M

M&S Computing

▶ Intergraph: Real-Time Operational Geospatial Applications

Machine Learning

▶ Clustering of Geospatial Big Data in a Distributed Environment

Machine Readable Geographic Data

▶ Feature Catalogue

Magik, Smallworld

▶ Smallworld Software Suite

Management of Linear Programming Queries

▶ MLPQ Spatial Constraint Database System

Manhattan Distance

▶ Distance Metrics

Manifold

▶ Spatial Data Transfer Standard (SDTS)

Manifold Rules

▶ Smallworld Software Suite

Manifolds

▶ Smallworld Software Suite

Mantel Test

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

Map Accuracy

▶ Spatial Data Transfer Standard (SDTS)

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

Map Construction from GPS Data

Sophia Karagiorgou
Institute for the Management of Information
Systems, R.C. ATHENA, Marousi, Greece

Synonyms

Road Map Extraction; Road Network Generation; Transportation Network Inference

Definition

Map construction algorithms automatically produce and/or update street map datasets using tracking data. The ubiquitous generation of geo-referenced tracking data provides us with a wealth of trajectories coming from mobile objects. For instance, smartphone applications involving check-ins, real-time navigation applications, fleet management services, etc., generate huge amounts of tracking data which can be used to construct road and transportation networks. Although there are many mapping efforts, road networks are intrinsically dynamic, especially in situations of sudden or catastrophic events. This makes map construction algorithms more valuable. To that end, the primary challenge is to provide novel techniques for constructing transportation networks of great spatial accuracy with quality guarantees.

However, the collected tracking data have errors and inherent inaccuracies such as GPS errors, transmission errors, low sampling rates, etc. Therefore, the map construction problem becomes very challenging. Consider the example in Fig. 1, which plots the trajectories from vehicle tracking data in Berlin (Fig. 1a), the actual road network (Fig. 1b) from OpenStreetMap (OSM), and the rendered OSM map (Fig. 1c). The goal is to construct the road network of Fig. 1b from the GPS tracking data of Fig. 1a. Clearly, constructing a road network from such tracking data is not a trivial task.

Historical Background

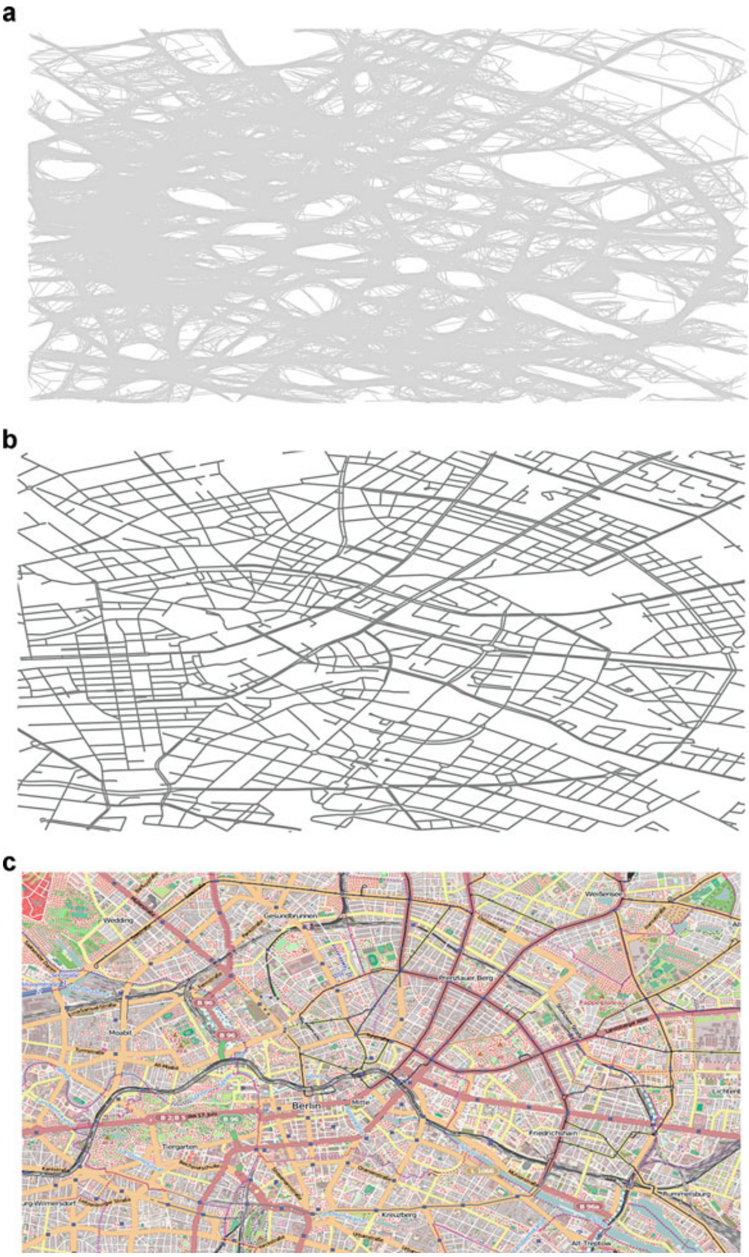
The commoditization of tracking technology provides us with huge amounts of tracking data which allow us to derive road and transportation networks. Existing methods are characterized by limited geographical scope, small-scale tracking datasets, and unconvincing map construction results. Map construction algorithms are coming from different scientific domains. In the following, related work in the area of map construction algorithms is presented by also outlining the limitations of existing approaches. The methods which have been proposed either cluster trajectories, or points, or infer transportation networks by applying track insertion or linking of intersections.

Trajectory Clustering

Various approaches of map construction are based on *trajectory clustering*. Algorithms in this category apply clustering techniques to the GPS tracking data by taking into account the position measurement of the moving objects. The majority of the proposed methods use well-known algorithms including k-means (Edelkamp and Schrödl 2003) and DBSCAN (Kisilevich et al. 2010), but they work strictly with point data and do not take the temporal aspect of trajectories into consideration. By disregarding the temporal dimension, one cannot derive attributes such as speed profiles, mean sampling rate, etc. By considering the temporal dimension of the tracking data in the clustering process it enables to infer additional characteristics such as road categories (highways, main and secondary roads, etc.), transportation hierarchies, etc.

Point Clustering

Algorithms in this category assume that the input consists of a set of points. These points are then clustered in various different ways to obtain street segments which are finally connected to a street map. The input point set either comprises the set of all raw input measurements or a dense sample of all input tracks obtained from interpolating, usually piecewise linearly, between GPS measurements. These algorithms include



Map Construction from GPS Data, Fig. 1 Vehicle tracking data, actual road network, actual map. (**a**) GPS tracking data. (**b**) Ground-truth road network (OSM). (**c**) Ground-truth rendered map (OSM)

methods based on k-means clustering, kernel density estimation, and computational geometry.

Some approaches employ the k-means algorithm to cluster the input point set, using distance measures (e.g., Euclidean distance) and possibly also vehicle direction, as a condition to introduce seeds at fixed distances along a path. These algorithms perform well for road segmentation, map-matching, and lane clustering.

Guo et al. (2007) make use of statistical analysis of GPS tracks, assuming that the GPS data follow a symmetric 2D Gaussian distribution. This assumption may become unrealistic, especially in error-prone environments or in areas with signal loss.

Other approaches related to map construction algorithms employ kernel density estimation (KDE) methods to first transform the input point set to a density-based discretized image. Most of the KDE-based algorithms function well either when the data are frequently sampled (i.e., once per second) or when there is a lot of data redundancy (Biagioni and Eriksson 2012; Davies et al. 2006). Generally, KDE algorithms have a hard time overcoming the problem of noisy samples especially when they accumulate in an area.

In the computational geometry community, the proposed map construction algorithms cluster the input points using local neighborhood properties by employing either Voronoi diagrams or Delaunay triangulations (Chen et al. 2010; Ge et al. 2011). All these algorithms assume a densely sampled input point set and provide theoretical quality guarantees for the constructed output map. Also, they make certain assumptions on the underlying street map and the input tracks.

Chen et al. (2010) focus on detecting "good" street portions in the road network and connect them subsequently. The theoretical quality guarantees that which are given, however, assume dense point sample coverage and error bounds and make constrained assumptions on the road geometry.

Incremental Track Insertion

Algorithms in this category construct a street map by incrementally inserting tracks into an initially empty map, often making use of map-matching ideas. Distance measures and vehicle headings are also used to perform additions and deletions during the incremental construction of the map.

Bruntrup et al. (2005) propose a spatial-clustering-based algorithm that requires high-quality tracking data (i.e. high sampling rate, and positional accuracy), while Ahmed and Wenk (2012) present an incremental method that employs the Fréchet distance to partially match the tracks to the map. Although partial quality guarantees are given, their approach does not address the basic connectivity problem, i.e., how to construct connected road networks and how to measure their respective quality.

Intersection Linking

While related to point clustering, intersection linking follows an alternate approach. The intersection linking method first detects the intersections of the street map. In a second step, the algorithm identifies suitable street segments and links the intersections of the street together.

Fathi and Krumm (2010) provide an approach which detects intersections by using a prototypical detector trained on ground-truth data from an existing map. While a map is finally derived, their approach works best for vertically aligned maps and high sampling rates.

Scientific Fundamentals

Toward the overall map construction goal, which is the provision of evolving and updated maps from vast amounts of GPS tracking data, two major issues have been identified: (i) the construction of road networks from redundant GPS tracking data and (ii) the construction and maintenance of road networks from sparse GPS tracking data.

In the following, the *TraceBundle* algorithm (Karagiorgou and Pfoser 2012) is presented which constructs road maps from redundant GPS tracking data. It is also presented the *TraceConflation* algorithm (Karagiorgou et al. 2013) which constructs and updates road maps from sparse GPS tracking data. They rely on detecting changes in the direction of mobile objects movement to infer intersections and then "bundling" the trajectories around them to create the map edges for road networks of arbitrary geometries. Both algorithms follow the intersection linking approach and perform well for road networks of different scale and density.

Map Construction from GPS Tracking Data

The *TraceBundle* algorithm is an intersection linking approach and emphasizes the correct detection of intersection nodes and the linkage of these nodes, both in terms of connectivity and actual geometry. Intersections are identified based on movement characteristics (i.e. speed, direction) and point density. The intersections are then linked by interpolating the geometry of the connecting traces. TraceBundle is efficient toward redundant data and road networks of arbitrary geometries. It also exploits the ubiquitous vehicle tracking data in order to analyze, reconstruct, and extract road network geometries enriched by attributes.

