integration, Fig. 3 A possible global schema created from the input schemas in Fig. 1



Database Schema Integration, Fig. 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



Database Schema Integration, Fig. 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

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., Bernstein et al. 2002; Halevy et al. 2003). 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, 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., Fagin et al. 2004; Madhavan and Halevy 2003).

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., Dhamankar et al. 2004) 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 (Madhavan et al. 2006).

Cross-References

- Conflation of Geospatial Data
- Geocollaboration
- Geospatial Semantic Integration
- Geospatial Semantic Web, Interoperability
- Metadata and Interoperability, Geospatial
- Mobile P_2P Databases
- Ontology-Based Geospatial Data Integration

References

Batini C, Lenzerini M, Navathe SB (1986) A comparative analysis of methodologies for database schema integration. ACM Comput Surv 18:323–364

Bernstein PA, Giunchiglia F, Kementsietsidis A, Mylopoulos J, Serafini L, Zaihrayeu I (2002) Data management for peer-to-peer computing: a vision. In: International workshop on the web and databases (WebDB), Madison, 6–7 June 2002

Buneman P, Davidson SB, Kosky A (1992) Theoretical aspects of schema merging. In: International conference on extending database technology (EDBT), Vienna, 23–27 Mar 1992

Calvanese D, Giacomo GD, Lenzerini M, Nardi D (1998) Rosati: schema and data integration methodology for DWQ. Technical report, DWQ Consortium. TR# DWQ-UNIROMA-004, Sept 1998, Università di Roma "La Sapienza"

Dhamankar R, Lee Y, Doan A, Halevy AY, Domingos P (2004) iMAP: discovering complex mappings between database schemas. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD), Paris, 13–18 June 2004

Doan A, Domingos P, Halevy A (2001) Reconciling schemas of disparate data sources: a machine learning approach. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD), Santa Barbara, 21–24 May 2001

Fagin R, Kolatis PG, Popa L, Tan WC (2004) Composing schema mappings: second-order dependencies to the rescue. In: Symposium on principles of database systems (PODS), Paris, 14–16 June 2004

Gray J, Chaudhuri S, Bosworth A, Layman A, Reichart D, Venkatrao M, Pellow F, Pirahesh H (1997) Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub totals. Data Min Knowl Disc 1:29–53

Halevy AY, Ives ZG, Suciu D, Tatarinov I (2003) Piazza: data management infrastructure for semantic web applications. In: Proceedings of the international conference on data engineering (ICDE), Bangalore, 5–8 Mar 2003

Madhavan J, Halevy AY (2003) Composing mappings among data sources. In: Proceedings of the very large data bases conference (VLDB), Berlin, 9–12 Sept 2003

Madhavan J, Halevy AY, Cohen S, Dong XL, Jeffrey SR, Ko D, Yu C (2006) Structured data meets the web: a few observations. IEEE Data Eng Bull 29: 19–26

Melnik S, GarciaMolina H, Rahm E (2002) Similarity flooding: a versitile graph matching algorithm and its application to schema matching. In: Proceedings of the international conference on data engineering, Madison, 3–6 June 2002

Melnik S, Rahm E, Bernstein PA (2003) Rondo: a programming platform for generic model management. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD), San Diego, 9–12 June 2003

Miller RJ, Haas LM, Hernández MA (2000) Schema mapping as query discovery. In: Proceedings of the very large data bases conference (VLDB), Cairo, 10–14 Sept 2000

Mitchell TM (1997) Machine learning, 1st edn. McGraw-Hill, New York

Pottinger RA, Bernstein PA (2000) Merging models based on given correspondences. In: Proceedings of the very large data bases conference (VLDB), Cairo, 10–14 Sept 2000

Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. VLDB J 10: 334–350

Shu NC, Housel BC, Lum VY (1975) CONVERT: a high level translation definition language for data conversion. Commun ACM 18:557–567

Ullman JD (1997) Information integration using logical views. In: Proceedings of the international conference on database theory (ICDT), Delphi, 8–10 Jan 1997

Databases, Relational

▸ Constraint Databases, Spatial

Datalog, SQL

▸ Constraint Database Queries

Data-Structure

▸ Quadtree and Octree

Data-Structures

▸ Indexing Schemes for Multidimensional 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

Business application; Marketing information system

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 45 years ago (Simon 1960). 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 Bonczek et al. 1981 and Sprague 1980).

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 25 years ago, a somewhat prophetic *Harvard Business Review* article (Takeuchi and Schmidt 1980) 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 Armstrong and Densham (1990) and Keenan (2003).

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 deci-

sions, 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 Lucas (1979) and Money and Wegner (1988). 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 (Crossland 1992) 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 Crossland et al. (1995) and Mennecke et al. (2000).

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 (Cheng et al. 2002).

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 Karlsson et al. (2006) and Zhang et al. (2006).

Planning

A broad application area involves various types of planning decisions, including urban, environmental, and telecommunications infrastructure (Culshaw et al. 2006).

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 Choi et al. (2005) and Ochola and Kerkides (2004).

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 (Hendriks and Vriens 2000).

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 Johnson (2005) and Rushton (2003).

Forestry Management and Conservation

A number of SDSS applications focus specifically on forestry management and natural resource conservation Geneletti (2004) and Rao and Kumar (2004).

Traffic Management

SDSS has bee utilized in studies concerning traffic management systems, where vehicular traffic flows are documented and analyzed Randall et al. (2005) and Tarantilis and Kiranoudis (2002).

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 (Hess et al. 2004).

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 Keramitsoglou et al. (2003) and Martin et al. (2004).

Water Resources Management

Water resources management applications focus on tracking locations, flows, quality, and sustainability of water resources Liu (2004) and Nauta et al. (2003).

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

References

Armstrong AP, Densham PJ (1990) Database organization strategies for spatial decision support systems. Int J Geogr Inf Syst 4(1):3–20

Bonczek RH, Holsapple CW, Whinston AB (1981) Foundations of decision support systems. Academic, Orlando

Cheng MY, Ko CH, Chang CH (2002) Computer-aided DSS for safety monitoring of geotechnical construction. Autom Constr 11(4):375–390

Choi JY et al (2005) Utilizing web-based GIS and SDSS for hydrological land use change impact assessment. Trans ASAE 48(2):815–822

Crossland MD (1992) Individual decision-maker performance with and without a geographic information system: an empirical study. Unpublished doctoral dissertation, Kelley School of Business. Indiana University, Bloomington

Crossland MD, Wynne BE, Perkins WC (1995) Spatial decision support systems: an overview of technology and a test of efficacy. Decis Support Syst 14(3):219–235

Culshaw MG et al (2006) The role of web-based environmental information in urban planning – the environmental information system for planners. Sci Total Environ 360(1–3):233–245

Geneletti D (2004) A GIS-based decision support system to identify nature conservation priorities in an alpine valley. Land Use Policy 21(2):149–160

Hendriks P, Vriens D (2000) From geographical information systems to spatial group decision support systems: a complex itinerary. Geogr Environ Model 4(1):83–104

Hess RL, Rubin RS, West LA (2004) Geographic information systems as a marketing information system technology. Decis Support Syst 38(2):197–212

Johnson MP (2005) Spatial decision support for assisted housing mobility counseling. Decis Support Syst 41(1):296–312

Karlsson J, Ronnqvist M, Frisk M (2006) RoadOpt: a decision support system for road upgrading in forestry. Scand J For Res 21(7):5–15

Keenan PB (2003) Spatial decision support systems. In: Mora M, Forgionne G, Gupta, JND (eds) Decision making support systems: achievements and challenges for the new decade. Idea Group, Harrisburg, pp 28–39

Keramitsoglou I, Cartalis C, Kassomenos P (2003) Decision support system for managing oil spill events. Environ Manag 32(2):290–298

Liu CW (2004) Decision support system for managing ground water resources in the Choushui River alluvial in Taiwan. J Am Water Res Assoc 40(2):431–442

Lucas HC (1979) Performance and the use of an information system. Manag Sci 21(8):908–919

Martin PH et al (2004) Development of a GIS-based spill management information system. J Hazard Mater 112(3):239–252

Mennecke BE, Crossland MD, Killingsworth BL (2000) Is a map more than a picture? The role of SDSS technology, subject characteristics, and problem complexity on map reading and problem solving. MIS Q 24(4):601–629

Money ATD, Wegner T (1988) The quantification of decision support within the context of value analysis. MIS Q 12(2):223–236

Nauta TA, Bongco AE, Santos-Borja AC (2003) Set-up of a decision support system to support sustainable development of the Laguna de Bay, Philippines. Mar Pollut Bull 47(1–6):211–219

Ochola WO, Kerkides P (2004) An integrated indicator-based spatial decision support system for land quality assessment in Kenya. Comput Electron Agric 45(1–3):3–26

Randall TA, Churchill CJ, Baetz BW (2005) Geographic information system (GIS) based decision support for neighborhood traffic calming. Can J Civ Eng 32(1):86–98
Rao K, Kumar DS (2004) Spatial decision support system for watershed management. Water Resour Manag 18(5):407–423
Rushton G (2003) Public health, GIS, and spatial analytic tools. Annu Rev Public Health 24:43–56
Simon H (1960) The new science of management decisions. Harper and Row, New York
Sprague RH (1980) Framework for DSS. MIS Q 4(4): 1–26
Takeuchi H, Schmidt AH (1980) The new promise of computer graphics. Harv Bus Rev 58(1):122–131
Tarantilis CD, Kiranoudis CT (2002) Using a spatial decision support system for solving the vehicle routing problem. Inf Manag 39(5):359–375
Zhang BS et al (2006) Predictive modeling of hill-pasture productivity: integration of a decision tree and a geographical information system. Agric Syst 87(1):1–17

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

Degree Library; Degree Open Source Framework; Geo-portal; GML; GNU; ISO/TC 211; Java; OGC; OGC Web service; Open source; Public-domain software; SDI (Spatial Data Infrastructure); WCS; Web coverage service; Web map service; WMS; XML

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) (de La Beaujardière 2003) and Web Coverage Service (WCS) (Evans 2002) 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 (Fitzke et al. 2004). 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 (Vretanos ; Vretanos 2002), a Gazetteer (WFS-G, Atkinson and Fitzke 2006) and support of additional data sources. From this time on, deegree WMS supported SLD (Styled Layer Descriptor) (Lalonde 2002) and a Catalog Service (Nebert 2002). 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 (Müller and Poth 2004). This new client framework is named iGeoPortal and is part of deegree and supports the OGC standard Web Map Context 1.0.0 (Humblet 2003).

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) (Kiehle and Greve 2006; Schut and Whiteside 2005) 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 so-

phisticated 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.

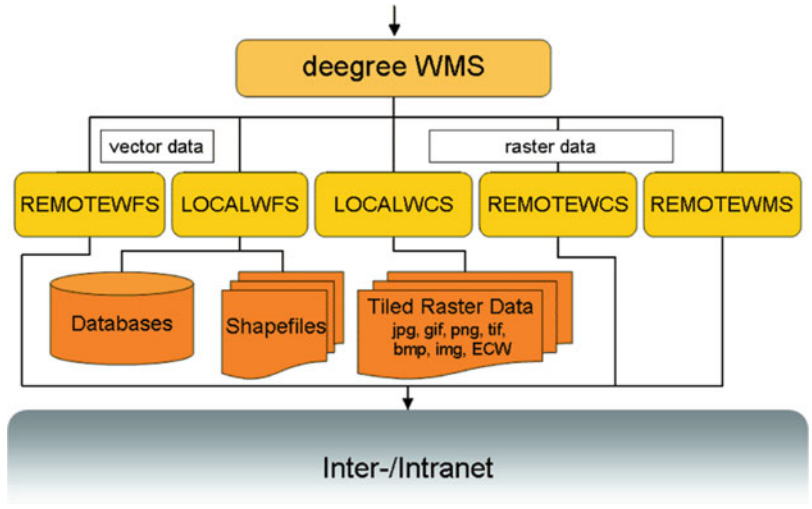
deegree as an open source/free software java project aims to offer these services as well as a client framework and API for more basic GIS functions to enable the realization of highly configurable and flexible SDIs.

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.

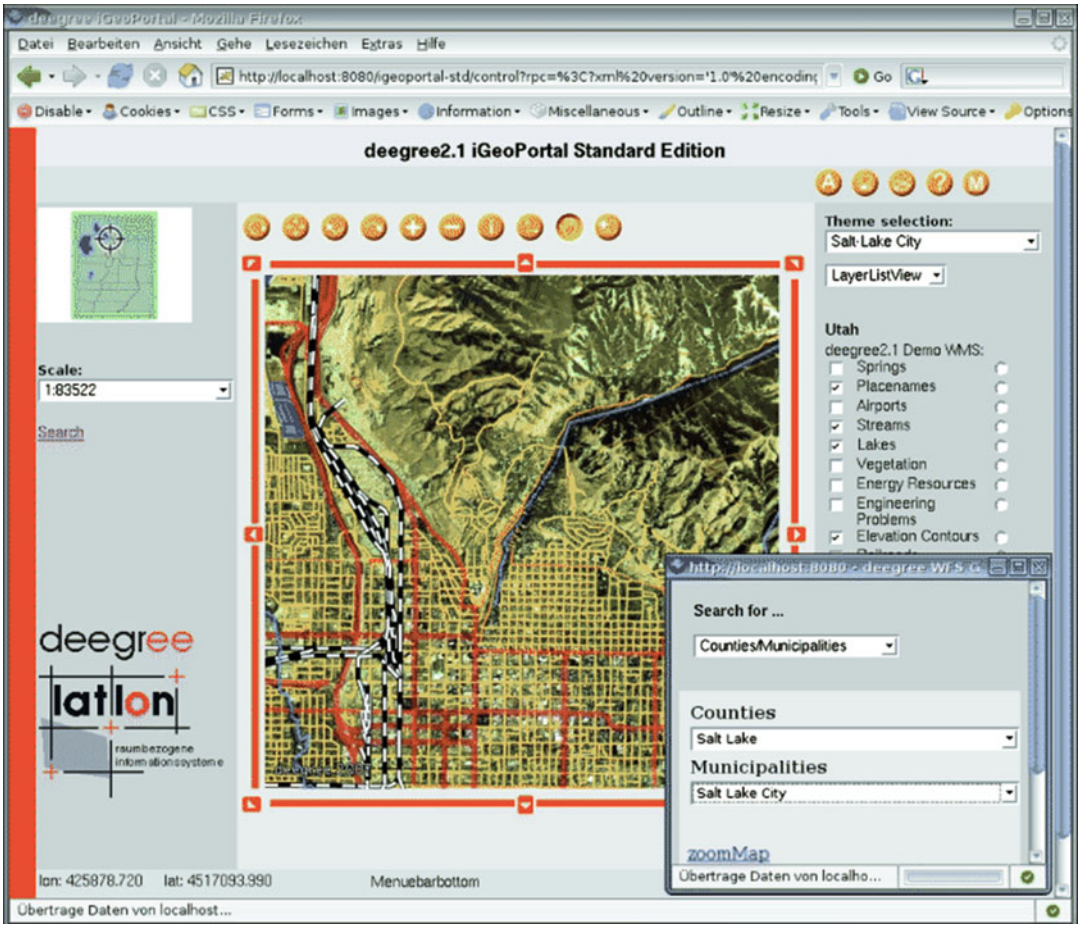
deegree WMS supports WMS versions 1.1.0, 1.1.1 and 1.3. It can access vector and raster data. For access to vector data, the WFS interface is used inside deegree, while for raster data access, the WCS interface is used. The terms "REMOTEWFS" and "REMOTEWCS" denote the possibility of using WFS and WCS 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 applica-



deegree Free Software, Fig. 1 Architecture diagram of deegree Web Map Service (*WMS*). *WCS* Web Coverage Service, WFS



deegree Free Software, Fig. 2 deegree iGeoPortal

tion 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 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 (Nebert 2002) as well as the so-called ISO Application Profile (Voges and Senkler 2004).

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)
- ▸ Metadata and Interoperability, Geospatial
- ▸ National Spatial Data Infrastructure (NSDI)
- ▸ OGC's Open Standards for Geospatial Interoperability
- ▸ Web Mapping and Web Cartography
- ▸ Web Services, Geospatial

References

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

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

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
Fitzke J, Greve K, Müller M, Poth A (2004) Building SDIs with free software – the deegree project. In: Proceedings of CORP 2004, Vienna, 25–17 Feb 2004. Available via http://www.corp.at
Humblet JP (ed) (2003) Web map context documents. OpenGIS Project Document 03-036r2. https://portal.opengeospatial.org/files/?artifact_id=3841
Kiehle C, Greve K (2006) Standardized geoprocessing – taking spatial data infrastructures one step further. In: Proceedings of AGILE, Minneapolis, 23–28 July 2006
Lalonde W (ed) (2002) Styled layer descriptor implementation specification. OpenGIS project document 02-070. https://portal.opengeospatial.org/files/?artifact_id=1188
Müller M, Poth A (2004) Deegree-components and framework for creation of SDIs. Geoinformatics 7
Nebert D (ed) (2002) OpenGIS catalog service specification. OpenGIS project document 04-021r3. http://portal.opengeospatial.org/files/?artifact_id=5929&version=2
Schut P, Whiteside A (ed) (2005) OpenGIS web processing service. OGC project document 05-007r4. OGC
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
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
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

Degree Library

▸ deegree Free Software

Degree 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

Destination Prediction

▸ Spatial Predictive Query Processing on Euclidean Space
▸ Spatial Predictive Query Processing on Road Networks

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 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 Pathology

▸ Medical Image Dataset Processing over Cloud/MapReduce with Heterogeneous Architectures

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

4IM; 9IM; Clementini Operators; DE-9IM; Egenhofer Operators; Four-Intersection Model; Nine-Intersection Model; 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.

- FourIntersection Model (4IM): Boolean set of operations (considering intersections between boundary and interior)
- NineIntersection Model (9IM): Egenhofer operators (taking into account exterior, interior and boundary of objects)
- **Dimensionally Extended NineIntersection 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 (Clementini et al. 1993; Clementini and Di Felice 1994, 1996) dimensionally extends the Nine-Intersection Model (9IM) of Egenhofer and Herring (1991). The Nine-Intersection Model in turn extends the Four-Intersection Model (4IM) from Egenhofer (1989) and Egenhofer and Herring (1990, 1991) by adding the intersections with the exteriors (Egenhofer et al. 1993, 1994).

The **Dimensionally Extended Nine-Intersection Model (DE-9IM)** is accepted by the ISO/TC 211 (ISO/TC211 2003) and by the Open Geospatial Consortium (OGC 2005), 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 (Table 1).

- 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.

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 (OGC 2005)

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

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 NineIntersection Matrix (DE-9IM) after Clementini and Di Felice (1996). 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 NineIntersection Model provides eight such spatial relationships between points, lines and polygons (q.v. OGC (2005) 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).

"Within": Example DE-9IM for the case where A is a Line which lies within the Polygon B.

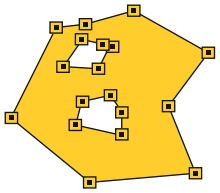
"Contains": Example DE-9IM for the case where A is a MultiPoint Object (squares) which contains another MultiPoint B (circles).

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 patternvalues, 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:

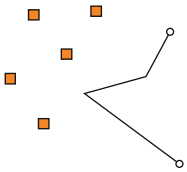
Dimensionally Extended Nine-Intersection Model (DE-9IM), Table 2 Topological predicates and their corresponding meanings after the dimensionally extended NineIntersection model, from Davis and Aquino (2003)

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)



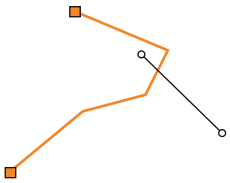
	Interior (B)	Boundary (B)	Exterior (B)
Interior (A)	2	–1	–1
Boundary (A)	–1	1	–1
Exterior (A)	–1	–1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Fig. 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), Fig. 2 Example for a "Disjoint"-relationship between a Line A and a 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), Fig. 3 Example for a "Disjoint"-relationship between a Line A and a MultiPoint B

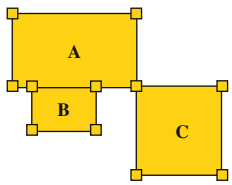
$p = \text{T} \Rightarrow \dim(x) \in (0, 1, 2)$, i. e. $x \neq \emptyset$
$p = \text{F} \Rightarrow \dim(x) = -1$, i. e. $x = \emptyset$
$p = * \Rightarrow \dim(x) \in (-1, 0, 1, 2)$, i. e.,Don't Care
$p = 0 \Rightarrow \dim(x) = 0$
$p = 1 \Rightarrow \dim(x) = 1$
$p = 2 \Rightarrow \dim(x) = 2$.

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

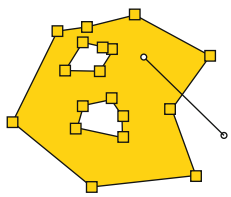
Dimensionally Extended Nine-Intersection Model (DE-9IM), Table 3 Topological predicates and the corresponding pattern matrices after the dimensionally extended NineIntersection model (DE-9IM), OGC (2005)

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)	$\begin{bmatrix} T & * & * \\ * & * & * \\ F & F & * \end{bmatrix}$



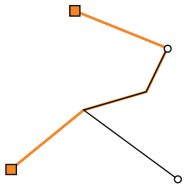
	Interior (B)	**B**oundary (B)	**E**xterior (B)
Interior (A)	–1	–1	2
Boundary (A)	–1	**1/0**	1
Exterior (A)	2	1	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Fig. 4 Example for a "Touches"-relationship between three Polygons A, B and C



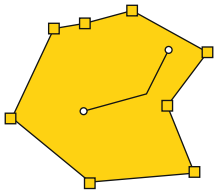
	Interior (B)	**B**oundary (B)	**E**xterior (B)
Interior (A)	1	0	2
Boundary (A)	0	–1	1
Exterior (A)	1	0	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Fig. 5 Example for a "Crosses"-relationship between a Polygon A and a Line B



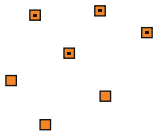
	Interior (B)	Boundary (B)	Exterior (B)
Interior (A)	1	**–1/0**	1
Boundary (A)	**0/–1**	–1	0
Exterior (A)	1	0	2

Dimensionally Extended Nine-Intersection Model (DE-9IM), Fig. 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), Fig. 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), Fig. 8 Example for a "Contains"-relationship between two MultiPoints A and B

can test for a large number of spatial relationships which topological predicate is the appropriate one. With the Relate method defined by OGC (2005), 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 (Egenhofer et al. 1994), composite regions (Clementini et al. 1995) and heterogeneous geometry-collection features (Zhong et al. 2004). 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, e.g., Egenhofer (2002), development of topological relationships for 3D Objects, e.g., Borrmann et al. (2006), and the

extension of the DE-9IM for spatio-temporal databases (Güting and Schneider 2005).

Cross-References

- ▸ Mathematical Foundations of GIS
- ▸ OGC's Open Standards for Geospatial Interoperability
- ▸ Open-Source GIS Libraries
- ▸ Oracle Spatial, Geometries
- ▸ PostGIS
- ▸ Spatiotemporal Query Languages

References

Borrmann A, van Treeck C, Rank E (2006) Towards a 3D spatial query language for building information models. In: Proceedings of the 11th international conference on computing in civil and building engineering (ICCCBE-XI), Montreal

Clementini E, Di Felice PA (1994) Comparison of methods for representing topological relationships. Inf Sci 80:1–34

Clementini E, Di Felice PA (1996) Model for representing topological relationships between complex geometric features in spatial databases. Inf Sci 90(1–4):121–136

Clementini E, Di Felice P, van Oosterom P (1993) A small set of formal topological relationships suitable for end-user interaction. In: Proceedings of the 3rd international symposium on large spatial databases, Singapore, pp 277–295

Clementini E, Di Felice P, Califano G (1995) Composite regions in topological queries. Inf Syst 20:579–594

Davis M, Aquino J (2003) JTS topology suite – technical specifications. Vivid Solutions, Victoria

Egenhofer M (1989) A formal definition of binary topological relationships. In: Litwin W, Schek HJ (eds) Proceedings of the 3rd international conference on foundations of data organization and algorithms (FODO). Lecture notes in computer science, vol 367. Springer, Paris, pp 457–472

Egenhofer M (2002) Toward the semantic geospatial web. In: Voisard A, Chen SC (eds) ACM-GIS 2002. McLean, Virginia, pp 1–4

Egenhofer MJ, Herring J (1990) A mathematical framework for the definition of topological relationships. In: Proceedings of the fourth international symposium on spatial data handling, Zürich, pp 803–813

Egenhofer M, Herring J (1991) Categorizing binary topological relationships between regions, lines, and points in geographic databases. Technical report, Department of Surveying Engineering, University of Maine, Orono

Egenhofer M, Sharma J, Mark D (1993) 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

Egenhofer MJ, Clementini E, di Felice PA (1994) Topological relations between regions with holes. Int J Geogr Inf Syst 8(2):129–142

Güting RH, Schneider M (2005) Moving objects databases. Morgan Kaufmann, San Francisco

ISO/TC211 (ed) (2003) ISO 19107: geographic information – spatial schema

OGC (ed) (2005) OpenGIS® implementation specification for geographic information – Part 1: common architecture (Version 1.1.0)

Zhong Z-N, Jing N, Chen L, Wu Q-Y (2004) Representing topological relationships among heterogeneous geometrycollection features. J Comput Sci Technol Arch Inst Comput Technol Beijing 19(3): 280–289

Directed Acyclic Graphs

- ▸ Bayesian Network Integration with GIS

Direction Relations

- ▸ Directional Relations

Directional Relations

Spiros Skiadopoulos
University of Peloponnese, Tripoli, Greece

Synonyms

Cardinal direction relations; Direction relations; Orientation relations

Definition

Directional relations are qualitative spatial relations that describe how an object or a region is placed relative to other objects or regions. This knowledge is expressed using symbolic (qualitative) and not numerical (quantitative) terms. For

instance, *north*, *southeast*, *front*, and *back-right* are directional relations. Such relations are used to describe and constrain the relative positions of objects or regions and can be used to pose queries such as "Find all objects/regions a, b, and c such that a is *north of* b and b is *southeast of* c."

Historical Background

Qualitative spatial relations (QSRels) approach commonsense knowledge and reasoning about space using symbolic and qualitative rather than numerical and quantitative terms and methods (Hernández 1994) (see also reference to ▸ Qualitative Spatial Reasoning entry). QSRels have found applications in many diverse scientific areas such as geographic information systems, artificial intelligence, databases, and multimedia. Most researchers in QSRels have concentrated on the three main aspects of space, namely, topology, distance, and direction. The uttermost aim in these lines of research is to define more expressive and more intuitive categories of spatial relations and operators. At a second stage, these efforts are accompanied with efficient algorithms for *(a)* the extraction of the relations, *(b)* the automatic processing of the corresponding operators, and *(c)* the processing of queries involving relations and operations.