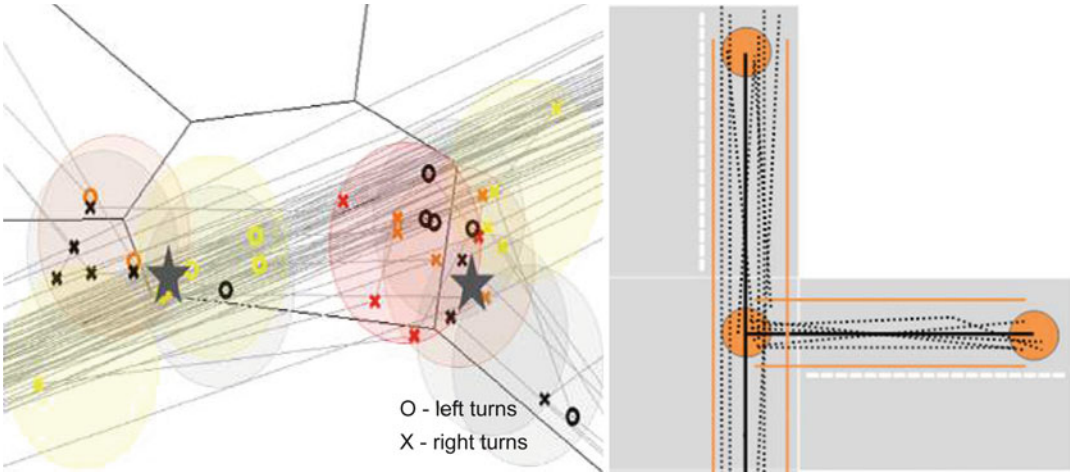
This intersection linking map construction algorithm is a heuristic approach that "bundles" trajectories around intersection nodes. It represents the street map as a directed graph in which each edge is labeled as directed or bi-directed.

The main contribution of the *TraceBundle* algorithm is its methodology to derive intersection nodes. It relies on detecting changes in movement to cluster "similar" nodes, in terms of turn classification and enabled maneuvers related with an intersection. A node at which a change in direction and speed occurs is considered a turn indicator. Change in direction is identified while processing trajectories by using math degrees and change in speed by using the reduction of mean speed over consecutive position samples. Turn clusters are produced based on (i) spatial proximity and (ii) turn type. The centroid of turn clusters then becomes an intersection node. Connecting the trajectories to intersection nodes and compacting them, one allows to derive links and consequently the entire geometry of the road network.

The *TraceBundle* algorithm has three tunable parameters, *angular difference*, *spatial proximity*, and *speed*. Angular difference is the relative change of the vehicle direction. The speed threshold indicates vehicles slowing down while they are turning. The speed threshold is an empirical maximum threshold to separate high-speed turns from turns at intersections. Spatial proximity distance threshold is used for clustering turn clusters into intersection nodes.

The essential steps of the *TraceBundle* algorithm are as follows: (1) *Turn samples* – given a trajectory, each position sample at which a significant change in direction and speed (parameters) occurs becomes a turn sample. (2) *Turn clusters* – clustering turn samples based on (i) proximity (parameter) and (ii) a static turn model to create turn clusters. (3) *Intersection nodes* – compute centroid of turn clusters. (4) *Connecting intersection nodes* – using constituting turn samples, connect trajectories to respective intersection nodes. (5) *Compacting links* – merge connecting trajectory portions between intersection nodes to generate links.

Figure 2 visualizes the basic steps of the algorithm. Figure 2a shows the constructed intersection nodes as gray stars from GPS tracking data. The constituting turn samples are shown as x and o markers. Turn clusters are shown as highlighting circles. Different colors represent different vehicle movements and thus different turn types related with an intersection. By using x and o markers, clockwise and counterclockwise turns



Map Construction from GPS Data, Fig. 2 The *TraceBundle* algorithm. (**a**) Intersection nodes. (**b**) Compacting links are represented, respectively. Figure 2b shows the created links between intersection nodes as black lines. The constituting trajectories of road portions between two intersections are shown as black dashed lines. These portions of trajectories are merged to create links.

The *TraceBundle* algorithm is efficient toward redundant data and road networks of arbitrary geometries. It also exploits the ubiquitous vehicle tracking data in order to analyze, reconstruct, and extract road network geometries enriched by attributes.

Map Construction from Sparse Tracking Data

The *TraceConflation* algorithm is also an intersection linking approach and converts movement trajectories into a hierarchical transportation network from sparse tracking data. It emphasizes the segmentation of the underlying movement network based on speed characteristics. It also incorporates changes and updates in a layered and incremental fashion using dynamically determined parameters. This method is more is robust and provides more accurate results when dealing with noisy and heterogeneous datasets with low and nonuniform sampling rates.

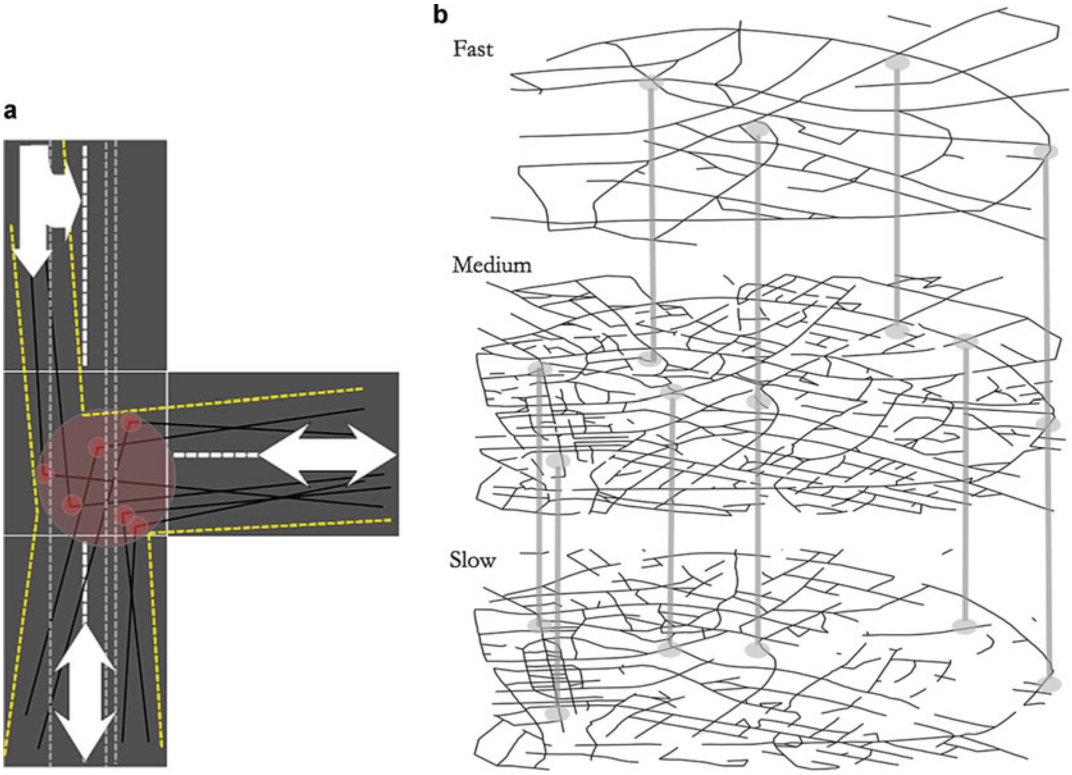
This intersection linking map construction algorithm is a heuristic approach that segments the input dataset into groups of trajectories in a layered form based on speed hierarchies. It then “conflates” the different layers, starting from higher-speed hierarchies by gradually incorporating lower hierarchies into the final map. It also provides a mechanism to accommodate automatic map maintenance on updates. It represents the street map as a directed graph in which each edge is labeled as directed or bi-directed edge.

The main contribution of the *TraceConflation* algorithm is its methodology to segment and “conflate” low sampled GPS tracking data based on movement characteristics by using derived speed categories while processing tracking data. This methodology also introduces a proximity-based expansion algorithm around turn samples based on turn similarity, which allows to create intersection nodes based on the available data by using sets of trajectories that belong to the same speed category. Finally, it applies a method to hierarchically construct road network layers, based on different types of movement in an urban context, which are then conflated and combined into a single network.

The essential steps of the *TraceConflation* algorithm are as follows: (1) *Node detection* – identifies position samples by using sets of trajectories that belong to the same speed category and clusters them into intersection nodes based on turn similarity criteria. (2) *Segmentation of trajectories* – analyzes the trajectories in the input data and splits them into subsets based on different speed characteristics. (3) *Construction of network layers* – creation of different network hierarchies based on different speed categories. (4) *Conflation of network layers* – fusion of the created network layers for the different speed categories to produce the overall road network.

Figure 3 visualizes the basic steps of the algorithm. Figure 3a shows the constructed intersection node based on trajectory turn samples of the same speed category. The intersection node is the centroid of the turn samples. Figure 3b shows an example of the conflation process. It depicts the three network layers that are constructed after segmenting the entire trajectory dataset of Fig. 1a. Gray lines link the various connection points between the constructed network layers.

The *TraceConflation* algorithm focuses on sparse tracking data and delivers methods which convert movement trajectories into a hierarchical transportation network. The algorithm has been motivated by the lack of map construction algorithms on tasks related with map maintenance and updates; the emphasis is given on building a road map hierarchically and gradually by only incorporating new map portions. For this purpose, the *TraceConflation* algorithm exploits the types of vehicle movement in an urban context to segment the input data accordingly, construct the network in a layered fashion, and deliver road maps of high spatial accuracy for sparse datasets.



Map Construction from GPS Data, Fig. 3 The *TraceConflation* algorithm. (**a**) Intersection nodes. (**b**) Layer conflation

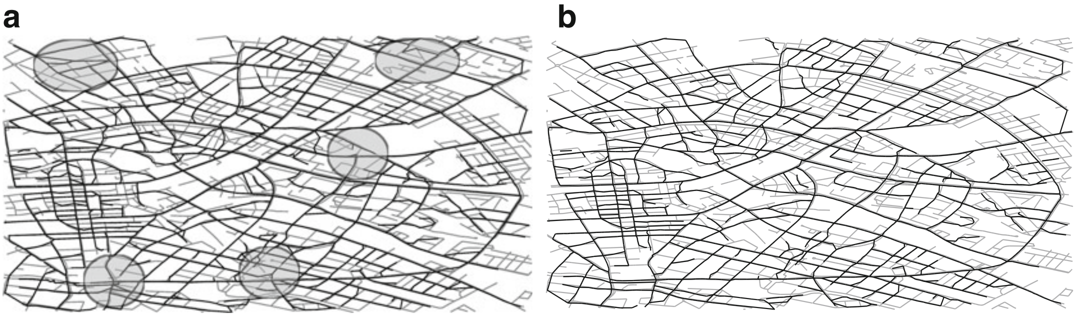
Key Applications

During the last years, the widespread adoption and use of GPS-enabled devices, in conjunction with the increasingly popular phenomenon of crowdsourcing, have opened up new opportunities for tracking the movement of various types of entities, including vehicles, humans, and animals. Consequently, this has enabled a wide spectrum of novel applications and services.