Specifically, for directional relations, research has provided a plethora of models. Such models have different properties and cover a wide range of applications (desirable properties of directional relations are discussed in Frank 1996). To start with, some directional relations models are able to accommodate *point* objects (or point-based approximations), and some others are able to handle *extended* objects and regions (or extended object or region approximations). Most models use an *absolute* or a *relative* frame of reference but in general can be adopted to an *intrinsic* frame of reference as well (reference to ▸ Reference Frames entry). Additionally, some models express *binary* directional relations (i.e., relations defined on two objects or regions), and some others express *ternary* directional relations (i.e., relations defined on three objects or regions).

Scientific Fundamentals

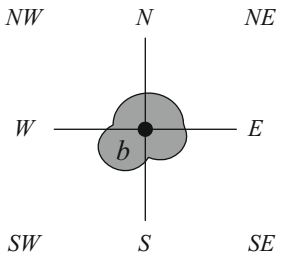
Several models capturing directional relations have been proposed in the literature. Most commonly, a directional relation is a binary relation that describes how a *primary object or region a* is placed relative to a *reference object or region b* (e.g., region a is *north of* b or object u is to the *right* of v). Early models for directional relations, to determine orientation, they:

- use an external coordinate system (i.e., use an absolute frame of reference – see also reference to ▸ Reference Frames entry) and
- are defined on points or point approximations of extended objects or regions (Frank 1996; Hernández 1994; Ligozat 1998).

For instance, in Fig. 1, region b is approximated by its centroid.

Typically, approximation models partition the space around the reference object b into a number of mutually exclusive areas. For instance, the *projection* model partitions the space using lines parallel to the axes (Fig. 1), while the *cone* model partitions the space using lines with an origin angle φ (Fig. 2).

Depending on the adopted model, the directional relation between two objects may be described using different terms. For instance, consider Fig. 3. According to the projection model,

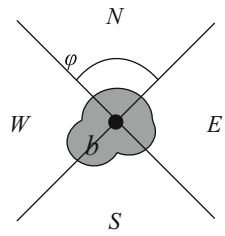


Directional Relations, Fig. 1 Point approximation: projection model space partition

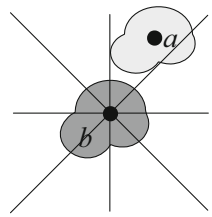
a is *northeast* of b, while according to the cone model, a is *north* of b.

Based on a different frame of reference, the *double-cross calculus* (Freksa 1992) to determine the directional relations of a point a with respect to a point b does not use an external coordinate system but uses an external point c. Figure 4a illustrates how the space around the reference point b is partitioned. Specifically, the reference space is divided into:

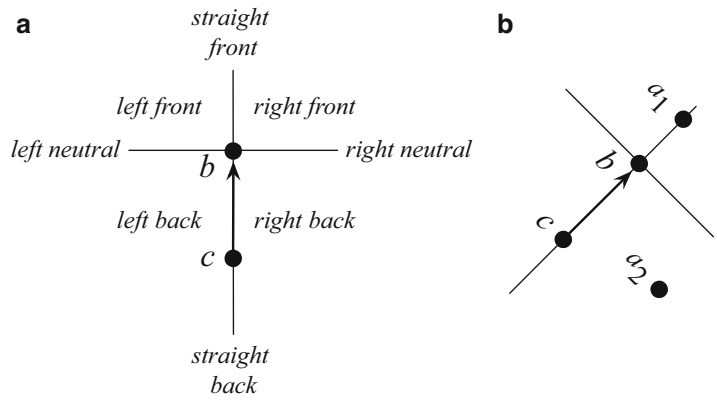
- Four two-dimensional areas (named *left front*, *right front*, *left back*, and *right back*),
- Four semi-lines (named *straight front*, *right neutral*, *straight back*, and *back neutral*),
- A point (named *equal*) that corresponds to b position.



Directional Relations, Fig. 2 Point approximation: cone model space partition



Directional Relations, Fig. 3 Projection and cone models for point approximations may describe different relations for the same regions



Directional Relations, Fig. 4 Determining the directional relation of a with respect to b using an external point c

For instance, in Fig. 4b, using double-cross calculus (Freksa 1992) we have that:

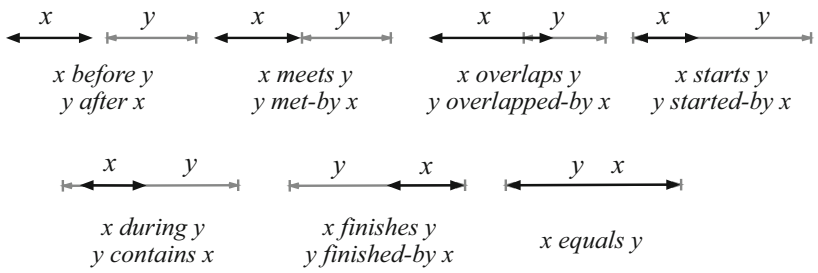
- a_1 is *straight front* of b with respect to c
- a_2 is *right back* of b with respect to c

Inference mechanisms and their computational complexity of the double-cross calculus have been studied in Freksa (1992) and Scivos and Nebel (2001).

In the same spirit, the *order model* of Schlieder (1995) defines relations $+$, $-$, and 0 between three regions a, b, and c. In more detail, $[a, b, c] = +$ iff the point c lies on the left side of the directed line through a and b, $[a, b, c] = -$ iff c lies on the right side of that line and $[a, b, c] = 0$ iff c falls on the line. For example, according to Schlieder (1995) in Fig. 4b, we have $[c, b, a_1] = 0$ and $[c, b, a_2] = +$. Clearly, the relations identified in Schlieder (1995) are more coarse than the relations in Freksa (1992).

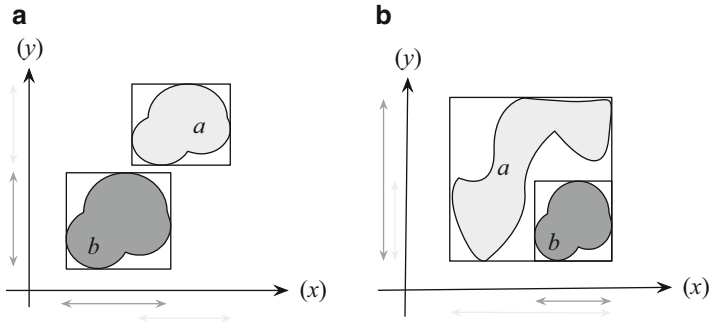
Ligozat (1993) presents the *flip-flop calculus* that extends the model of Freksa (1992) by incorporating into a single relation *(a)*, the relation of a with respect to b using c *and (b)* the relation of a with respect to c using b. This work also devises appropriate refinements to the model and proposes inference methods.

Point-based approximations may be crude (Goyal 2000); thus, later models more finely approximate an object or a region using a representative area (instead of a point) and express directional relations on these approximations (Mukerjee and Joe 1990; Papadias 1994;



Directional Relations, Fig. 5 The 13 relations of interval algebra (Allen 1983)

Directional Relations, Fig. 6 Minimum bounding box approximations

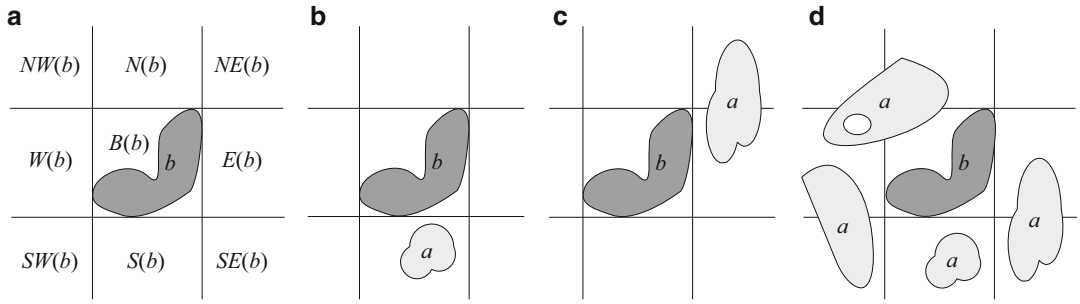


Papadias and Sellis 1994). Most commonly, such methods use the minimum bounding box as a representative area (the minimum bounding box of an object a is the smallest rectangle, aligned with the reference axis, that encloses a) and express the directional relation with respect to the projections on the x- and y-axis. To describe the relation between the projections, the 13 relations of Allen's interval algebra are used (Allen 1983) (see also Fig. 5). For instance, in Fig. 6a, we have a (*overlapped-by, after*) b denoting that the projection on the x-axis (respectively y-axis) of a is overlapped-by (respectively is after) the projection of b. Similarly, in Fig. 6b, we have a (*finished-by, started-by*) b.

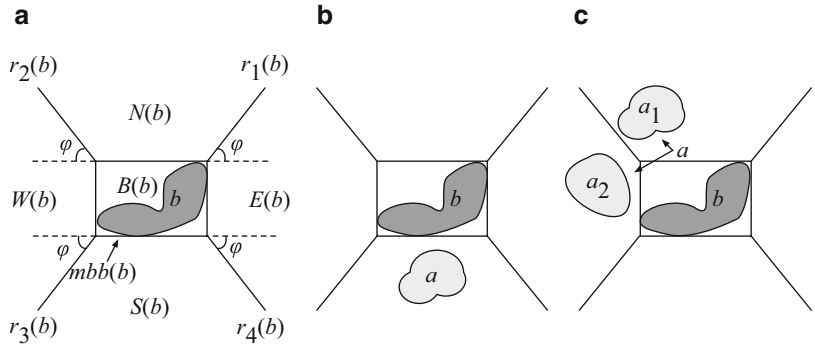
Unfortunately, even with such finer approximations, models that approximate both the primary and the reference object may give misleading directional relations when objects are overlapping, intertwined, or horseshoe shaped (Goyal 2000) (see also Fig. 6b).

Recently, more precise models for directional relations have been proposed. Such models define directions on the exact shape of the primary object and only approximate the reference object (using its minimum bounding box). The *projection-based directional relations* ($\mathcal{P}$DR) model is the first model of this category (Goyal 2000; Skiadopoulos and Koubarakis 2004, 2005). The $\mathcal{P}$DR model partitions the plane around the reference object into nine areas similarly to the projection model (Fig. 7a). These areas correspond to the minimum bounding box (B) and the eight cardinal directions (*N, NE, E*, etc.). Intuitively, the directional relation is characterized by the names of the reference areas occupied by the primary object. For instance, in Fig. 7b, object a is partly *NE* and partly *E* of object b. This is denoted by a *NE*:*E* b. Similarly in Fig. 7c, a *B*:*S*:*SW*:*W*:*NW*:*N*:*E*:*SE* b holds. In total, the $\mathcal{P}$DR model identifies 511 ($= 2^9 - 1$) relations.

Clearly, the $\mathcal{P}$DR model offers a more precise and expressive model than previous approaches that approximate objects using points or rectangles (Goyal 2000). The $\mathcal{P}$DR model adopts a projection-based partition using lines parallel to the axes (Fig. 7). Typically, most people find it more natural to organize the surrounding space using lines with an origin angle similar to the cone model (Fig. 2). This partition of space is adopted by the *cone-based directional relations*



Directional Relations, Fig. 7 Extending the projection model

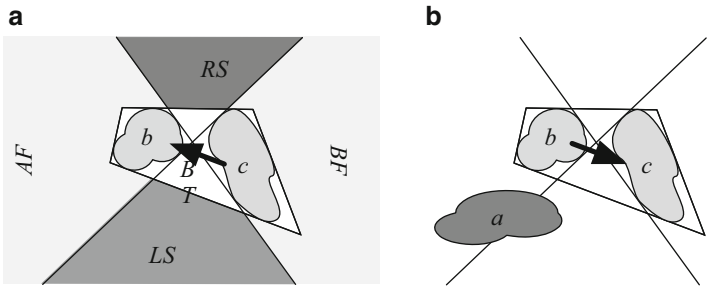


Directional Relations, Fig. 8 Extending the cone model

($\mathcal{C}$DR) model. Similar to the $\mathcal{P}$DR model, the $\mathcal{C}$DR model uses the exact shape of the primary object and only approximates the reference object using its minimum bounding box (Skiadopoulos et al. 2007). Interestingly, for the $\mathcal{C}$DR model, the space around the reference object is partitioned into five areas using a cone-like partition (Fig. 8a). The directional relation is formed by the areas that the primary object falls in. For instance, in Fig. 8b, a is south of b. This is denoted by $a\ S\ b$. Similarly, in Fig. 8c, $a\ B{:}W{:}N\ b$ holds. In total, the $\mathcal{C}$DR model identifies 31 $(= 2^5 - 1)$ relations.

Note that the $\mathcal{P}$DR and $\mathcal{C}$DR models use an absolute frame of reference (see also reference to ▸ Reference Frames entry) where directions were defined using an external coordinate system that determines *north*, *south*, *east*, *west*, etc. Analogously, the above models can be also be defined as intrinsic where the axis and the respective directions *front*, *back*, *right*, *left*, etc. are determined based on the properties of the reference object or region.