The map construction problem has two broad categories of application scenarios. The first refers to cases where the entities move along specific trails, which, however, are not already mapped. Such examples include the movement of hikers, or animals, the gamification of trails (e.g., activity trackers), as well as cases where a map exists but is not publicly available because it is too costly to acquire. The goal in these cases is to track the movement of the entities and use the extracted trajectories to infer a map of the movement or the transportation network.

The second scenario refers to cases where a map exists but needs to be maintained and updated, or enhanced with additional properties. For example, maps of road networks are traditionally created through the use of aerial imagery, a method which is not suitable for keeping up with road changes or determining dynamic aspects, such as traffic controls, turn restrictions, blockages due to accidents or natural phenomena, etc. Map construction algorithms facilitate the provision of evolving and updated transportation maps from vast amounts of trajectory data.

Figure 4 visualizes the constructed road networks for Berlin using the *TraceBundle* and the *TraceConflation* algorithms, respectively, by using the tracking data of Fig. 1a. In each case, the inferred network is visualized using black lines, while the ground-truth network is shown



Map Construction from GPS Data, Fig. 4 Constructed road networks. (**a**) TRACEBUNDLE. (**b**) TRACECONFLATION

using light gray lines. The ground-truth network is derived from the OpenStreetMap dataset.

For better illustration, some areas, in which improvements of *TraceConflation* over *TraceBundle* can be observed, are highlighted. The overall observation is that the *TraceConflation* method produces results in which the core network is depicted accurately, especially in the cases of sparse and less frequently sampled GPS tracking data, i.e., for sampling rate >40 s. The *TraceBundle* algorithm performs well, but it becomes more efficient when there is data redundancy.

Future Directions

Nowadays, vast amounts of geo-referenced trajectory data are being collected due to the ubiquitous availability of positioning technologies. The presented algorithms have focused on the provision of evolving and updated transportation maps from vast amounts of trajectory data. In these emerging environments, classical maps are no longer sufficient to keep up with sudden changes. Instead, the users typically would not need a high-quality map, but only a generalized map which shows specific aspects such as road closures, landmarks, and important routes. In this direction, map construction algorithms address the very timely challenge of analyzing such data. They also propose novel methods for the construction and maintenance of digital street maps, which are among the most valuable digital data resources in today's society.

The presented map construction approaches create navigable road networks of arbitrary geometries and from tracking data of low sampling rate. Previous research efforts have focused on map construction from frequently sampled tracking data and vertically aligned road networks. The presented algorithms elaborate on these approaches and extend them providing more advanced capabilities. For further research purposes, the *mapconstruction.org* site has been established by making available map construction and evaluation algorithms, datasets, and constructed maps, to motivate other researchers toward the contribution in the area of map construction. Still, several research issues remain open. In the conclusion, most of the prominent ones are identified and outlined.

The problems considered in this approach are very relevant to trends and requirements that can be identified in other emerging paradigms as well, most notably geomarketing. Geomarketing solutions answer the "where" questions which impact performance, drive growth, and optimize business so it can exploit potential markets. Map construction algorithms can facilitate the discovery and analysis of these geo-referenced data, and it will be very challenging to provide solutions to such activities in the near future.

Another important issue when inferring tracking data from different sources and with different characteristics is related to the transportation networks. Transportation networks are commonly represented using networks as an analogy for their structure and flows. The existing approaches target road networks. They can be extended to cover various types of links between points along which movements can take place. Thus, it would be very interesting to study how the techniques proposed can be applied to integrate noticeable interdependencies among the different nodes and networks over time, based on spatial and functional proximity. It would be also interesting to study means of multimodal transportation from heterogeneous tracking data sources and to develop more generic approaches for other types of transportation networks.

When tracking data are gathered from several heterogeneous sources, potentially of different levels of quality, it may be uncertain, incomplete, or inconsistent. This requires advanced techniques in order to manage and reason with such data. Hence, it would be a challenging issue to study how the techniques proposed can be adapted or extended to take into account this additional aspect.

Cross-References

- ▶ Data Analysis, Spatial
- ▶ Data Mining Techniques for the Characterization of Dynamic Regions in Spatiotemporal Data
- ▶ Geographic Knowledge Discovery
- ▶ Trajectory Mining

References

Ahmed M, Wenk C (2012) Constructing street networks from GPS trajectories. In: Proceedings of the 20th annual European symposium on algorithms, Ljubljana, pp 60–71

Biagioni J, Eriksson J (2012) Map inference in the face of noise and disparity. In: Proceedings of the 20th ACM SIGSPATIAL GIS conference, Redondo Beach, pp 79–88

Bruntrup R, Edelkamp S, Jabbar S, Scholz B (2005) Incremental map generation with GPS traces. In: Proceedings of the IEEE intelligent transportation systems, Vienna, pp 574–579

Chen D, Guibas LJ, Hershberger J, Sun J (2010) Road network reconstruction for organizing paths. In: Proceedings of the 21st annual ACM-SIAM symposium on discrete algorithms, Austin, pp 1309–1320

Davies JJ, Beresford AR, Hopper A (2006) Scalable, distributed, real-time map generation. IEEE Pervasive Comput 5(4):47–54

Edelkamp S, Schrödl S (2003) Route planning and map inference with global positioning traces. Comput Sci Perspect 2598:128–151

Fathi A, Krumm J (2010) Detecting road intersections from GPS traces. Geogr Inf Sci 6292:56–69

Ge X, Safa I, Belkin M, Wang Y (2011) Data skeletonization via Reeb graphs. In: Proceedings of the 25th annual conference on neural information processing systems, Granada, pp 837–845

Guo T, Iwamura K, Koga M (2007) Towards high accuracy road maps generation from massive GPS traces data. In: Proceedings of the IEEE international geoscience and remote sensing symposium, Barcelona, pp 667–670

Karagiorgou S, Pfoser D (2012) On vehicle tracking data-based road network generation. In: Proceedings of the 20th ACM SIGSPATIAL GIS conference, Redondo Beach, pp 89–98

Karagiorgou S, Pfoser D, Skoutas D (2013) Segmentation-based road network construction. In: Proceedings of the 21th ACM SIGSPATIAL GIS conference, Orlando, pp 470–473

Kisilevich S, Mansmann F, Keim D (2010) P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geotagged photos. In: Proceedings of the 1st international conference and exhibition on computing for geospatial research and application, Bethesda, MD, USA, pp 38:1–38:4

Map Data

▶ Photogrammetric Products

Map Distribution

▶ Data Infrastructure, Spatial

Map Generalization

Anne Ruas
Laboratory COGIT, IGN-France, Saint-Mandé, France

Synonyms

Generalization

Definition

Map generalization is the name of the process that simplifies the representation of geographical data to produce a map at a certain scale with a defined and readable legend. To be readable at a smaller scale, some objects are removed; others are enlarged, aggregated and displaced one to another, and all objects are simplified. During the process, the information is globally simplified but stays readable and understandable.

The smaller the scale, the less information is given per square kilometer. Conversely, the larger the scale, the more detailed is the area mapped for the same map size. For a given size of map sheet, nearly the same quantity of information is given for different scales, either privileging the density of field information (for larger scale) or the spatial extension (for smaller scale).

This process is used both in manual and digital cartography.

Main Text

Generalisation can be first defined by means of graphical constraints and scale. On a map the information is represented by means of symbols. These symbols are coded representations which ensure the interpretation of the meaning. These symbols have minimum sizes that ensure not only good perception but also the recognition of the symbols and their associated meaning. As an example, a very small black polygon representing a building will be seen as a dot and not a building. In the same way two symbols too close one to another will be seen as a single symbol even if in the real world they represent two different entities. These graphical constraints are called legibility or readability constraints. When represented on a map the graphical objects are not a faithful representation of the entities' sizes at a given scale but are symbolic representations which maximize communication of information.

In order to respect graphical constraints, some objects are enlarged and some are displaced one to another. These geometric distortions are minor at the large scale (1:2,000–1:10,000), common at the medium scale (1:15,000–1:50,000) and frequent and very large at small scales. As an example a 6 m width road represented by a line of 0.6 mm on a map is enlarged 10 times at 1:100,000 and 100 times at 1:1,000,000! Of course, when the scale decreases it is physically not possible to enlarge and displace all objects: many objects are removed, some are aggregated, and they are all simplified. These operations of enlargement, displacement, selection, aggregation and geometric simplifications are the main operations of generalization. There is a good large amount of literature devoted to defining and specifying the operators and algorithms of generalization.

But generalization can also be defined through a more geographical view point. A geographical representation-being a map or a data base-represents the real world at a certain level of detail. This level of detail implicitly defines the type of information represented and their reasonable use. Geographical phenomena have a certain size and they can only be depicted within a certain scale range. Generalization is a process that aggregates and simplifies the information in order to make apparent more general concepts through their representation. When scale decreases some concepts disappear while others appear. The transition between scales can by smooth and continuous for some themes or abrupt for others. Generally networks are preserved but their density decreases, while small size objects such as houses are changed into urban areas, dots and even nothing. Müller et al. (1995) speaks of change under scale progression (CUPS) to characterize these transformations through scale.

The complexity of the automation of the generalization process is due to the diversity of geographical information and its contextual nature. Generalization should preserve relationships and properties that are, most of the time, implicit. Current models of generalization rely on optimization techniques or on a multiagent system paradigm.

Cross-References

- ▶ Abstraction of GeoDatabases
- ▶ Generalization and Symbolization

References

Müller J-C, Lagrange J-P, Weibel R (1995) GIS and generalization-methodology and practice. Taylor and Francis, London

Recommended Reading

Mackaness W, Ruas A, Sarjakoski LT (2007) Generalisation of geographic information: cartographic modelling and applications. Elsevier, Burlington

McMaster R, Buttenfield B (1991) Map generalization. Longman, Harlow

Map Overhaul

▶ Positional Accuracy Improvement (PAI)

Map Overlay

▶ Polygonal Overlay Computation on Cloud, Hadoop, and MPI

Map Quality

▶ Positional Accuracy Improvement (PAI)

Map, Bi-plot

▶ Geographic Dynamics, Visualization and Modeling

Map, Centrographic Timeseries

▶ Geographic Dynamics, Visualization and Modeling

Map-Matching

Dieter Pfoser
RA Computer Technology Institute, Athens, Greece

Definition

Sampling vehicular movement using GPS is affected by error sources. Given the resulting inaccuracy, the vehicle tracking data can only be related to the underlying road network by using *map-matching* algorithms.

Main Text

Tracking data is obtained by sampling movement, typically using GPS. Unfortunately, this data is not precise due to the *measurement error* caused by the limited GPS accuracy, and the *sampling error* caused by the sampling rate, i.e., not knowing where the moving object was in between position samples. A processing step is needed that matches tracking data to the road network. This technique is commonly referred to as *map matching*.

Most map-matching algorithms are tailored towards mapping *current positions* onto a vector representation of a road network. Onboard systems for vehicle navigation utilize dead reckoning besides continuous positioning to minimize the positioning error and to produce accurate vehicle positions that can be easily matched to a road map. For the purpose of processing tracking data, the *entire trajectory*, given as a sequence of historic position samples, needs to be mapped. The fundamental difference in these two approaches is the error associated with the data. Whereas the data in the former case is mostly affected by the measurement error, the latter case is mostly concerned with the sampling error.

Cross-References

- ▶ Dynamic Travel Time Maps
- ▶ Floating Car Data

Recommended Reading

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

Mapping

▶ Public Health and Spatial Modeling

Mapping and Analysis for Public Safety

▶ Hotspot Detection, Prioritization, and Security

MapReduce

Qunying Huang
Department of Geography, University of
Wisconsin – Madison, Madison, WI, USA

Synonyms

Hadoop; Hadoop MapReduce

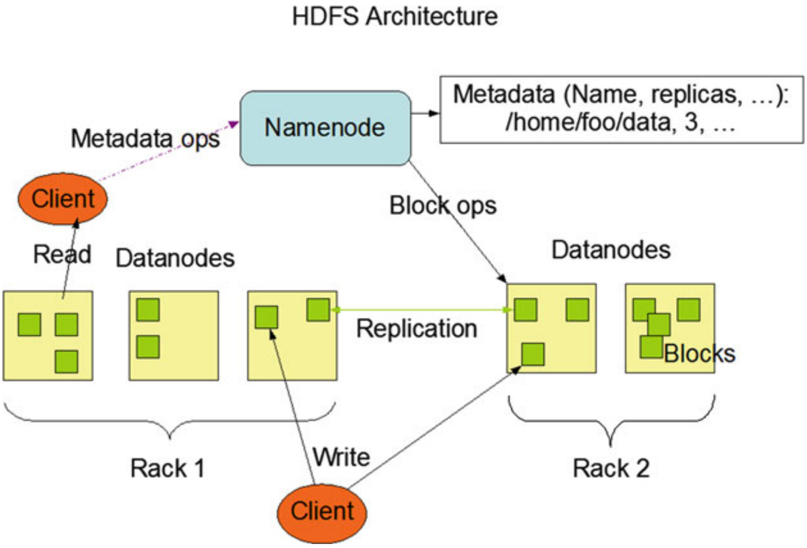
Definition

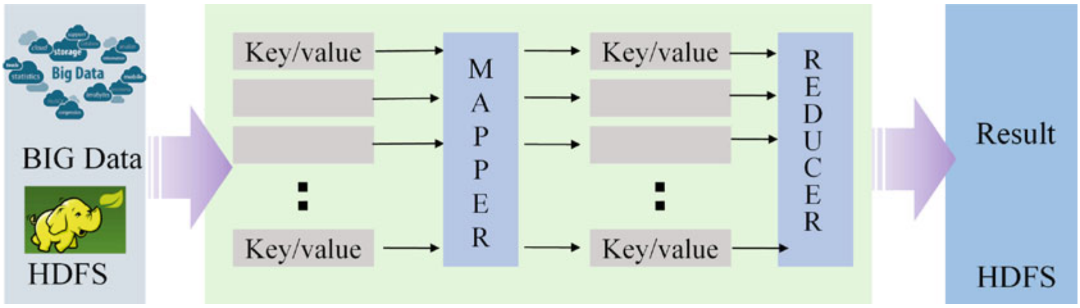
MapReduce refers to a parallel programming model and an associated implementation for processing and generating large datasets (Dean and Ghemawat 2008). It is built on the simple concept of mapping which filters and sorts data (e.g., sorting words alphabetically into queues, one queue for each word) and reducing that reduces the data through summary operations (e.g., counting the number of words in each queue and producing word frequencies). MapReduce has become widely used in today's big data processing work due to the following reasons (Dean and Ghemawat 2008):

- It hides the details of parallelization, fault tolerance, data locality optimization, and load balancing and therefore relieves the burden of the programmers dealing with distributed programming;
- A wide range of computing problems could be addressed by the MapReduce model, e.g., generation of data for Google's production web search service, sorting, data mining, machine learning, etc.;
- MapReduce's framework can leverage and scale thousands of machines, making it well suitable for real-world workloads.

Today, the Apache Hadoop project is the most widely used implementation of MapReduce framework. Hadoop distributed file system (HDFS), another important component of the Hadoop project, is typically used along with Hadoop MapReduce. Many IT companies have adopted HDFS as their big data storage technology, such as Yahoo, Intel, and IBM. The HDFS holds the actual data in an array of storage cluster nodes. It consists of a single NameNode and a number of DataNodes (Fig. 1). The NameNode is configured on the master node that manages the file metadata information and regulates access to files by clients. Typically, each computing node in the cluster would have one DataNode to manage storage attached to the computing node. Therefore, an HDFS cluster uses master/slave architecture in managing and organizing the data with each computing node serving as a DataNode. Within such an architecture, each file is split into one or more blocks, and these blocks are stored in a set of DataNodes (Hadoop 2014). The NameNode handles file system namespace operations such as opening, closing, and renaming files and directories. It also provides the support of the mapping of blocks to DataNodes. The DataNodes respond to read and write requests from the file system's clients that they run on. The DataNodes also support block creation, deletion, and replication upon instruction from the NameNode (Hadoop 2014).

MapReduce, Fig. 1 The architecture of an HDFS cluster (Source: Hadoop 2014)



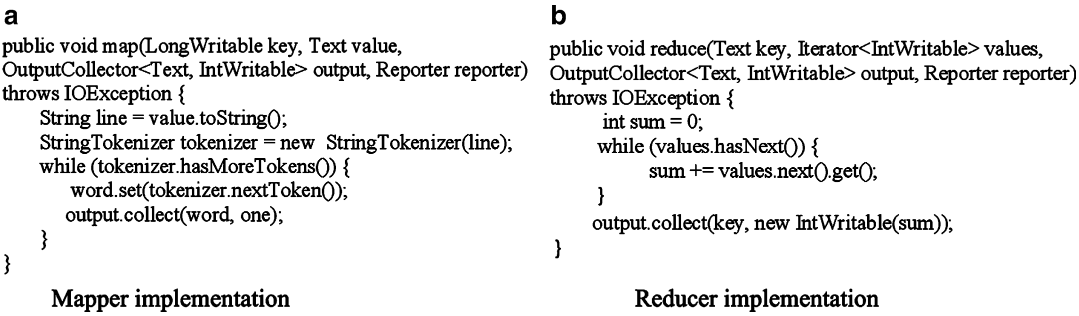


MapReduce, Fig. 2 How MapReduce works

A traditional high-performance computing (HPC) architecture is usually implemented based on the Message Passing Interface (MPI) programming model, and each node is designed to access the same remote data storage to execute tasks in parallel. In general, the data storage is attached to a master node and then shared among the computing nodes using a parallel file system, such as Network File System (NFS) or Parallel Virtual File System (Huang et al. 2013). In this regard, all the data sits on one machine, and all of the data processing software is installed on a set of servers. Therefore, the communication overhead is introduced while passing and synchronizing the data across the nodes.

However, by using DFS as the underlying file system, data and MapReduce processing are hosted on the same nodes in the cluster. Such configuration allows the framework to effectively schedule tasks on the nodes where data are already present, resulting in very high aggregate bandwidth across the cluster (Hadoop 2014). Therefore, MapReduce-style architecture can help relieve the input/output (I/O) and networking bottleneck to some degree. Additionally, when a new node is added into the cluster, the system gains the space of the hard drive and the power of the new processor.

So how does MapReduce work? It has two primary procedures (Fig. 2), the Mapper and the Reducer. A MapReduce job usually splits the input dataset into independent blocks which in turn are processed by the Mapper tasks in parallel. The framework sorts the outputs of the maps, which are then input to the Reducer tasks. Typically, both the input and the output of the job



a

```
public void map(LongWritable key, Text value,
OutputCollector<Text, IntWritable> output, Reporter reporter)
throws IOException {
      String line = value.toString();
      StringTokenizer tokenizer = new  StringTokenizer(line);
      while (tokenizer.hasMoreTokens()) {
           word.set(tokenizer.nextToken());
          output.collect(word, one);
      }
}
```

Mapper implementation

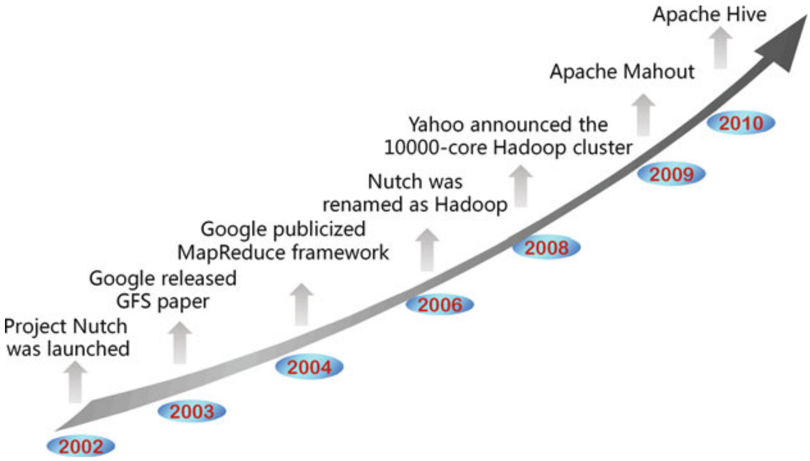
b

```
public void reduce(Text key, Iterator<IntWritable> values,
OutputCollector<Text, IntWritable> output, Reporter reporter)
throws IOException {
         int sum = 0;
        while (values.hasNext()) {
                 sum += values.next().get();
        }
       output.collect(key, new IntWritable(sum));
}
```

Reducer implementation

MapReduce, Fig. 3 An example of MapReduce application that counts the number of occurrences of each word in a given input set. (**a**) Mapper implementation. (**b**) Reducer implementation (Source: Hadoop 2014)

MapReduce, Fig. 4 Evolution of Hadoop MapReduce


Apache Hive
Apache Mahout
Yahoo announced the 10000-core Hadoop cluster
Nutch was renamed as Hadoop
Google publicized MapReduce framework
Google released GFS paper
Project Nutch was launched
2002
2003
2004
2006
2008
2009
2010

are stored in the HDFS. The framework is able to take care of scheduling and monitoring tasks and re-execute any failed tasks (Hadoop 2014). Figure 3 shows an example of a MapReduce application that counts the number of occurrences of each word in a given input set using Java programming. The Mapper implementation (Fig. 3a) processes one line in a document as provided in text format at a time. It then splits the line into tokens separated by whitespaces and creates a key-value pair of <<word>, 1>. The Reducer implementation (Fig. 3b) then sums up the values, which are the occurrence counts for each key (i.e., words in this example).