In another line of research, directional relations are modeled as ternary relations (Clementini and Billen 2006). Given three objects a, b, and c, the ternary model expresses the directional relation of the primary object a with respect to a reference frame constructed by objects b and c. Specifically, the convex hull and the internal and external tangents of objects b and c divide the space into five areas as in Fig. 9a. These areas correspond to the following directions: right-side (*RS*), before (*BF*), left-side (*LS*), after (*AF*), and between (*BT*). Similar to $\mathcal{P}$DR and $\mathcal{C}$DR, the name of the areas that a falls into determines the relation. For instance, in Fig. 9b, a is *before and to the right side* of b and c. This is denoted by $BF{:}RS\,(a, b, c)$. Notice that, if the order of the reference objects changes, the relation also changes. For instance, in Fig. 9b, $AF{:}LS\,(a, c, b)$ also holds.



Directional Relations, Fig. 9 Ternary directional relations

Directional Relations, Table 1 Operations for directional relations

Model	Computation	Inverse	Composition	Consistency
Point approximations (absolute frame of ref.)	Peuquet and Ci-Xiang (1987)	Ligozat (1998)	Frank (1996) and Ligozat (1998)	Ligozat (1998)
Point approximations (relative frame of ref.)	Peuquet and Ci-Xiang (1987)	Freksa (1992) and Scivos and Nebel (2001)	Freksa (1992) and Scivos and Nebel (2001)	Freksa (1992) and Scivos and Nebel (2001)
Rectangle approximations	Papadias (1994)	Papadias (1994)	Papadias (1994) and Papadias and Sellis (1994)	Papadias (1994) and Papadias and Sellis (1994)
$\mathcal{P}$DR	Skiadopoulos et al. (2005)	Cicerone aand Di Felice (2004)	Skiadopoulos and Koubarakis (2004)	Liu and Li (2011), Liu et al. (2010) and Skiadopoulos and Koubarakis (2005)
$\mathcal{C}$DR	Open problem	Skiadopoulos et al. (2007)	Skiadopoulos et al. (2007)	Open problem
Ternary	Clementini and Billen (2006)	Clementini and Billen (2006)	Clementini et al. (2010)	Open problem

Directional relations have also been studied in the context of images where directionality was encoded using 2D strings (Chang and Jungert 1996) and symbolic arrays (Glasgow and Papadias 1992).

For all the above models of directional relations, research has focused on four interesting operators:

- Efficiently determining the relations that hold between a set of objects.
- Calculating the inverse of a relation.
- Computing the composition of two relations.
- Checking the consistency of a set of relations.

These operators are used as mechanisms that compute and infer directional relations. Such mechanisms are important as they are in the heart of any system that retrieves collections of objects similarly related to each other using spatial relations. Table 1 summarizes current research on the aforementioned problems.

Key Applications

Directional relations intuitively describe the relative position of objects and can be used to constrain and query spatial configurations. This information is very useful in several applications like geographic information systems, spatial databases, spatial arrangement and planning, etc.

Future Directions

There are several open and important problems concerning directional relations. For the models discussed in the previous section, as presented in

Table 1, there are three operators that have not been studied (two for the $\mathcal{C}$DR and one for the ternary model). Another open issue is the integration of directional relations with existing spatial query answering algorithms and data indexing structures (like the R-tree). Finally, with respect to the modeling aspect, even the most expressive directional relations models define directional relations by approximating the reference objects. Currently, there is no intuitive, simple, and easy-to-use model that defines direction relations on the exact shape of the involved objects.

Another interesting topic is the integration of the three main aspects of space, i.e., topology, distance, and direction. Toward this direction, Hernández (1994) aims to combine topological and directional information. Similarly, Clementini et al. (1997) combine directional and distance information. Still, the synthesis of topological, directional, and distance relations in a fully unified framework able to reason and infer knowledge by taking advantage of all available information does not currently exist.

Cross-References

- Projective Relations
- Qualitative Spatial Reasoning
- Reference Frames
- Visibility Relations

References

Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843

Chang S-K, Jungert E (eds) (1996) Symbolic projection for image information retrieval and spatial reasoning. Academic Press, London

Cicerone S, Di Felice P (2004) Cardinal directions between spatial objects: the pairwise-consistency problem. Inf Sci 164(1–4):165–188

Clementini E, Billen R (2006) Modeling and computing ternary projective relations between regions. IEEE Trans Knowl Data Eng 18(6):799–814

Clementini E, Di Felice P, Hernandez D (1997) Qualitative representation of positional information. Artif Intell 95(2):317–356

Clementini E, Skiadopoulos S, Billen R, Tarquini F (2010) A reasoning system of ternary projective relations. IEEE Trans Knowl Data Eng 22(2): 161–178

Frank AU (1996) Qualitative spatial reasoning: cardinal directions as an example. Int J Geogr Inf Sci 10(3):269–290

Freksa C (1992) Using orientation information for qualitative spatial reasoning. In: Proceedings of COSIT'92, Piza. LNCS, vol 639, pp 162–178. http://dblp.uni-trier.de/rec/bibtex/conf/gis/Freksa92

Glasgow J, Papadias D (1992) Computational imagery. Cogn Sci 16:355–394

Goyal R (2000) Similarity assessment for cardinal directions between extended spatial objects. Ph.D. thesis, Department of Spatial Information Science and Engineering, University of Maine

Hernández D (1994) Qualitative representation of spatial knowledge. LNCS, vol 804. Springer, Berlin

Ligozat G (1993) Qualitative triangulation for spatial reasoning. In: Proceedings of COSIT'93, Elba Island. LNCS, vol 716, pp 54–68

Ligozat G (1998) Reasoning about cardinal directions. J Vis Lang Comput 9:23–44

Liu W, Li S (2011) Reasoning about cardinal directions between extended objects: the NP-hardness result. Artif Intell 175(18):2155–2169

Liu W, Zhang X, Li S, Ying M (2010) Reasoning about cardinal directions between extended objects. Artif Intell 174(12–13):951–983

Mukerjee A, Joe G (1990) A qualitative model for space. In: Proceedings of AAAI'90, Boston, pp 721–727. http://dblp.uni-trier.de/rec/bibtex/conf/aaai/MukerjeeJ90

Papadias D (1994) Relation-based representation of spatial knowledge. Ph.D. thesis, Department of Electrical and Computer Engineering, National Technical University of Athens

Papadias D, Sellis TK (1994) Qualitative representation of spatial knowledge in two-dimensional space. VLDB J 3(4):479–516

Peuquet DJ, Ci-Xiang Z (1987) An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. Pattern Recognit 20(1):65–74

Schlieder C (1995) Reasoning about ordering. In: Proceedings of COSIT'95, Semmering. LNSC, vol 988, pp 341–349

Scivos A, Nebel B (2001) Double-crossing: decidability and computational complexity of a qualitative calculus for navigation. In: Proceedings of COSIT'01, Morro Bay. LNCS, vol 2205, pp 431–446. http://dblp.uni-trier.de/rec/bibtex/conf/cosit/ScivosN01

Skiadopoulos S, Koubarakis M (2004) Composing cardinal direction relations. Artif Intell 152(2):143–171

Skiadopoulos S, Koubarakis M (2005) On the consistency of cardinal directions constraints. Artif Intell 163(1):91–135

Skiadopoulos S, Giannoukos C, Sarkas N, Vassiliadis P, Sellis T, Koubarakis M (2005) Computing and managing cardinal direction relations. IEEE Trans Knowl Data Eng 17(12):1610–1623

Skiadopoulos S, Sarkas N, Sellis T, Koubarakis M (2007) A family of directional relation models for extended objects. IEEE Trans Knowl Data Eng 19(8): 1116–1130

Directory Rectangles

▸ R*-Tree

Dirichlet Tessellation

▸ Voronoi Diagram

Disaster Risks

▸ Climate Extremes and Informing Adaptation

Discord or Non-specificity in Spatial Data

▸ Uncertainty, Semantic

Discretization of Quantitative Attributes

▸ Geosensor Networks, Qualitative Monitoring of Dynamic Fields

Disease Mapping

▸ Public Health and Spatial Modeling

Disk Page

▸ Indexing Schemes for Multidimensional 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

D

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:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_3)^2 + \cdots + (a_n - b_n)} \quad (1)$$

where a and b are two points in some spatial space and n is the dimension.

The Manhattan distance can be mathematically described as:

$$|x_1 - x2| + |y1 - y2| \quad (2)$$

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 Networks, Estimating Continuous Phenonena

Distributed Caching

- ▸ OLAP Results, Distributed Caching

Distributed Computing

- ▸ Clustering of Geospatial Big Data in a Distributed Environment
- ▸ 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, College of Sciences, George Mason University, Fairfax, VA, USA

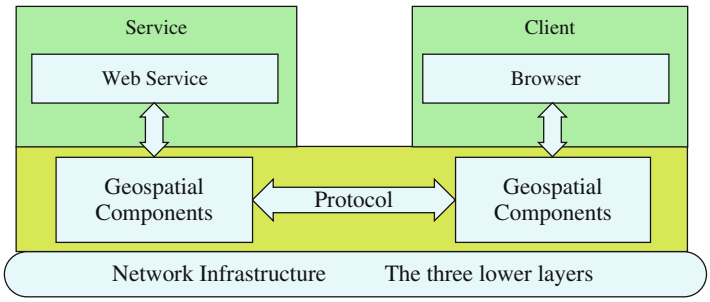
Synonyms

DGC; Distributed computing; Distributed geocomputation; Distributed geospatial information processing; Parallel computing

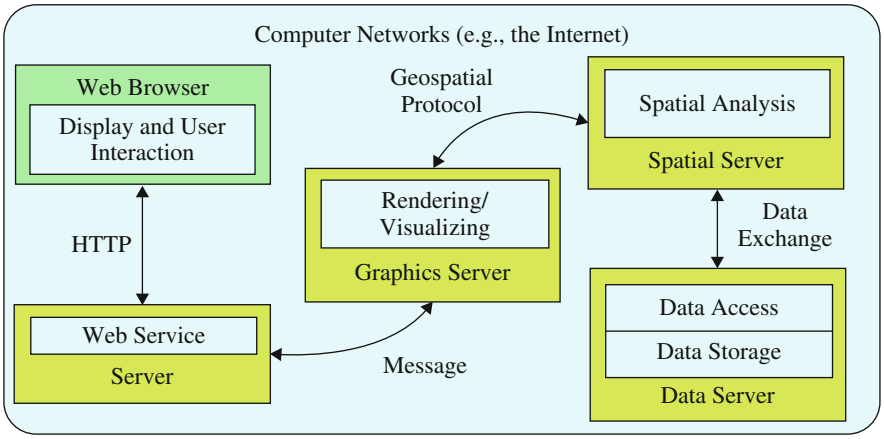
Definition

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 (Yang et al. 2006a): 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



Distributed Geospatial Computing (DGC), Fig. 1 Distributed geospatial computing (DGC) within computer network architecture



Distributed Geospatial Computing (DGC), Fig. 2 Distributions of DGC components onto servers within a computer network

querying language (SQL) (ISO/IEC 2005), the geospatial protocol can be arc extensible markup language (ArcXML) (ESRI 2002), the message can be transmitted through pipe, and the client to web service can be HTTP (Yang et al. 2006b).

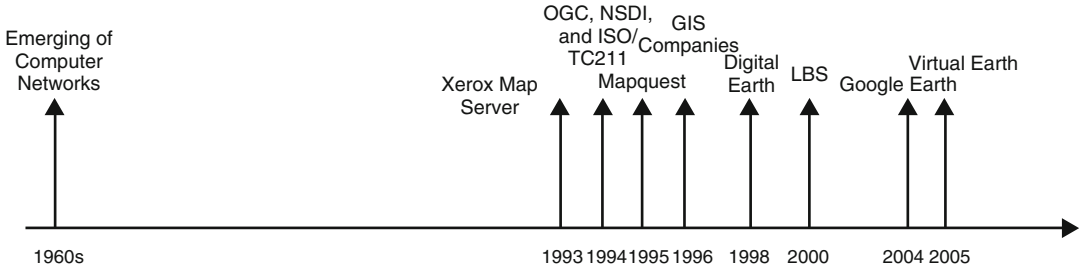
This distribution of geospatial computing components matches the needs to integrate the legacy and future components of geospatial computing deployed at different computers and hosted by different organizations (Yang and Tao 2005).

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 (Plewe 1997). In 1994, the Federal Geographic Data Committee (FGDC) (1994) was established to share geospatial computing across distributed platform, and the Open Geospatial Consortium (OGC) (1994) and the International Standards Organization/Technical Committee 211 (ISO/TC211) (1994) 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



Distributed Geospatial Computing (DGC), Fig. 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

participate in the DGC effort by fully implementing geospatial computing components in the distributed environment (Plewe 1997; Peng and Tsou 2003).

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 (Tao 2006).

Accompanying these events, which have profoundly impacted on DGC, many scholars also addressed issues on how to distribute geospatial computing (Yang et al. 2005a), improve performance of DGC (Yang et al. 2005b), parallelize geospatial computing (Healey et al. 1998), and leverage grid platforms and agent-based environments (Nolan 2003) for distributed geospatial computing. Several centers/labs, such as the Joint Center for Intelligent Spatial Computing (CISC) (Joint Center for Intelligent Spatial Computing at George Mason University 2005) and Pervasive Technology Lab (PTL) (Pervasive Technology Labs at Indiana University 2004) are established to address the research needs in this field.