Historical Background

Started in 2006, Hadoop is now almost 10 years old. Figure 4 illustrates the historical evolution of Hadoop MapReduce. Internet Archive search director (Doug Cutting) launched a project called Nutch with the intention of building a better open-source search engine in 2002. This project planted the seeds of Hadoop, and Nutch was able to crawl and index hundreds of millions of pages after 1 year. In October 2003, Google released the Google File System (GFS) paper (Ghemawat et al. 2003). The MapReduce framework was publicized by Google at the Sixth Symposium on Operating System Design and Implementation conference in 2004 (Dean and Ghemawat 2004). In 2006, Cutting went to work with Yahoo and spun out the storage and processing parts of Nutch to form Hadoop as an open-source Apache Software Foundation project. In the same year, the Nutch DFS was renamed as HDFS.

In 2008, Yahoo announced that a 10,000 core Hadoop cluster was used to run its Yahoo! Search Webmap, producing data that are now used in every Yahoo! web search query. Most recently, many other Apache projects have been launched, such as Mahout (released in March 2009) and Hive (released in October 2010), to bring large-scale data analysis one step closer to the average users by offering MapReduce programming framework. Mahout is an open-source machine-learning package, and many classic algorithms for data mining, such as naïve Bayes and logistic regression are implemented using the Apache Hadoop platform. In practice, Hive has been widely adopted as a warehousing solution by many enterprises, including Facebook.

As a distributed processing framework and parallel programming model with good scalability and fault tolerance, MapReduce becomes a promising solution to support various GIS operations and analysis (Chen et al. 2008; Huang et al. 2015; Cao et al. 2015). For example, Chen et al. (2008) proposed a high-performance workflow system MRGIS, a distributed computing platform based on MapReduce clusters, to support GIS applications efficiently. Huang et al. (2015) presented a CyberGIS framework based on Hadoop system to synthesize multisourced data (e.g., social media, socioeconomic data) for disaster management. Cao et al. (2015) also presented a scalable computational framework using an Hadoop cluster to process social media data for efficient and systematic spatiotemporal data analysis. To support massive spatial queries, the MapReduce framework has also been increasingly used. For example, Hadoop-GIS (Aji et al. 2013), extending Hive, can handle multiple types of spatial queries and parallel spatial query execution on top of MapReduce.

Scientific Fundamentals

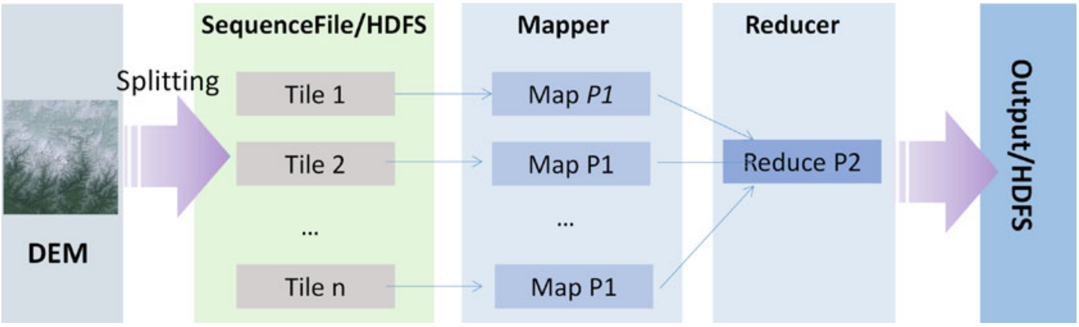
The true value of the MapReduce framework lies in its ability to run the tasks in parallel while balancing storage, central processing unit (CPU), and I/O evenly across each computing node in a computing cluster (http://java.dzone.com/articles/evolution-mapreduce-and-hadoop). In this regard, the success of such a framework relies on the development of a distributed file system (DFS) that conjures an army of cheap hardware to provide cost-efficient, fault-torrent, and scalable data storage service. By splitting the data into multiple tiles and replicating across the nodes, DFS technologies meet the requirements of large-scale and high-performance concurrent access of data. They (e.g., HDFS) therefore have become a foundation of and are utilized in conjunction with Parallel Data Processing Frameworks (e.g., MapReduce) for big data storage, management, and analysis. However, applications developed with MapReduce framework require quick response time. Correspondingly, improving the performance of MapReduce tasks is of great significance in practice and has attracted more and more attentions from both academia and industry (Gu et al. 2014). In fact, much progress has been made in the performance improvement of Hadoop from different perspectives, such as job scheduling, memory issues, I/O bottlenecks, etc. (Gu et al. 2014; Jiang et al. 2015).

Key Applications

MapReduce frameworks, e.g., Hadoop, can easily scale data processing over multiple computing nodes and therefore have been used extensively in different applications that include terabyte sorting, machine learning, and graph processing. With the massive amounts of spatial data accumulated daily, there is an increasing demand to take advantage of such a framework in spatial data processing and analysis. Specific applications include the following.

Geospatial Processing: An Example of Digital Evaluation Model Interpolation

Digital Evaluation Model (DEM) refers to the digital representation of ground surface topography or terrain. A DEM can be represented as a raster (a grid of cells) or vector (a triangular irregular network) in a geographic information system (Audenino et al. 2001). The interpolation of DEMs for large geographic areas can encounter challenges in practical applications



MapReduce, Fig. 5 DEM MapReduce processing

such as terrain visualization, especially in web applications, where a fast response is required. Additionally, computational demands exceed the capacity of a traditional single processing unit performing serial processing (Huang and Yang 2011). Therefore, we typically divide the DEM domain into multiple subdomains and leverage different computing resources to process those subdomains in parallel.

Figure 5 shows the workflow of using the MapReduce processing model to interpolate a large domain DEM. Spatial data decomposition, in which data are split into multiple tiles or subdomains, is an essential initial step to achieve parallelization. Each tile is then stored in the HDFS in the format of SequenceFiles, which are flat files consisting of binary key/value pairs. A mapping process (P1) is then used to interpolate each tile using an interpolation algorithm (e.g., Inverse Distance Weighted). A reducing process (p2) is applied to aggregate the interpolated tiles into a single file as the interpolation result.

Remote Sensing Image Processing and Analysis

With the rapid improvement of data acquisition technologies, the number of available remote sensing images is growing exponentially. Correspondingly, parallel computing and distributed systems have been widely leveraged to process massive high-resolution remote sensing images to derive customized products. Examples of remote sensing processing and analysis using the MapReduce framework include Sobel filtering, image resizing, image format conversion, auto contrasting, image sharpening, picture embedding, text embedding, image quality inspection, k-means clustering, and gridding problem (Lv et al. 2010; Almeer 2012; Golpayegani and Halem 2009).

Spatial Data Warehouse and Spatial Query

The past several years have witnessed a new trend shifting away from deploying analytical databases on high-end proprietary machines and moving toward cheaper, lower-end, commodity hardware, typically arranged in a shared-nothing massively parallel processing (MPP) architecture (Abouzeid et al. 2009). Such an architecture can leverage hundreds to thousands of machines to perform data analysis in parallelMapReduce-style architecture that has recently emerged as a scalable and cost-effective solution for MPP. As a result, much research has been devoted to integrating both the MapReduce framework and database management system (DBMS) technologies for analytical workloads. For example, the open-source Hive project aims to integrate declarative query constructs from the database community into MapReduce-like software to allow greater data independence, code reusability, and automatic query optimization (Hive 2014). Within the Hive system, data queries are expressed in a SQL-like declarative language, called HiveQL, and transformed into MapReduce tasks. HadoopDB (Abouzeid et al. 2009) integrated Hadoop and open-source DBMS software for data analysis, achieving the performance and efficiency of parallel databases yet still yielding the scalability, fault tolerance, and flexibility of MapReduce-based systems.

Big data derived from spatial applications have been a long-existing issue in geospatial fields. In the last several years, geospatial communities have begun to acknowledge the benefit of integrating MapReduce when building spatial data warehousing systems and high-performance spatial querying engines for data- and computation-intensive spatial applications, such as location-based services (Aji et al. 2013; Eldawy and Mokbel 2013). SpatialHadoop (Eldawy and Mokbel 2013), for example, is a comprehensive extension to Hadoop that pushes spatial data inside the core functionality of Hadoop. SpatialHadoop employs a simple spatial high-level language, a two-level spatial index structure, basic spatial components built inside the MapReduce layer, and three basic spatial operations: range queries, k-NN queries, and spatial join.

Future Directions

In the era of digital world, huge volumes of data need to be stored and analyzed daily. With the benefit of scalable and cost-effective data processing, mining, and analysis, the MapReduce framework offers a potential solution to address the computing challenges posed by big data. In spite the great success of MapReduce, we still need to consider the few constraints of Hadoop MapReduce model. One widely acknowledged constraint is that a MapReduce cluster consists of a set of computing nodes with each node including several CPUs and connected with 1Gb/s network. In order to achieve high performance, hundreds or thousands of nodes (e.g., Yahoo! Search Webmap running on a 10,000 core Hadoop cluster) must be used. This leads to a substantial up-front investment required to build a private large-scale MapReduce cluster, plus high ongoing power consumption costs (https://sites.google.com/site/mapreduceongpu/home/why-how). One approach to tackle this constraint is to leverage graphics processing unit (GPU) computing technology, which has become popular in the past few years (Nickolls and Dally 2010). While CPUs consist of a few cores optimized for serial processing, GPUs consist of thousands of smaller, more efficient cores designed for parallel performance. With its notable parallel computing capabilities for processing large-volume data, GPU computing has begun to be used in geocomputation (Li et al. 2013). Therefore, more research should be done on exploring and implementing the MapReduce framework on GPUs for geospatial applications.

Many open-source tools and packages have been developed over MapReduce to store, access, and mine big data. Examples include Pig, Hive, and Mahout, to name just a few. However, common tools and libraries dedicated to spatial data processing using MapReduce are much less common and available among the geospatial communities. Additionally, only limited geospatial applications have been developed to leverage the MapReduce framework. Therefore, much effort should be devoted to identifying applications of massive impact and of fundamental importance and requiring the latest parallel programming model. Additionally, traditional GIS operations and functions that can be parallelized by the data decomposition method and have been previously supported by using programming models such as MPIs and multi-threads can also be redesigned to support MapReduce-based framework. The development procedures and tools should be documented and shared as a reference among geospatial scientists when they are seeking MapReduce solutions for specific applications.

Cross-References

- ▶ Cuda/GPU
- ▶ GIS
- ▶ Medical Image Dataset Processing over Cloud/MapReduce with Heterogeneous Architectures

References

Abouzeid A, Bajda-Pawlikowski K, Abadi D, Silberschatz A, Rasin A (2009) HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads. Proc VLDB Endow 2(1):922–933. doi:10.14778/1687627.1687731, http://dx.doi.org/10.14778/1687627.1687731

Aji A, Wang F, Vo H, Lee R, Liu Q, Zhang X, Saltz J (2013) Hadoop GIS: a high performance spatial data warehousing system over mapreduce. Proc VLDB Endow 6(11):009–1020

Almeer MH (2012) Cloud Hadoop map reduce for remote sensing image analysis. J Emerg Trends Comput Inf Sci 3(4):637–644

Cao G, Wang S, Hwang M, Padmanabhan A, Zhang Z, Soltani K (2015) A scalable framework for spatiotemporal analysis of location-based social media data. Comput Environ Urban Syst 51:70–82

Chen Q, Wang L, Shang Z (2008) MRGIS: a MapReduce-enabled high performance workflow system for GIS. In: IEEE fourth international conference on eScience, eScience'08, Indianapolis, 7–12 Dec 2008. IEEE, pp 646–651

Dean J, Ghemawat S (2004) MapReduce: simplified data processing on large clusters. In: Proceedings of the sixth symposium on operating system design and implementation, San Francisco, Dec 2004, pp 137–150

Eldawy A, Mokbel MF (2013) A demonstration of spatial-hadoop: an efficient mapreduce framework for spatial data. Proc VLDB Endow 6(12):1230–1233

Ghemawat S, Gobioff H, Leung ST (2003) The Google file system. ACM SIGOPS Oper Syst Rev 37(5): 29–43

Golpayegani N, Halem M (2009) Cloud computing for satellite data processing on high end compute clusters. In: Proceedings of IEEE 2009 international conference on cloud computing, 21–25 Sept 2009, Bangalore, pp 88–92

Gu R, Yang X, Yan J, Sun Y, Wang B, Yuan C, Huang Y (2014) SHadoop: improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters. J Parallel Distrib Comput 74(3):2166–2179

Hadoop (2014) Apache Hadoop. Acquired from http://hadoop.apache.org/

Hive (2014) Apache Hive. Acquired from http://hive.apache.org/

Huang Q, Yang C (2011) Optimizing grid configuration to support geospatial processing – an example with DEM interpolation. Comput Geosci 37(2):165–176

Huang Q, Yang C, Benedict K, Rezgui A, Xie J, Xia J, Chen S (2013) Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting. Int J Geogr Inf Sci 27(4):765–784

Huang Q, Cervone G, Jing D, Chang C (2015) DisasterMapper: a CyberGIS framework for disaster management using social media data. In: ACM SIGSPATIAL international workshop on analytics for big geospatial data, Seattle. ACM

Jiang H, Chen Y, Qiao Z, Weng T-H, Li K-C (2015) Scaling up mapreduce-based big data processing on multi-GPU systems. Clust Comput 18(1):369–383

Li J, Jiang Y, Yang C, Huang Q, Rice M (2013) Visualizing 3D/4D environmental data using many-core graphics processing units (GPUs) and multi-core central processing units (CPUs). Comput Geosci 59:78–89. doi:j.cageo.2013.04.029

Lv Z, Hu Y, Zhong H, Wu J, Li B, Zhao H (2010) Parallel K-means clustering of remote sensing images based on mapreduce. In: Web information systems and mining. Springer, Berlin/Heidelberg, pp 162–170

Nickolls J, Dally WJ (2010) The GPU computing era. IEEE Micro 30(2):56–69

Recommended Reading

Audenino P, Rognant L, Chassery JM, Planes JG (2001) Fusion strategies for high resolution urban DEM. In: Proceedings of the IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (Cat. No.01EX482), Rome, Italy, pp. 90-94. IEEE: New Jersey

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1):107–113

Zhao J, Tao J, Streit A (2014) Enabling collaborative MapReduce on the cloud with a single-sign-on mechanism. Computing, 98(1-2):55–72

Maps on Internet

▶ Web Mapping and Web Cartography

Maps, Animated

▶ Geographic Dynamics, Visualization and Modeling

MapServ

▶ University of Minnesota (UMN) Map Server

MapServer

▶ Quantum GIS
▶ Web Feature Service (WFS) and Web Map Service (WMS)

MapWindow GIS

Daniel P. Ames[1], Christopher D. Michaelis[1],
Allen Anselmo[1], Lailin Chen[2], and
Harold Dunsford[2]
[1]Department of Geosciences, Geospatial
Software Lab, Idaho State University, Pocatello,
ID, USA
[2]Department of Geosciences, Idaho State
University, Pocatello, ID, USA

Synonyms

ActiveX components; Free GIS; .NET framework; Open-source GIS; Programmable GIS components

Definition

MapWindow GIS is an open source geographic information system that includes a desktop application and a set of programmable mapping and geoanalytical components. Because it is distributed as open source software under the Mozilla Public License, MapWindow GIS can be reprogrammed to perform different or more specialized tasks and can be extended as needed by end users and developers. MapWindow GIS has been adopted by the United States Environmental Protection Agency as a platform for its BASINS watershed analysis system and is downloaded over 3000 times per month by end users who need a free GIS data viewer and programmers who need tools for commercial and non-commercial software applications.

Main Text

MapWindow GIS is a desktop open source GIS and set of programmable objects intended to be used in the Microsoft Windows operating system. Because it is developed using the Microsoft .NET Framework, it is optimized for the Windows environment and can be extended by programmers using the Visual Basic and C# languages. The MapWindow GIS desktop application plug-in interface supports custom tool development by end users and the MapWindow Open Source team. Additionally, software developers can use the core MapWindow ActiveX and .NET programming components to add GIS mapping and geoprocessing functionality to custom standalone applications.

MapWindow GIS has been adopted by the United States Environmental Protection Agency, United Nations University and others as a development and distribution platform for several environmental models including BASINS/HSPF, FRAMES-3MRA and SWAT. Other users and developers have modified and applied MapWindow GIS for use in the fields of transportation, agriculture, community planning and recreation. MapWindow GIS is continually maintained by an active group of nearly 50 student and volunteer developers from around the world who regularly release updates and bug fixes through the www.MapWindow.org web site.

Cross-References

▶ Open-Source GIS Libraries

Marginalia

▶ Metadata and Interoperability, Geospatial

Market Behavior

▶ Financial Asset Analysis with Mobile GIS

Market Efficiency

▶ Financial Asset Analysis with Mobile GIS

Market Intelligence

▶ Mobile Marketing

Market-Basket Analysis

▶ Frequent Itemset Discovery

Marketing Information System

▶ Decision-Making Effectiveness with GIS
▶ Mobile Marketing

Markov Random Field (MRF)

▶ Hurricane Wind Fields, Multivariate Modeling

Mass Migration

▶ Climate and Human Stresses on the Water-Energy-Food Nexus

Massive Agent-Based Systems (MABS)

▶ High-Performance Agent-Based Geo-Spatial Modeling and Simulation

Massive Evacuations

▶ Emergency Evacuation Plan Maintenance

Mathematical Foundations of GIS

Timothy G. Feeman
Department of Mathematical Sciences, Villanova University, Villanova, PA, USA

Definition

- *Geodesy* is the branch of mathematics concerned with the shape and area of the earth and with the location of points on it.
- *Cartography* is the art, science, and practice of making maps.
- A *map projection*, or simply a projection, is any systematic representation of the earth's surface onto another surface.
- The *Global Positioning System*, or GPS, comprises a network of satellites that orbit the earth and, by radio communications with land-based receivers, enable the accurate determination of the coordinates of points on the earth's surface.
- *Spherical geometry* is the study of lines, angles, and areas on a spherical surface.

Historical Background

The foundation of geographical information science (GIS) lies in our ability to determine the size and shape of the earth, locate points on its surface, measure its features, and to portray the earth in maps. Thus, geodesy and cartography form the basis of GIS. In turn, both of these subjects are built on strong mathematical foundations.

The Shape and Size of the Earth

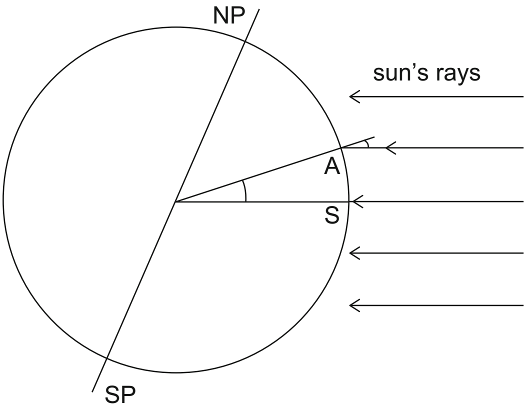
In this age of space exploration, photographs of Earth and other heavenly bodies taken from space offer convincing evidence that Earth's basic shape is spherical. This conception of a spherical Earth has endured, though not without some lapses, at least since the sixth century B.C., when Anaximander and Thales of Miletus, two of the earliest classical Greek geometers, described the earth as a sphere positioned at the center of a huge celestial sphere to which were fixed the other visible planets and stars.

Towards the end of the seventeenth century, Isaac Newton's work on gravitation and planetary motion led him to conclude that Earth had the shape of an ellipsoid, flattened at the poles and somewhat bulging around the equator. Newton's conjecture was confirmed by French-sponsored expeditions to Peru and Lapland during the 1730s in which arcs of meridians at high and low latitudes were measured. In the early 1800s, Gauss and others provided further verifications. Since the introduction of satellite technology, new measurements have resulted in the development of several reference ellipsoids, including the World Geodetic Systems ellipsoid of 1984 (WGS 84, for short), developed by the US Defense Mapping Agency, and the Geodetic Reference System ellipsoid of 1980 (GRS 80), adopted by the International Union of Geodesy and Geophysics in 1979.

When the highest level of precision is needed, an ellipsoid provides the best mathematical model for the earth's shape. In this article, for simplicity's sake, a spherical Earth is assumed, with measurements and calculations made accordingly.

Assuming the earth to be a sphere, the single most important measurement is its circumference, which was estimated by Eratosthenes of Alexandria in roughly 230 B.C.

Eratosthenes knew that, at noon on any given day of the year, the angle of the sun above the horizon would be different at two different places located north and south of one another. Assuming the earth to be a sphere and all of the sun's rays to be parallel, Eratosthenes called upon a basic geometric fact – that a line transversal to two parallel lines will make equal angles with both – to conclude that the difference between the angles of the sun would correspond to the central angle of the portion of the earth's circumference between the two points (Fig. 1). He also knew that, at noon on the summer solstice, the sun shone directly overhead in the town of Syene (now Aswan, Egypt), famously illuminating a well there. Eratosthenes then determined that, on the summer solstice, the noonday sun in Alexandria was one-fiftieth part of a full circle (7.2°) shy of being directly overhead. It followed that the distance between Syene and Alexandria must be one-fiftieth part of a full circumference of the earth. The distance between the two towns was measured to be about 5000 stadia (approximately 500 miles). Multiplication by 50 yielded Eratosthenes' remarkably accurate estimate of 250,000 stadia, about 25,000 miles, for the circumference of the earth. The basic method employed by Eratosthenes is completely valid and is still used today.