Scientific Fundamentals

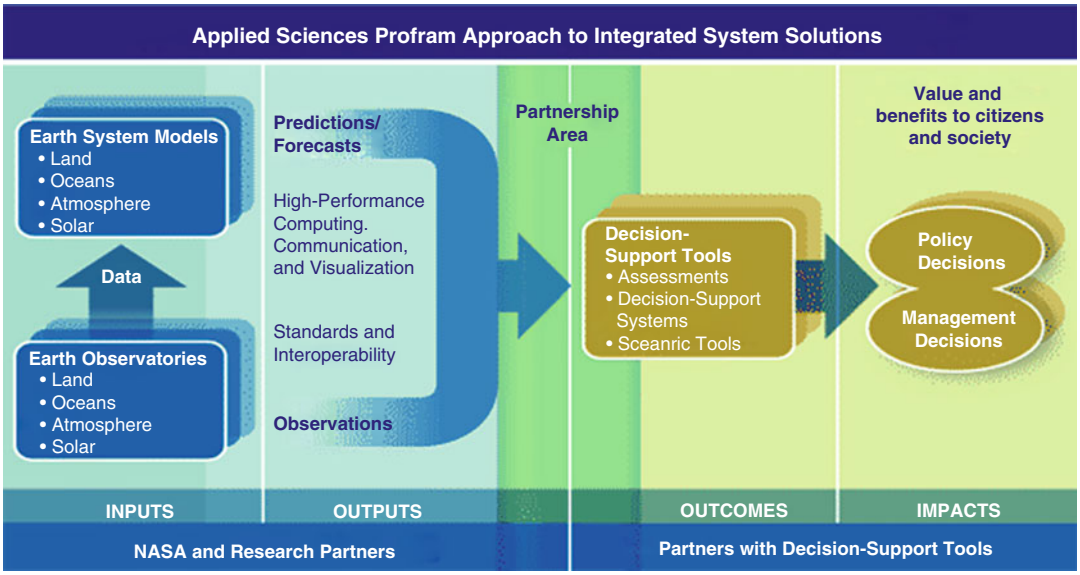
The scientific fundamentals of DGC rely on geometry, topology, statistics, and cartography principles to design and implement algorithms. Geospatial recognition and physical science provides conceptualization and model foundations for DGC. Earth sciences provide support for the DGC application design. 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.



Distributed Geospatial Computing (DGC), Fig. 4 DGC is embedded in the integrated system solutions (Birk et al. 2006)

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 (Executive Office of the President 1994).

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 (Mayer et al. 2004).

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 (Cervone et al. 2005).

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 (Birk et al. 2006) and GEO identified 9 application areas of interest at the global level (GEO 2005). Figure 4 illustrates the integration of earth observations, earth system models, with decision support tools to support decision or policy making (Birk et al. 2006). 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-Based Spatial Information Retrieval
▸ Internet GIS

References

Birk R, Frederick M, Dewayne LC, Lapenta MW (2006) NASA's applied sciences program: transforming research results into operational success. Earth Imaging J 3(3):18–23

Cervone G, Singh RP, Kafatos M, Yu C (2005) Wavelet maxima curves of surface latent heat flux anomalies associated with Indian earthquakes. Nat Hazards Earth Syst Sci 5:87–99

ESRI (2002) ArcXML programmer's reference guide, ArcIMS 4

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

FGDC (1994) This website has information for clearinghouse, NSDI, GOS portal, and funding opportunities. http://www.fgdc.gov/. Accessed 20 Aug 2007

GEO (2005) The global earth observation system of systems (GEOSS) 10-year implementation plan. http://earthobservations.org/docs/10Year%20Implementation%20Plan.pdf. Accessed 20 Aug 2007

Healey R, Dowers S, Gittings B, Mineter M (1998) Parallel processing algorithms for GIS. Taylor & Francis, Bristol

International Standards Organization (1994) Technical Committee 211: GeoInformation, Geomatics. http://www.isotc211.org/

ISO/IEC 13249 (2005) Information Technology Database languages-SQL Multimedia and Application Packages, Part 3: Spatial

Joint Center for Intelligent Spatial Computing at George Mason University (2005) http://landscan.scs.gmu.edu:8080/. Accessed 20 Aug 2007

Mayer L, Barbor K, Boudreau P, Chance T, Fletcher C, Greening H, Li R, Mason C, Metcalf K, Snow-Cotter S, Wright D (2004) A geospatial framework for the coastal zone: national needs for coastal mapping and charting. National Academies, Washington, DC

Nolan J (2003) An agent-based architecture for distributed imagery and geospatial computing. PhD thesis, George Mason University

Open Geospatial Consortium (OGC) (1994) This website has information on WMS, WFS, WCS, CS-W, and other OGC specifications. http://www.opengeospatial.org

Peng ZR, Tsou MH (2003) Internet GIS: distributed geographic information services for the internet and wireless networks. Wiley, Hoboken

Pervasive Technology Labs at Indiana University (2004) http://grids.ucs.indiana.edu/ptliupages/. Accessed 20 Aug 2007

Plewe B (1997) GIS online: information retrieval, mapping, and the internet. OnWord, Santa Fe

Tao V (2006) Microsoft virtual earth: new horizons in on-line mapping. Paper presented at GeoInformatics'2006, Wuhan, 28–29 Oct 2006

Yang C, Tao V (2005) Distributed geospatial information service In: Rana S, Sharma J (eds) Frontiers of geographic information technology. Springer, London, pp 113–130

Yang C, Wong D, Li B (2005) Computing and computational issues of distributed GIS. Int J Geogr Inf Sci 10(1):1–3

Yang C, Wong D, Yang R, Kafatos M, Li Q (2005) WebGIS performance improving techniques. Int J Geogr Inf Sci 19:319–342

Yang C, Wong D, Kafatos M, Yang R (2006) Implementing computing techniques to accelerate network GIS. In: Li D, Xia L (eds) Proceedings of SPIE, vol 6418, 64181C, Geoinformatics 2006: GNSS and integrated geospatial applications
Yang C, Wong D, Kafatos M, Yang R (2006) 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

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

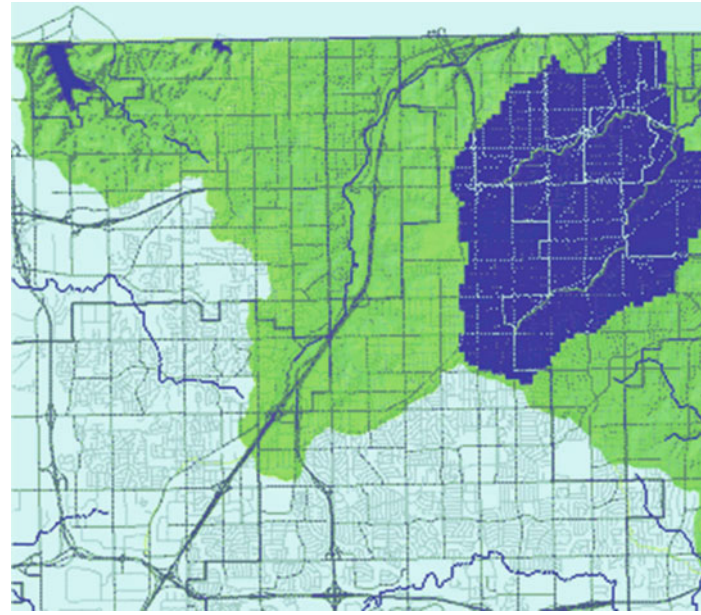
GIS-based hydrology; Hydrogeology; Hydrology; Spatial hydrologic modeling

Definition

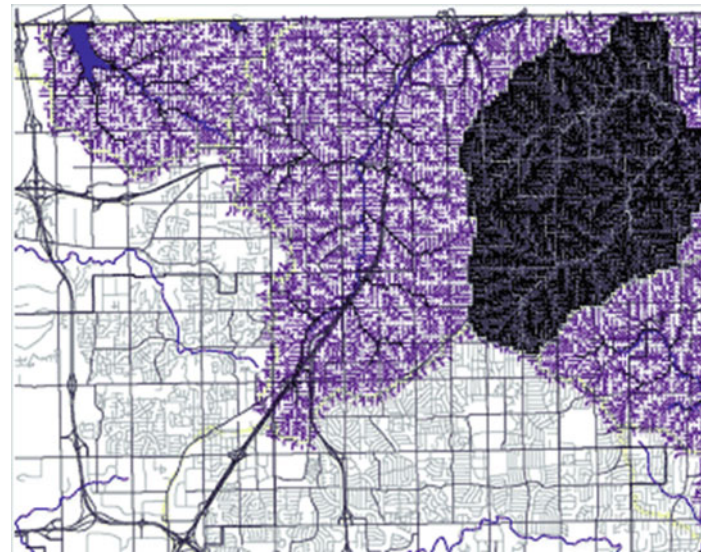
Distributed hydrologic modeling within a GIS framework is the use of parameter maps derived from geospatial data to simulate hydrologic processes. Distributed models of hydrologic processes rely on representing characteristics of the earth's surface that affect components of the water balance. Capturing 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 System s 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



Distributed Hydrologic Modeling, Fig. 1 Subwatershed model representation



Distributed Hydrologic Modeling, Fig. 2 Gridded representation with flow direction indicating the drainage network

application of GIS for lumped and distributed hydrologic modeling may be found in Bedient et al. (2007).

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 solu-

tion. 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,

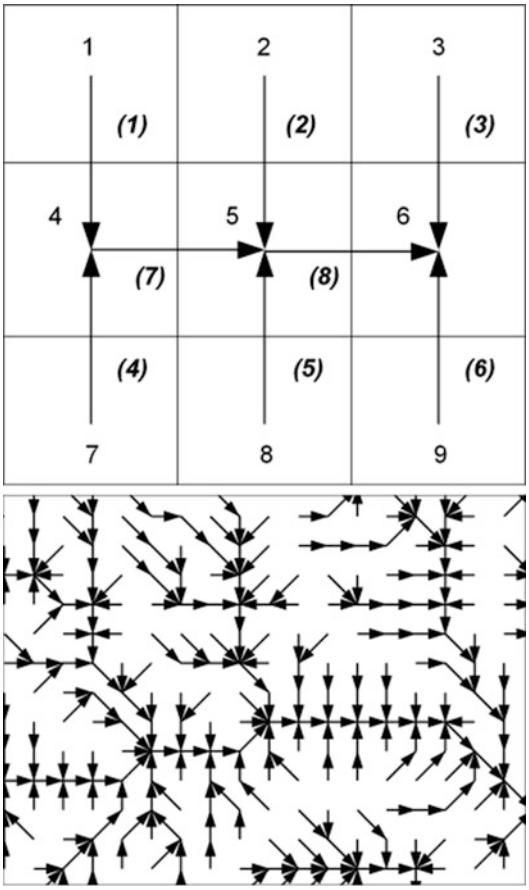
$$\frac{\partial h}{\partial t} + \frac{\partial q}{\partial x} = R - I \, . \tag{1}$$

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, Fig. 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.

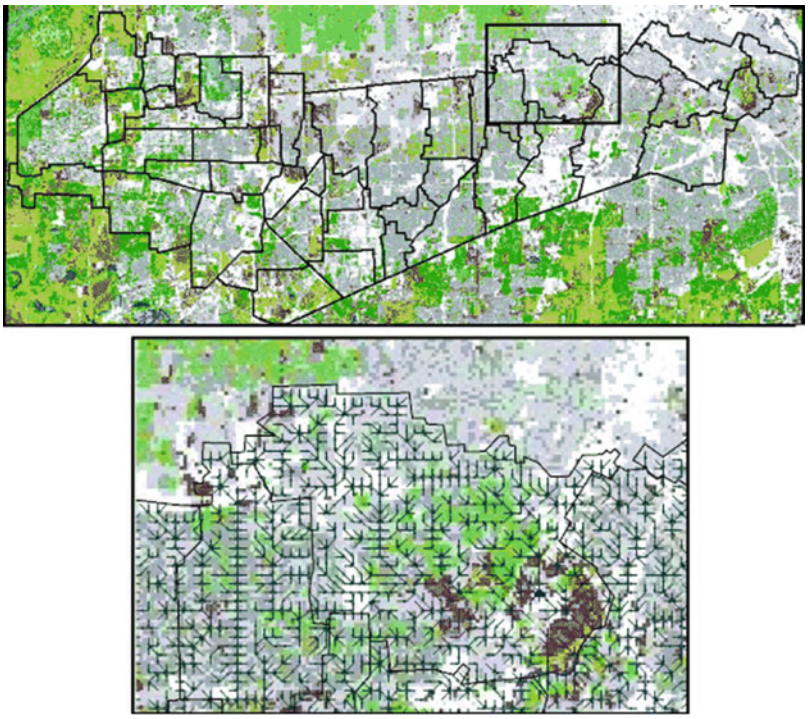
1. Soils/geologic material maps for estimating infiltration

   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 highresolution 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



Distributed Hydrologic Modeling, Fig. 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*)

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 manually 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,

$$\{R^*\} = \gamma \cdot \{R\} \tag{3}$$

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

References

Bedient PB, Huber WC, Vieux BE (2007) Hydrology and floodplain analysis, 4th edn. Prentice Hall, Upper Saddle River

Moore ID (1996) Hydrologic modeling and GIS. In: Goodchild MF, Steyaert LT, Parks BO, Johnston C, Maidment DR, Crane MP, Glendinning S (eds) GIS and environmental modeling: progress and research issues. GIS World Books, Fort Collins, pp 143–148