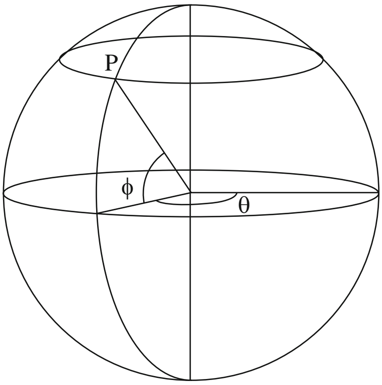
Mathematical Foundations of GIS, Fig. 1 The sun is overhead at Syene (S). The angle of the sun at Alexandria (A) is the same as the central angle

In fact, the equatorial and polar radii and circumferences of the earth are different, because of the earth's ellipsoidal shape. The mean radius of the earth is approximately 6371 km.

Scientific Fundamentals

Location

The latitude of any given point is defined to be the difference between the angles made by the sun at noon at the point in question and at the equator. This is illustrated in Fig. 2. Latitude is designated as north (N) or south (S) according to which hemisphere the point lies in. The points at the same latitude form a circle whose plane is parallel to that of the equator. Thus, a circle of latitude is called a *parallel*.



Mathematical Foundations of GIS, Fig. 2 The point P has longitude θ and latitude ϕ

As the earth rotates about its axis, the position of the sun in the sky changes, ascending from the eastern horizon at dawn to its zenith at noon, then descending until it sets in the west. The arrival of local solar noon, the moment at which the sun reaches its zenith, is a simultaneous event at all points along a semi-circular arc, called a *meridian*, that extends from the north pole to the south pole. Where two meridians come together at the poles, they form an angle that is the basis for determining *longitude*. The difference in longitudes of two locations is the portion of a full circle through which the earth rotates between the occurrences of local solar noon at the two places. Thus, the measurement of longitude is fundamentally a problem of the measurement of time.

One way to measure time differences between two places is by the relative positions in the sky of various celestial bodies, such as certain stars, the planets, or the moons of Jupiter. The astrolabe and the sextant were among the tools developed to measure these positions, and elaborate tables were compiled showing known positions of celestial objects. Alternatively, longitude can be determined using two clocks, one set to the time at a fixed location and the other to local time. Though sufficiently accurate clocks are readily available today, their introduction just a few centuries ago marked a giant leap in the technology of navigation. The first sea-worthy chronometer was developed in England by John Harrison and presented to the English Longitude Board in 1735, though it took some years of refinements in size, weight, and ease of reproduction for the new clocks to catch on.

In addition to the technical problem of measuring time differences, there is the political issue of deciding which meridian will serve as the reference, or prime meridian, for longitude calculations. Ptolemy placed his prime meridian through the Canary Islands. Others have used the meridians through Mecca, Jerusalem, Paris, Rome, Copenhagen, the Cape Verde Islands, St. Petersburg, Philadelphia, and more. After 1767, when the Royal Observatory in Greenwich, England, published the most comprehensive tables of lunar positions available, sailors increasingly calculated their longitude from Greenwich. This practice became official in 1884 when the International Meridian Conference established the prime meridian at Greenwich. Longitude is designated as east (E) or west (W) according to whether local noon occurs before or after local noon in Greenwich.

Latitude and longitude together give a complete system for locating points on the earth's surface.

Coordinates

Where cartographers often measure angles in degrees, here **angles will be measured in radians** in order to simplify trigonometric computations. Also, positive and negative angle measurements will be used instead of the directional designations north/south or east/west, with south and west assigned negative values. The symbols θ and ϕ will denote longitude and latitude, respectively. Thus, the point with longitude 75°W and latitude 40°N has coordinates ($\theta = -75\pi/180$, $\phi = 40\pi/180$) while the point with longitude 75°E and latitude 40°S has coordinates ($\theta = 75\pi/180$, $\phi = -40\pi/180$).

For the Cartesian coordinate system in three-dimensional space, the line through the north and south poles will be taken as the z-axis, with the north pole on the positive branch. The plane of the equator corresponds to the xy-plane with the positive x-axis meeting the equator at the prime meridian and the positive y-axis meeting the equator at the point with longitude $\pi/2$ (or 90°E). On a sphere of radius R, then, the point with longitude θ and latitude ϕ will have Cartesian coordinates $(x, y, z) = (R\cos(\theta)\cos(\phi),\ R\sin(\theta)\cos(\phi),\ R\sin(\phi))$. Conversely, latitude and longitude can be recovered from the Cartesian coordinates. The equation $\phi = \arcsin(z/R)$ determines ϕ uniquely as an angle between $-\pi/2$ and $\pi/2$. Also, θ satisfies the equations $\tan(\theta) = y/x$, $\cos(\theta) = x/\sqrt{x^2+y^2}$, and $\sin(\theta) = y/\sqrt{x^2+y^2}$. Any two of these together will determine a unique angle θ between $-\pi$ and π.

Distance

The distance between two points is the length of the shortest path connecting the two points. On a sphere, the path must lie entirely on the surface and, so, cannot be a straight line segment as it is in a plane. Instead, the shortest path is the straightest possible one which, intuitively, is an arc of the largest possible circle, called a great circle.

A *great circle* on a sphere is the intersection of the sphere with a plane that contains the center of the sphere. Every great circle has a radius equal to that of the sphere. The equator is a great circle while each meridian is half of a great circle. Any two great circles must intersect each other at a pair of antipodal points. Indeed, the planes defined by the circles will intersect in a line through the sphere's center, which, therefore, will intersect the sphere at two opposite points.

Any two non-antipodal points on the sphere determine a unique great circle, namely the intersection of the sphere with the plane generated by the two points together with the center of the sphere. The two points divide this circle into two arcs, the shorter of which is the shortest path connecting the points. If O denotes the center of a sphere of radius R and by A and B the two points of interest, then the distance between A and B along the shorter great circle arc is equal to the product of R and the central angle, measured in radians, formed by the two vectors $\overrightarrow{OA}$ and $\overrightarrow{OB}$. (This is essentially the definition of radian measure.)

To derive a formula for distance in terms of the longitudes and latitudes of the points in question, assume for simplicity's sake that the sphere has radius $R = 1$ unit and let the points A and B have longitudes θ_1 and θ_2 and latitudes ϕ_1 and ϕ_2, respectively. Converting to three-dimensional Cartesian coordinates,

$$\begin{aligned}A &= (\cos\phi_1\ \cos\theta_1,\ \cos\phi_1 \sin\theta_1,\ \sin\phi_1) \text{ and}\\ B &= (\cos\phi_2\ \cos\theta_2,\ \cos\phi_2 \sin\theta_2,\ \sin\phi_2).\end{aligned}$$

From vector geometry, the *cosine* of the angle between the two vectors $\overrightarrow{OA}$ and $\overrightarrow{OB}$ is given by their dot product divided by the product of the vectors' lengths, each of which is 1 in this case. The value of the dot product is

$$\begin{aligned}&\overrightarrow{OA} \bullet \overrightarrow{OB}\\ &\quad = \cos\phi_1 \cos\theta_1 \cos\phi_2 \cos\theta_2\\ &\qquad + \cos\phi_1 \sin\theta_1 \cos\phi_2 \sin\theta_2\\ &\qquad + \sin\phi_1 \sin\phi_2\end{aligned}$$

$$\begin{aligned}&= \cos\phi_1\ \cos\phi_2\ (\cos\theta_1\ \cos\theta_2 + \sin\theta_1\ \sin\theta_2)\\ &\quad + \sin\phi_1\ \sin\phi_2\\ &= \cos\phi_1\ \cos\phi_2\ \cos(\theta_1 - \theta_2) + \sin\phi_1\ \sin\phi_2\,.\end{aligned}$$

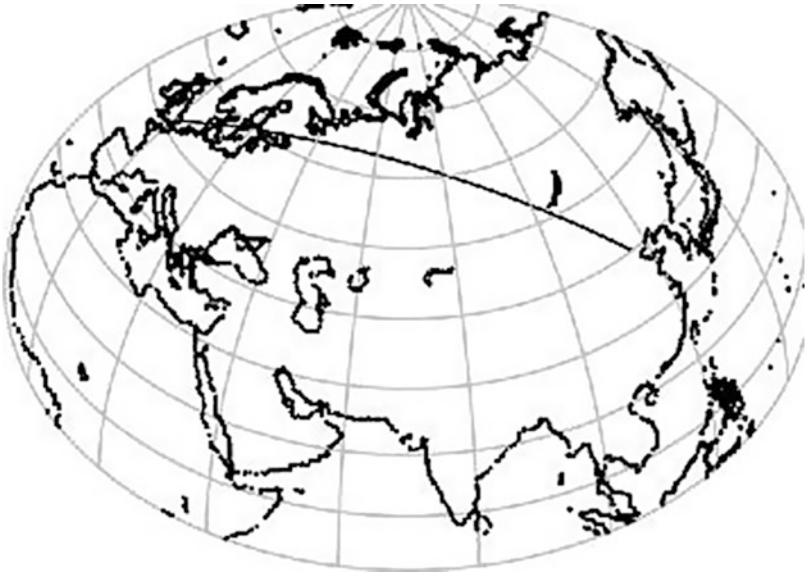
The angle between the vectors is then

$$\begin{aligned}\text{angle} &= \arccos(\overrightarrow{OA} \bullet \overrightarrow{OB})\\ &= \arccos(\cos\phi_1\ \cos\phi_2\ \cos(\theta_1 - \theta_2)\\ &\quad + \sin\phi_1\ \sin\phi_2). \qquad (1)\end{aligned}$$

Now multiply this angle by the radius of the sphere to get the distance. That is,

$$\begin{aligned}\text{distance} = R\arccos(&\cos\phi_1\ \cos\phi_2\ \cos(\theta_1 - \theta_2)\\ &+ \sin\phi_1\ \sin\phi_2). \qquad (2)\end{aligned}$$

For example, London, England, has longitude $\theta_1 = 0$ and latitude $\phi_1 = 51.5\pi/180$, while $\theta_2 = 116.35\pi/180$ and $\phi_2 = 2\pi/9$ are the coordinates of Beijing. Therefore, the central angle between London and Beijing is about 1.275 radians. The mean radius of the earth is about $R = 6371$ km and, hence, the great circle distance from London to Beijing is approximately $1.275R$, or 8123 km. The great circle route is illustrated in Fig. 3.