Vieux BE (2004) Distributed hydrologic modeling using GIS. Water science technology series, vol 48, 2nd edn. Kluwer Academic, Norwell, p 289. ISBN:1-4020-2459-2. CD-ROM including model software and documentation

Vieux BE, Bedient PB (2004) Assessing urban hydrologic prediction accuracy through event reconstruction. Spec Issue Urban Hydrol J Hydrol 299(3–4):217–236

Vieux BE, Moreda FG (2003) Ordered physics-based parameter adjustment of a distributed model. In: Duan Q, Sorooshian S, Gupta HV, Rousseau AN, Turcotte R (eds) Advances in calibration of watershed models. Water science and application series, vol 6. American Geophysical Union, Washington, DC, pp 267–281

Recommended Reading

DeBarry PA, Garbrecht J, Garcia L, Johnson LE, Jorgeson J, Krysanova V, Leavesley G, Maidment D, Nelson EJ, Ogden FL, Olivera F, Quimpo RG, Seybert TA, Sloan WT, Burrows D, Engman ET (1999) GIS modules and distributed models of the watershed. American Society of Civil Engineers Water Resources Division – Surface Water Hydrology Committee, Reston, 120pp

Gurnell AM, Montgomery DR (2000) Hydrological applications of GIS. Wiley, New York
Jain MK, Kothyari UC, Ranga Raju KG (2004) A GIS based distributed rainfall-runoff model. J Hydrol 299(1–2):107–135
Maidment DR, Djokic D (2000) Hydrologic and hydraulic modeling support with geographic information systems. ESRI Press, Redlands
Moore ID, Grayson RB, Ladson AR (1991) Digital terrain modeling: a review of hydrological, geomorphological and biological applications. J Hydrol Process 5(3–30)
Newell CJ, Rifai HS, Bedient PB (1992) Characterization of non-point sources and loadings to Galveston Bay. Galveston Bay National Estuary Program (GBNEP-15), Clear Lake, 33pp
Olivera F (2001) Extracting hydrologic information from spatial data for HMS modeling. ASCE J Hydrol Eng 6(6):524–530
Olivera F, Maidment D (1999) GIS-based spatially distributed model for runoff routing. Water Resour Res 35(4):1155–1164
Maidment DR, Morehouse S (eds) (2002) Arc hydro: GIS for water resources. ESRI Press, 218pp. ISBN:1-5894-8034-1
Shamsi U (2002) GIS tools for water, wastewater, and stormwater systems. American Society of Civil Engineers (ASCE Press), Reston, p 392. ISBN:0-7844-0573-5
Singh VP, Fiorentino M (1996) Geographical information systems in hydrology, water science and technology library, vol 26. Kluwer Academic, Norwell. ISBN:0-7923-4226-7
Safiolea E, Bedient PB, Vieux BE (2005) Assessment of the relative hydrologic effects of land use change and subsidence using distributed modeling. In: Moglen GE (ed) ASCE, engineering, ecological, and economic challenges watershed, Williamsburg, pp 178, 87
Tate E, Maidment D, Olivera F, Anderson D (2002) Creating a Terrain model for floodplain mapping, ASCE. J Hydrol Eng 7(2):100–108
Vieux BE (1993) DEM aggregation and smoothing effects on surface runoff modeling. J Comput Civ Eng 7(3):310–338
Wilson JP (ed) Gallant JC (2000) Terrain analysis: principles and applications. Wiley, New York, p 512. ISBN:0-471-32188-5

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)

Doughnut Hole Detection

▸ Ring-Shaped Hotspot Detection

Downscaling

▸ Aggregate Data: Geostatistical Solutions for Reconstructing Attribute Surfaces

Driving Direction

▸ Fastest-Path Computation

Dual Space-Time Representation

▸ Indexing Schemes for Multidimensional Moving Objects

Dynamic Generalization

▸ Generalization, On-the-Fly

Dynamic Nearest Neighbor Queries in Euclidean Space

Sarana Nutanong[1], Mohammed Eunus Ali[2], Egemen Tanin[3], and Kyriakos Mouratidis[4]
[1]City University of Hong Kong, Hong Kong, China
[2]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh
[3]University of Melbourne, Melbourne, VIC, Australia
[4]Singapore Management University, Singapore, Singapore

Synonyms

Nearest neighbor monitoring; Temporal nearest neighbor query

Definition

Given a query point **q** and a set $\mathcal{D}$ of data points, a nearest neighbor (NN) query returns the data point **p** in $\mathcal{D}$ that minimizes the distance $\text{DIST}(\boldsymbol{q},\boldsymbol{p})$, where the distance function DIST(,) is the L_2 norm. One important variant of this query type is *kNN query*, which returns k data points with the minimum distances. When taking the temporal dimension into account, the kNN query result may change over a period of time due to changes in locations of the query point and/or data points. Formally, the k-nearest neighbor (kNN) query is defined as follows.

Definition 1 (k-Nearest Neighbor (kNN) Query) Given a set $\mathcal{D}$ of data objects and a query point **q**, the kNN query finds a set $\mathcal{R}$ of objects such that: (i) $\mathcal{R}$ contains k objects from $\mathcal{D}$. (ii) for any object $\mathbf{x} \in \mathcal{R}$ and object $\mathbf{y} \in (\mathcal{D} - \mathcal{R})$, $\text{DIST}(\boldsymbol{q}, \boldsymbol{x}) \leq \text{DIST}(\boldsymbol{q},\boldsymbol{y})$.

A *dynamic kNN query in Euclidean space* returns kNN query results over a period of time in a dynamically changing environment.

Figure 1 provides an example of a dynamic 1NN query with a moving query point and a static dataset. The example shows that the query point **q** moves from left to right in three successive snapshots t_1, t_2, and t_3, where **a** is the nearest neighbor at times t_1 and t_2, and the result is updated to **c** at t_3. A straightforward approach to processing a dynamic kNN query is to issue multiple kNN queries repetitively. However, the result accuracy of this approach highly depends on the query frequency, and a higher query frequency incurs a greater query processing cost. In this example, if we assume that a kNN query is issued at t_1, at t_2, and at t_3, then there is a time period between t_2 and t_3 in which the result is obsolete.

Historical Background

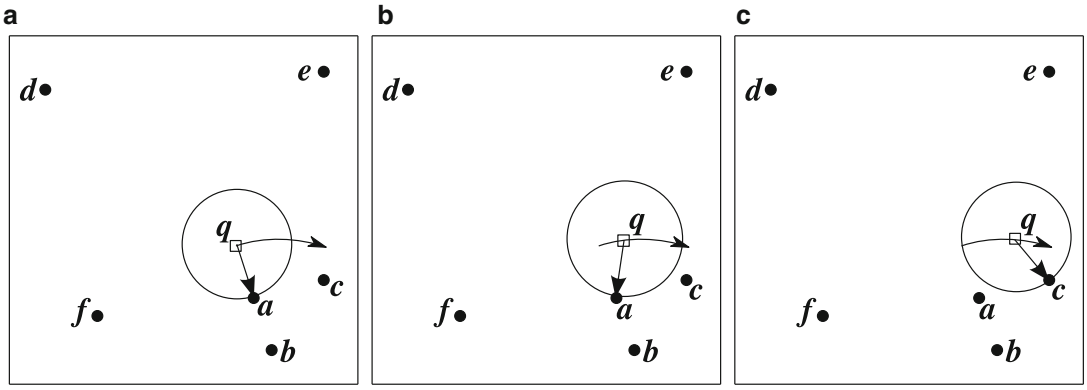
The study of continuous kNN queries is generally concerned with deriving query processing techniques to reduce the query processing cost without sacrificing the result accuracy. Since the early 2000s, considerable research attention has been given techniques to process variants of continuous kNN queries for moving query points and moving data objects.

Scientific Fundamentals

Continuous kNN processing relies on the same fundamental concept as other continuous query types, i.e., figuring out the conditions that may invalidate the current result set. For a continuous kNN query, the result set can be invalidated by having an object **y** that is not included in the current result set $\mathcal{R}$ coming closer to the query point **q** than an object in $\mathcal{R}$. The manner in which **y** can come closer to the query point than an object in $\mathcal{R}$ depends on whether the query point and/or the data objects are assumed to be moving.

Moving Query over Static Data Objects

This variant is also known as the *moving kNN* (MkNN) query. In this case, changes in the result set are caused by the query point **q** moving closer to an object **y** that is not in the result set $\mathcal{R}$



Dynamic Nearest Neighbor Queries in Euclidean Space, Fig. 1 Continuous k-nearest neighbor query with a moving query point **q** and a static dataset {**a**, **b**, **c**, **d**, **e**, **f**} in Euclidean space, where for $k = 1$, **a** is the nearest neighbor at times t_1 and t_2, and **b** is the nearest neighbor at time t_3. (**a**) Time t_1: NN(**q**) = **a**. (**b**) Time t_2: NN(**q**) = **a**. (**c**) Time t_3: NN(**q**) = **c**

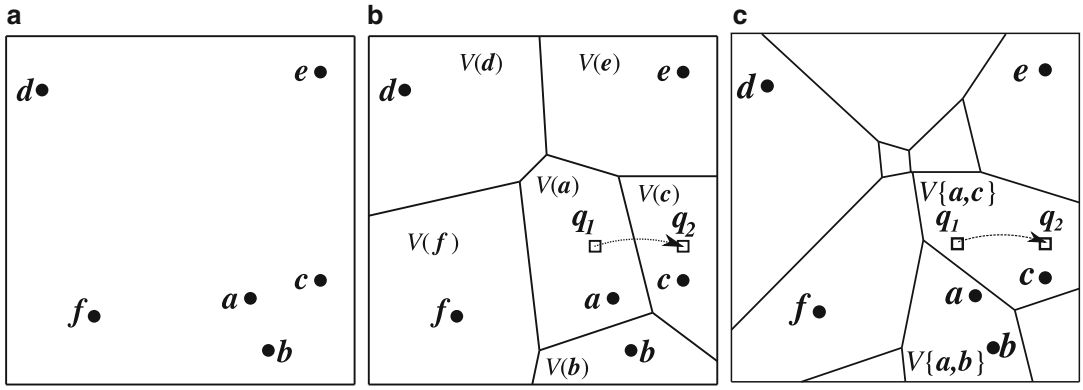
than any object **x** in $\mathcal{R}$. One popular processing approach for this variant is identifying all possible pairs of **x** and **y** in order to compute the boundaries $B_{x,y}$, where $B_{x,y}$ is formally defined as a set of points **p** such that DIST($\boldsymbol{p}$, $\boldsymbol{x}$) is equal to DIST($\boldsymbol{p}$,$\boldsymbol{y}$). The area inside these boundaries is also known as a *safe region*. As long as the movement of **q** is confined within the safe region, $\mathcal{R}$ remains valid.

Precomputing safe regions. A classic example of safe region-based techniques is the *Voronoi diagram* (Aurenhammer 1991; Okabe et al. 1992). The Voronoi diagram is a well-known space decomposition technique determined by distances to a given discrete set of objects, typically a set of points. Figure 2b shows a Voronoi diagram of six data objects {**a**, **b**, **c**, **d**, **e**, **f**}. Assume that the query point is originally at $\mathbf{q_1}$, the safe region is the area confined by five boundaries, $B_{a,b}$, $B_{a,c}$, $B_{a,d}$, $B_{a,e}$, and $B_{a,f}$. As long as the query point does not cross any of these boundaries, **a** remains the first NN. As exemplified in Fig. 2c, the Voronoi diagram can be generalized to the *kth-order Voronoi diagram (kVD)*, which can be used to help process kNN queries for any given location in the data space.

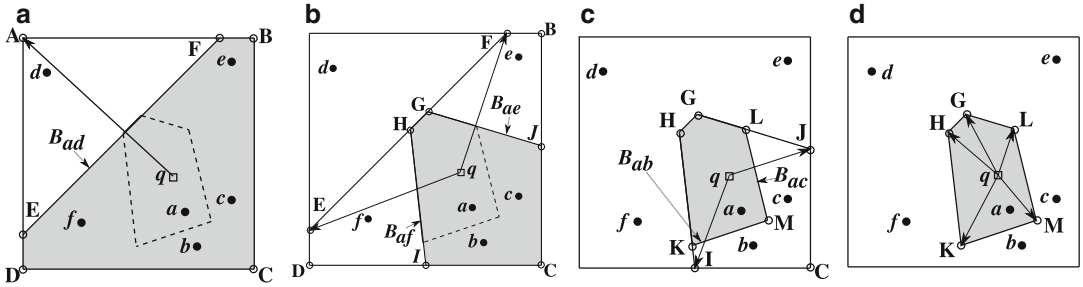
Processing an MkNN query using a kVD can be done by identifying the Voronoi cell in which the query point **q** is currently residing and monitoring the location of **q** constantly. The result set is updated only when **q** crosses a boundary. The main benefit of this approach is the query processing costs which are logarithmic with respect to the number of data objects for the initial lookup and constant for safe region checking (Aurenhammer 1991; Okabe et al. 1992).

The main drawback of using a kVD to process moving kNN queries is that the technique requires an evaluation of Voronoi cells of the entire dataset. This is considered undesirable especially when the movement of the query point **q** is confined in a small area with respect to the entire data space. Furthermore, one may require different kVDs for different needs. For example, a driver may need to find a gas station with a restroom facility, while another driver needs one with a special type of fuel. Precomputing kVDs for all possible scenarios is impractical.

Predefined query trajectories. When the trajectory is known in advance, the MkNN query processing problem can be simplified to identification of the point along the trajectory at which the kNN result set changes, i.e., where the trajectory intersects a safe region boundary. Tao and Papadias (2002) proposed the *time-parameterized kNN (TPkNN) query*. Assuming a linear trajectory of the query point, a TPkNN query finds (i) the current kNN set, (ii) a position on the trajectory where the kNN result set changes (the *influence point*), and (iii) the object



Dynamic Nearest Neighbor Queries in Euclidean Space, Fig. 2 The Voronoi diagram and its generalizations. (**a**) Point set $\mathcal{D}$, {**a**, **b**, **c**, **d**, **e**, **f**}. (**b**) Voronoi diagram of $\mathcal{D}$. (**c**) Second-order Voronoi diagram of $\mathcal{D}$



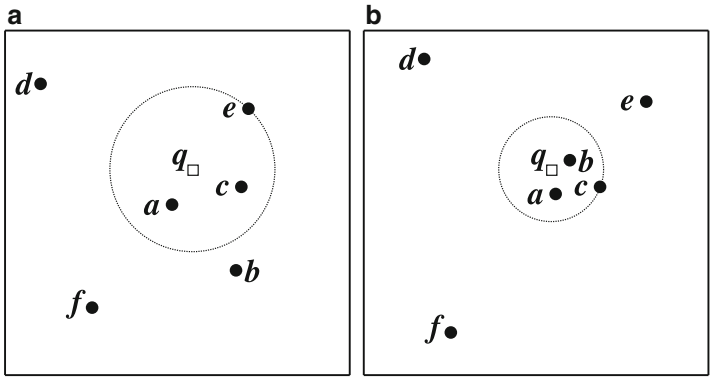
Dynamic Nearest Neighbor Queries in Euclidean Space, Fig. 3 Locally compute a kVD cell ($k = 1$). (**a**) Step 1. (**b**) Steps 2 and 3. (**c**) Steps 4 and 5. (**d**) Final steps

that causes the change (the *influence object*). Finding the influence object is done by ranking candidate influence objects according to how early their corresponding influence points appear on the trajectory.

Another well-known method to handle an MkNN query with a predefined query trajectory is to use the *continuous* k*NN* (CkNN) query (Tao et al. 2002). CkNN query splits the query trajectory into segments where each segment corresponds to a particular kNN result set. This is done by identifying the influence points along the query trajectory. The main difference between CkNN and TPkNN is that CkNN obtains all kNN result sets along a given trajectory, but TPkNN provides only the segment corresponding to the current kNN result set.

Unknown query trajectories. One method to handle an MkNN query for an unknown trajectory is to locally construct the Voronoi cell that currently contains the query point. Since only one Voronoi cell is needed at a time, this method does not suffer from the same drawback as the Voronoi diagram method. Zhang et al. (2003) proposed a method that executes multiple instances of TP-kNN queries to discover all possible influence objects and cell boundaries around the query point (as illustrated in Fig. 3). Due to the convexity property of Voronoi cells generated from a set of data points, it is guaranteed that all boundaries are discovered when all safe region corners share the same kNN result set as the query point.

Result caching. Based on the principle of spatial locality of references, we can cache data objects around the query point and attempt to reproduce query results using the cached data objects as the query point moves (Nutanong et al. 2010; Song and Roussopoulos 2001). Once the cached



Dynamic Nearest Neighbor Queries in Euclidean Space, Fig. 4 Continuous k-nearest neighbor query with a static query point $\mathbf{q}$ and a moving dataset $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}\}$ in Euclidean space. (**a**) Time t_1: $3\text{NN}(\mathbf{q}) = \{\mathbf{a}, \mathbf{c}, \mathbf{e}\}$. (**b**) Time t_2: $3\text{NN}(\mathbf{q}) = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$

data can no longer produce accurate query results, the cache is updated. One method to utilize cached data for processing an MkNN query is to apply the sampling-based approach (Song and Roussopoulos 2001), i.e., periodically ranking the cached data objects according to the distance from the query point. The cache is updated when the correctness of the kNN result set cannot be guaranteed.

When truly continuous query results are required, cached objects can also be used to build a safe region. Nutanong et al. (2010) proposed a method which incrementally maintains two types of safe regions: (i) one that keeps the rank of cached data object constant and (ii) one that ensures the validity of the kNN obtained from the cache. Li et al. (2014) propose an incremental method to compute and to update a set of data objects that may invalidate the current kNN result set. The kNN result set is guaranteed to be valid as long as the query point is closer to kNN than any of the invalidating object. This method can also be regarded as a safe region-based method in the sense that we can compute the boundary between each pair of the invalidating object and its corresponding object in the kNN set. The region that is enclosed by these boundaries is a safe region.

The main difference between the sampling-based method (Song and Roussopoulos 2001) and the safe region-based method (Li et al. 2014; Nutanong et al. 2010) can be described as follows. The sampling-based method (Song and Roussopoulos 2001) does not produce truly continuous query answers; its main objective is to reduce the cost of each kNN query execution in order to accommodate a higher query frequency. On the other hand, the safe region-based method (Li et al. 2014; Nutanong et al. 2010) can produce continuous query results by keeping track of locations at which the query point crosses a safe region boundary.

Static Query over Moving Data Objects

When the query point is static and data objects are moving, processing a continuous kNN query involves keeping track of data objects moving in and out of the kNN result set. Figure 4 shows how the kNN result set changes from $\{\mathbf{a}, \mathbf{c}, \mathbf{e}\}$ to $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ due to the movements of data objects in the dataset.

Predefined object trajectories. When object trajectories are known in advance, processing a continuous kNN query involves identifying object trajectories that may involve in the result set and ruling out those that are guaranteed to be outside the result set with respect to a given query location and time of interest. To accommodate querying in the temporal dimension, a time-parameterized data structure (Tao et al. 2003; Saltenis and Jensen 2002) is often used to index object trajectories. In this way, the location of each data object is represented as a function of time (i.e., its initial location and its velocity vector). Each object updates its location function only when it no longer accurately describes its movement or when the location function is older than a predefined threshold.

A notable example of techniques which utilize time parameterization is the *extended time parameterization (ETP)* algorithm, proposed by Iwerks et al. (2006). The ETP algorithm provides support for moving objects by extending the TP-kNN algorithm (Tao and Papadias 2002). Iwerks et al. (2006) also formulated an approach to processing a kNN query on moving objects by continuous evaluation of a range query. Their proposed approach is based on the observation that a continuous fixed-range query is easier to process than a continuous kNN query with moving objects. By imposing a condition that the scope of the range query must include at least $k + 1$ data objects, the k nearest objects can be evaluated based on only objects within the scope. The query scope is allowed to expand and to contract according to the current density of objects around the query point.

Unknown object trajectories. Unknown object trajectories are handled similarly to the case of an unknown query trajectory. Specifically, a safe region is associated with every data object. As long as all data objects remain inside their respective safe regions, the current kNN result set is guaranteed to be up to date. Mouratidis et al. (2005b) proposed a threshold-based approach to monitoring the k nearest objects in a setting for moving objects. Each monitored object is associated with a range of safe distances from the query point. An object cannot influence the query answer as long as it remains within the range of safe distances. Hu et al. (2005) proposed a safe region-based technique for *static* window and kNN queries on moving objects. Each moving object maintains its own safe region and only reports its new location if it may affect any query result.

Moving Query over Moving Data Objects

When both query and data objects are allowed to move, processing continuous kNN queries involves monitoring the locations of query points and moving objects periodically. Mouratidis et al. (2005b) showed that the threshold-based approach to handling moving objects can be extended to support multiple moving query points. Using this method, the server keeps track of the location of each query point, while each data object keeps track of the locations and threshold of all queries. The server checks the thresholds as location updates arrive from the objects and refreshes the query results when one or more threshold violations occur.

Gedik and Liu (2004) presented a distributed solution to continuous monitoring of moving queries and moving objects that utilize the computational power of the mobile devices attached to mobile objects. The authors proposed a technique which enables trade-offs between query precision and query processing cost (in terms of network bandwidth and energy consumption). In order to handle a large number of queries, the authors also proposed a query-grouping mechanism to reduce the computational cost on the mobile device side. This mechanism allows continuous queries in proximity to be coprocessed.

Another stream of work drops the need for safe regions, as well as any assumptions about the movement patterns of objects and queries. Objects and queries move arbitrarily and unpredictably and report their new location to a processing server whenever they move. The aim of methods in this category is for the processing server to refresh the query results as quickly as possible in order to cater for the time-critical nature of monitoring applications. The dominant approach taken indexes the queries using a regular grid and augments that index with bookkeeping information to track the *influence region* of each query, i.e., the circular disk around the query location that includes its k nearest objects. Only objects leaving/entering the influence region of a query may affect its kNN result. The main representatives of this stream of work are YPK-CNN (Yu et al. 2005), SEA-CNN (Xiong et al. 2005), and CPM (Mouratidis et al. 2005a). A survey of techniques in this category can be found in Mouratidis (2009).

Key Applications

Aviation Safety

Keeping track of nearby locations at which an aircraft can land at all times is extremely crucial to aviation safety. An aircraft pilot can use the

continuous kNN query to precompute the nearest emergency landing sites along a predefined flight path.

Location-Based Advertising

Location-based advertising (LBA) is a form of advertising that uses location information to help identify potential customers. A business establishment can use continuous kNN query to monitor movements of nearby LBA participants over a period of time in order to offer promotional deals to participants who frequently appear in the kNN result set.

Location-Based Tour Guide System

A location-based tour guide system provides related tourist information based on a user's location. The system can make use of the continuous kNN query type to continuously report a list of nearby tourist attractions. A user can browse the list and select to retrieve information about the attraction in which they are most interested.

Multiplayer Online Gaming

In a multiplayer online gaming environment where different groups of players compete against each other, complete awareness of the surroundings is very important to each player. The continuous kNN query type can be used to report nearby threats to each player at all times.

Future Directions

- **Privacy Issues.** In location-based services, users may choose to obfuscate their locations before submitting them to a service provider for greater privacy (Duckham and Kulik 2005; Gruteser and Grunwald 2003). Continuous queries require users to repetitively share their locations with the service provider. This may provide the opportunity for the service provider to infer the trajectory of a user from a set of altered locations that they share by applying physical constraints such as speed, road network topology, etc. (Chow and Mokbel 2007).
- **Probabilistic Queries.** One research direction is to capture the uncertain nature of object/query trajectories (Cheng et al. 2004; Niedermayer et al. 2013). Probabilistic continuous querying is concerned with presenting a number of possible results along with probability assessments to a user over a period of time.
- **Continuous kNN with Spatial Constraints.** Another important research direction is concerned with incorporating spatial constraints into problem modeling and query processing. For example, in the presence of obstacles, one may be interested in monitoring k nearest objects that are visible from the query (Gao et al. 2011). In the context of objects and queries that move along the roads of a city, one would want to monitor the k nearest objects in terms of traveling distance within the road network (Mouratidis et al. 2006).

Cross-References

- ▸ Nearest Neighbor Query
- ▸ Queries in Spatiotemporal Databases, Time Parameterized

References

Aurenhammer F (1991) Voronoi diagrams – a survey of a fundamental geometric data structure. ACM Comput Surv 23(3):345–405

Cheng R, Kalashnikov DV, Prabhakar S (2004) Querying imprecise data in moving object environments. IEEE Trans Knowl Data Eng 16(9):1112–1127

Chow C, Mokbel MF (2007) Enabling private continuous queries for revealed user locations. In: SSTD, Boston, MA, USA, pp 258–275

Duckham M, Kulik L (2005) A formal model of obfuscation and negotiation for location privacy. In: Pervasive computing, third international conference, PERVASIVE 2005, Munich, 8–13 May 2005, Proceedings, pp 152–170

Gao Y, Zheng B, Chen G, Chen C, Li Q (2011) Continuous nearest-neighbor search in the presence of obstacles. ACM Trans Database Syst 36(2):9

Gedik B, Liu L (2004) Mobieyes: distributed processing of continuously moving queries on moving objects in a mobile system. In: EDBT, Heraklion, Crete, Greece, pp 67–87

Gruteser M, Grunwald D (2003) Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the first international conference on mobile systems, applications, and services, MobiSys 2003, San Francisco, 5–8 May 2003

Hu H, Xu J, Lee DL (2005) A generic framework for monitoring continuous spatial queries over moving objects. In: SIGMOD, Baltimore, Maryland, USA, pp 479–490
Iwerks GS, Samet H, Smith KP (2006) Maintenance of *k*-nn and spatial join queries on continuously moving points. ACM Trans Database Syst 31(2): 485–536
Li C, Gu Y, Qi J, Yu G, Zhang R, Yi W (2014) Processing moving knn queries using influential neighbor sets. PVLDB 8(2):113–124
Mouratidis K (2009) Continuous monitoring of spatial queries. In: Encyclopedia of database systems. Springer, New York, pp 479–484
Mouratidis K, Hadjieleftheriou M, Papadias D (2005a) Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring. In: SIGMOD, Baltimore, Maryland, USA, pp 634–645
Mouratidis K, Papadias D, Bakiras S, Tao Y (2005b) A threshold-based algorithm for continuous monitoring of k nearest neighbors. IEEE Trans Knowl Data Eng 17(11):1451–1464
Mouratidis K, Yiu ML, Papadias D, Mamoulis N (2006) Continuous nearest neighbor monitoring in road networks. In: VLDB, Seoul, Korea, pp 43–54
Niedermayer J, Züfle A, Emrich T, Renz M, Mamoulis N, Chen L, Kriegel H (2013) Probabilistic nearest neighbor queries on uncertain moving object trajectories. PVLDB 7(3):205–216
Nutanong S, Zhang R, Tanin E, Kulik L (2010) Analysis and evaluation of v*-*k*nn: an efficient algorithm for moving *k*nn queries. VLDB J 19(3):307–332
Okabe A, Boots B, Sugihara K (1992) Spatial tessellations: concepts and applications of Voronoi diagrams. Wiley, New York
Saltenis S, Jensen CS (2002) Indexing of moving objects for location-based services. In: ICDE, San Jose, California, USA, pp 463–472
Song Z, Roussopoulos N (2001) K-nearest neighbor search for moving query point. In: SSTD, Redondo Beach, CA, USA, pp 79–96
Tao Y, Papadias D (2002) Time-parameterized queries in spatio-temporal databases. In SIGMOD, Madison, Wisconsin, USA, pp 334–345
Tao Y, Papadias D, Shen Q (2002) Continuous nearest neighbor search. In: VLDB, Hong Kong, China, pp 287–298
Tao Y, Papadias D, Sun J (2003) The tpr*-tree: an optimized spatio-temporal access method for predictive queries. In: VLDB, Berlin, Germany, pp 790–801
Xiong X, Mokbel MF, Aref WG (2005) SEA-CNN: scalable processing of continuous k-nearest neighbor queries in spatio-temporal databases. In: ICDE, Tokyo, Japan, pp 643–654
Yu X, Pu KQ, Koudas N (2005) Monitoring k-nearest neighbor queries over moving objects. In: ICDE, Tokyo, Japan, pp 631–642
Zhang J, Zhu M, Papadias D, Tao Y, Lee DL (2003) Location-based spatial queries. In: SIGMOD, San Diego, California, USA, pp 443–454

Dynamic Travel Time Maps

Sotiris Brakatsoulas[1], Dieter Pfoser[1], Nectaria Tryfona[2,3], and Agnès Voisard[4]
[1]RA Computer Technology Institute, Athens, Greece
[2]Talent Information Systems SA, Athens, Greece
[3]Talent SA, Athens, Greece
[4]Fraunhofer ISST and FU Berlin, Berlin, Germany

Synonyms

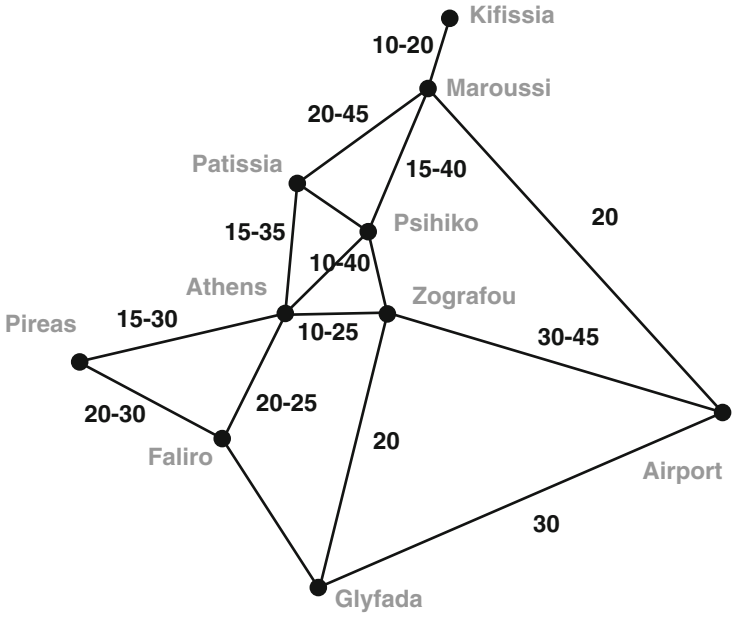
Characteristic Travel Time; Spatial Causality; Travel Time Computation

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



Dynamic Travel Time Maps, Fig. 1 Fluctuating travel times (in minutes) in a road network example (Athens, Greece)

vehicles (Pfoser et al. 2006). 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 $w : E \rightarrow \mathbf{R}$ 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.

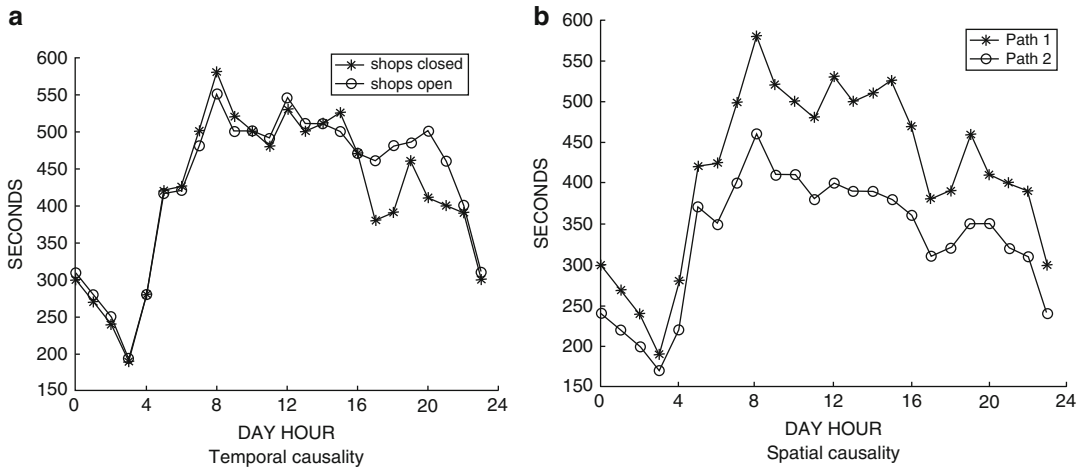
Although the various algorithmic solutions for routing and navigation problems (Dechter and Pearl 1985), Russell and Norvig (2003) are still subject to further research, the basic place for improving solutions to routing problems is this underlying weight-based database $DB(w(u, v))$.

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 (Brakatsoulas et al. 2005). 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) (Chen and Chien 2002; Schaefer et al. 2002).

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 profiles, i.e., the



Dynamic Travel Time Maps, Fig. 2 Relating travel times

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 16–22 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:

$$\rho(e) = \frac{\tau(e)}{l(e)}\,, \tag{1}$$

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,

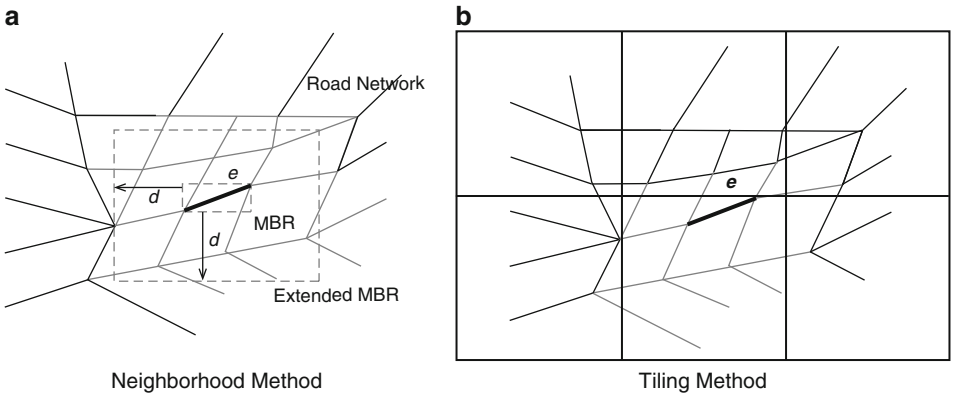
$$\chi(P) = \{|P|, E[P], V[P]\}\,, \tag{2}$$

$$E[P] = \sum_{\rho \in P} \frac{\rho}{|P|}\,, \tag{3}$$

$$V[P] = \sum_{\rho \in P} \frac{(\rho - E[P])^2}{|P|}\,. \tag{4}$$

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.

$$P(e) = \{\rho(e^*, t) : e^* \in I_S(e) \wedge t \in I_T(e)\}\,. \tag{5}$$



Dynamic Travel Time Maps, Fig. 3 Characteristic travel time computation methods

$I_S(e)$, 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$(e) \bigwedge L(e^*) = L(e)\}$, 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. $I_S(e) = \{e^* : e^* \in Tile(e) \bigwedge L(e^*) = L(e)\}$

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 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 m 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*, 1–24, 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, 1–52, *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.

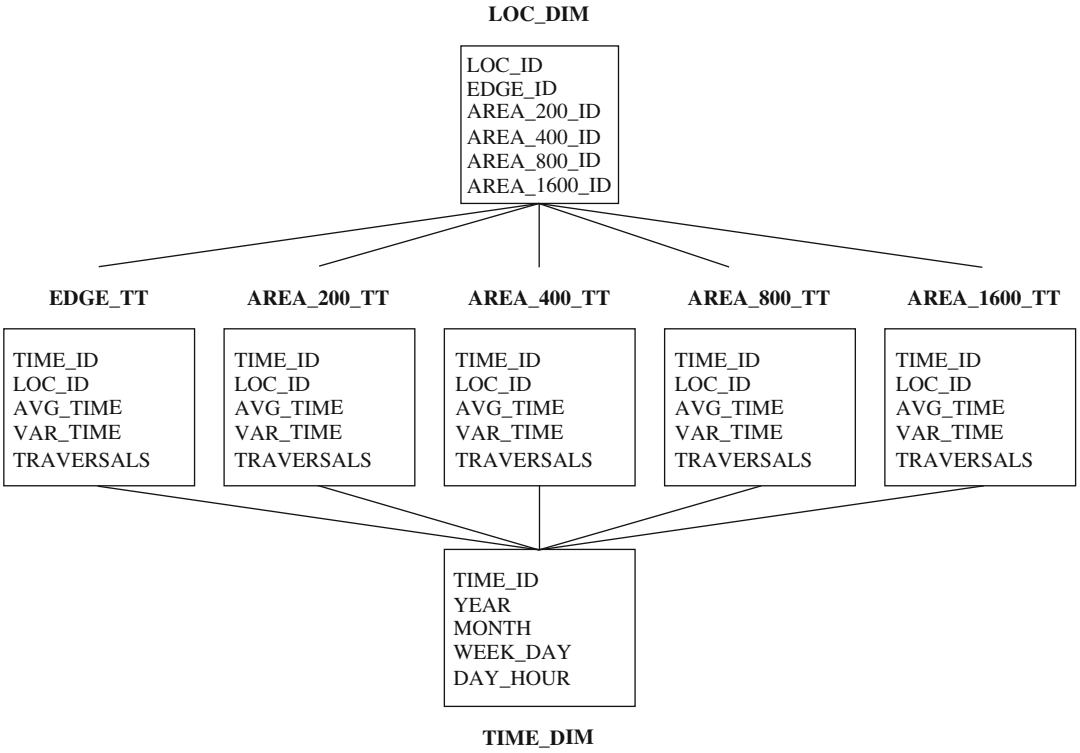
$$|C_i| = \sum_{S_j \in C_i} |S_j| \tag{6}$$

$$E[C_i] = \frac{\sum_{S_j \in C_i} |S_j| \cdot E[S_j]}{|C_i|} \tag{7}$$

$$V[C_i] = \frac{\sum_{S_j \in C_i} |S_j| \left(V(S_j) + E[S_j]\right)}{|C_i|} - E^2[C_i] \tag{8}$$

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 Kimball (2002) and Scalzo (2003). 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



Dynamic Travel Time Maps, Fig. 4 Data warehouse schema

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 26,000 trajectories that in turn consist of 11 million segments. The data was collected using GPS vehicle tracking during the years 2000–2003 in the road network of Athens, Greece. Details on the experimental evaluation can be found in Pfoser et al. (2006).

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 (Fleischmann et al. 2004).

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

References

Brakatsoulas S, Pfoser D, Sallas R, Wenk C (2005) On map-matching vehicle tracking data. In: Proceedings of the 31st VLDB conference, Norway, pp 853–864

Chen M, Chien S (2002) Dynamic freeway travel time prediction using probe vehicle data: link-based vs. path-based. J Trans Res Board 1768:157–161

Dechter R, Pearl J (1985) Generalized best-first search strategies and the optimality of A*. J ACM 32(3):505–536

Fleischmann B, Sandvoß E, Gnutzmann S (2004) Dynamic vehicle routing based on on-line traffic information. Trans Sci 38(4):420–433

Kimball R (2002) The data warehouse toolkit, 2nd edn. Wiley, New York

Pfoser D, Tryfona N, Voisard A (2006) Dynamic travel time maps – enabling efficient navigation. In: Proceedings of the 18th SSDBM conference, Vienna, pp 369–378

Russell S, Norvig P (2003) Artificial intelligence: a modern approach, 2nd edn. Prentice Hall, New York

Scalzo B (2003) Oracle DBA guide to data warehousing and star schemas, 1st edn. Prentice Hall, New York

Schaefer R-P, Thiessenhusen K-U, Wagner P (2002) A traffic information system by means of real-time floating-car data. In: Proceedings of the ITS world congress, Chicago

Dynamics

- ▸ Geographic Dynamics, Visualization and Modeling