Mathematical Foundations of GIS, Fig. 3 Great circle route from London to Beijing

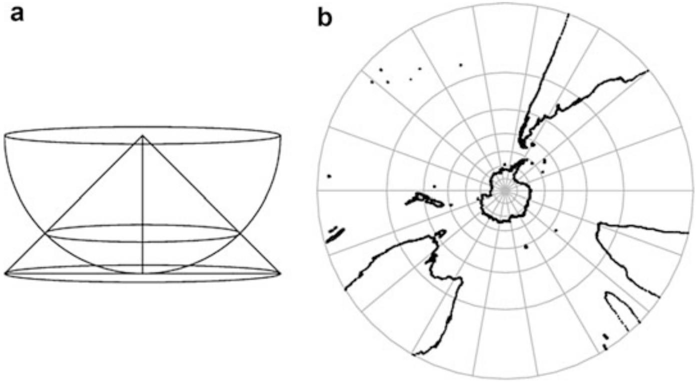
The most important map projection for the depiction of great circle routes is the gnomonic projection, which is constructed by projecting a spherical globe onto a plane tangent to the globe using a light source located at the globe's center. Any two points on the sphere, along with the light source for the projection, define a plane that intersects the sphere in a great circle and the plane of the map in a straight line, which is, therefore, the image of the great circle joining the points. In other words, the gnomonic projection has the property that the shortest route connecting any two points A and B on the sphere is projected onto the shortest route connecting their images on the flat map.

The gnomonic projection was most likely known to Thales of Miletus and would have been particularly useful to navigators and traders of the Golden Age of Greece, a time of intensified trade and geographical discovery during which the Bronze Age gave over to the Age of Iron.

The gnomonic projection can be constructed from elementary geometry. Place a globe of radius R on a flat piece of paper with the south pole at the bottom and with a projecting light source at the center of the globe. Points on or above the equator won't project onto the paper, so the map will show only the southern hemisphere. Arrange the map's coordinate axes so that the prime meridian is projected onto the positive x-axis, in which case the image of the meridian at longitude θ makes an angle of θ, measured counterclockwise, with the positive x-axis. The parallels, meanwhile, will be shown on the map as concentric circles having the pole as their common center. When the globe is viewed from the side, as in Fig. 4, two similar right triangles can be seen. Each has the center of the globe as a vertex. The horizontal side of the smaller triangle is a radius of the parallel at latitude ϕ. Hence, the vertical and horizontal sides of the smaller triangle have lengths $R\cos(\pi/2+\phi)$ and $R\sin(\pi/2+\phi)$, respectively. (Note that $\phi < 0$ in this context.) The vertical side of the larger triangle is a radius of the globe, so its length is R. Let $r(\phi)$ denote the length of the horizontal side of the larger triangle, which is the radius of the projected image of the parallel at ϕ. The proportionality of sides for similar triangles yields the equation

$$\frac{r(\phi)}{R} = \frac{R\sin(\pi/2+\phi)}{R\cos(\pi/2+\phi)}\,,\text{ whence}$$

$$\begin{aligned} r(\phi) &= \frac{R^2\sin(\pi/2+\phi)}{R\cos(\pi/2+\phi)} = R\tan(\pi/2+\phi) \\ &= -R\cot(\phi)\,. \end{aligned} \tag{3}$$



Mathematical Foundations of GIS, Fig. 4 The gnomonic projection. The basic construction is depicted (**a**), and the resulting map of most of the southern hemisphere (**b**)

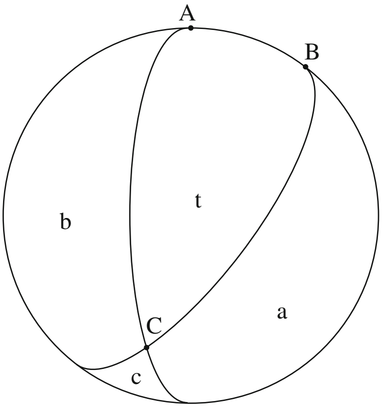
A base grid for a gnomonic projection of most of the southern hemisphere can now be constructed, either on a computer or by hand using a protractor, ruler, and compass. The resulting map is shown in Fig. 4.

Spherical Triangles

A spherical triangle is formed when arcs of three different great circles meet in pairs. For a planar triangle, the sum of the three interior angles is always π radians. For a spherical triangle, however, the sum of the interior angles is directly related to the size of the triangle. For instance, two points on the equator and a third point near the equator define a spherical triangle that almost fills up a hemisphere. Each angle will be almost π, so the sum of the angles will be just under 3π. On the other hand, a small triangle will be nearly flat, so its angles will add up to a number close to π. The exact relationship between the area of a spherical triangle and the sum of its angles is expressed in the formula

$$\begin{aligned}&\text{spherical triangle area}\\&\quad = R^2\ (\text{sum of the angles} - \pi)\,.\end{aligned} \tag{4}$$

To prove formula (4), consider a spherical triangle with corners at A, B, and C, where, with no loss of generality, the edge $\widehat{AB}$ lies on the horizon with A at the north pole and C is in the front hemisphere. Extend the arcs $\widehat{AC}$ and $\widehat{BC}$ to divide the hemisphere into four parts, as depicted in Fig. 5, with areas labeled as a, b, c, and t. The area of the triangle is t.



Mathematical Foundations of GIS, Fig. 5 The *spherical triangle* with vertices at A, B and C has area t. The areas a, b, c, and t together fill up a hemisphere. Also, each of the areas a, b, and c forms a lune when combined with t

When the arcs $\widehat{AB}$ and $\widehat{AC}$ are extended into semicircles, they intersect at the antipode to A and enclose a portion of the sphere's surface, called a *lune*, whose area is $a + t = (\angle BAC/\pi)\,(2\pi R^2) = 2\,(\angle BAC)\,R^2$. Similarly, one can form two other lunes with areas $b + t = 2\,(\angle ABC)\,R^2$ and $c + t = 2\,(\angle BCA)\,R^2$. Note that the lune formed when the arcs $\widehat{CA}$ and $\widehat{CB}$ are extended actually consists of the region t together with a copy of region c on the back side of the sphere. Adding up these three areas, one has $a + b + c + 3t = 2R^2\,(\text{sum of the angles})$. But regions a, b, c, and t collectively form a hemisphere, so, also, $a + b + c + t = 2\pi R^2$. Subtract this from the previous equation to get $2t =$

$2R^2$ (sum of the angles) $- 2\pi R^2$, from which it follows that $t = R^2$ (sum of the angles $- \pi$), as was to be proved.

Classical Spherical Trigonometry

Two classical results from spherical trigonometry that have useful applications to problems such as measurement of distance and determination of azimuths are the Law of Cosines and the Law of Sines. Like their Euclidean counterparts, they relate the measurements of various parts of a triangle.

Referring again to the spherical triangle in Fig. 5, note that each side of the triangle, being an arc of a great circle, has length equal to R times the central angle formed by the vectors connecting the origin to the corresponding vertices. With this in mind, let α, β, and γ denote the central angles corresponding to the sides opposite the vertices A, B, and C, respectively. Thus, $\alpha := \widehat{BC}/R$, $\beta := \widehat{AC}/R$, and $\gamma := \widehat{AB}/R$.

Without loss of generality, assume that A is at the north pole, so that its latitude is $\phi_1 = \pi/2$ and its longitude θ_1 is arbitrary. The points B and C have generic coordinates $B(\theta_2, \phi_2)$ and $C(\theta_3, \phi_3)$. An application of formula (1) above yields

$$\begin{aligned} \cos(\alpha) &= \cos(\phi_2)\cos(\phi_3)\cos(\theta_2 - \theta_3) \\ &\quad + \sin(\phi_2)\sin(\phi_3), \\ \cos(\beta) &= \sin(\phi_3) \\ \text{since } \phi_1 &= \pi/2, \text{ and} \\ \cos(\gamma) &= \sin(\phi_2) \\ \text{since } \phi_1 &= \pi/2\,. \end{aligned}$$

Moreover, the angle in the triangle itself at vertex A is $\angle BAC = (\theta_2 - \theta_3)$, the difference in the longitudes of B and C.

It follows that $\sin(\beta) = \sqrt{1-\cos^2(\beta)} = \cos(\phi_3)$ and that $\sin(\gamma) = \sqrt{1-\cos^2(\gamma)} = \cos(\phi_2)$. Hence,

$$\begin{aligned} \cos(\alpha) &= \cos(\beta)\cos(\gamma) \\ &\quad + \sin(\beta)\sin(\gamma)\cos(\angle BAC)\,. \end{aligned} \tag{5}$$

This is the Law of Cosines for spherical trigonometry.

As an example, consider the spherical triangle formed by the north pole (A), Beijing (B), and London (C). As was shown above, $\alpha = \widehat{BC}/R \approx 1.275$ radians. Moreover, from the latitudes of Beijing and London, one has $\beta = \pi/2 - 51.5\pi/180 \approx 0.672$ radians and $\gamma = \pi/2 - 2\pi/9 \approx 0.873$ radians. The Law of Cosines (5) implies that

$$\begin{aligned} \cos(\gamma) &= \cos(\alpha)\cos(\beta) \\ &\quad + \sin(\alpha)\sin(\beta)\cos(\angle ACB)\,. \end{aligned}$$

The vertex angle at London is, therefore,

$$\begin{aligned} \angle ACB &= \arccos\left(\frac{\cos(\gamma) - \cos(\alpha)\cos(\beta)}{\sin(\alpha)\sin(\beta)}\right) \\ &\approx 0.8005 \text{ radians or } 45.866^\circ\,. \end{aligned}$$

Similarly, the vertex angle at Beijing is $\angle ABC \approx 0.6227$ radians, or 35.678°. As the angle at the north pole is $\angle BAC = 116.35\pi/180 - 0 \approx 2.031$ radians, it follows from formula (4) that the area of this spherical triangle is approximately $(6371)^2(2.031 + 0.8005 + 0.6227 - \pi) \approx 12{,}688{,}727$ square kilometers. This is not quite 2.5 % of the earth's surface.

The Law of Cosines (5) implies that

$$\cos(\angle BAC) = \frac{\cos(\alpha) - \cos(\beta)\cos(\gamma)}{\sin(\beta)\sin(\gamma)}\,.$$

Hence, using basic trigonometric identities, it follows that

$$\frac{\sin^2(\angle BAC)}{\sin^2(\alpha)} = \frac{\begin{array}{l} 1 - \cos^2(\alpha) - \cos^2(\beta) - \cos^2(\gamma) \\ \qquad + 2\cos(\alpha)\cos(\beta)\cos(\gamma) \end{array}}{\sin^2(\alpha)\sin^2(\beta)\sin^2(\gamma)}\,.$$

This last expression is symmetric in α, β, and γ and, therefore, the value of the left-hand side does not depend on the vertex chosen. That is,

$$\frac{\sin(\angle BAC)}{\sin(\alpha)} = \frac{\sin(\angle ABC)}{\sin(\beta)} = \frac{\sin(\angle ACB)}{\sin(\gamma)}, \tag{6}$$

which is the Law of Sines for spherical trigonometry.

Returning to the example of the triangle formed by the north pole, London, and Beijing, it has been shown already that $\angle BAC \approx 2.031$ radians, $\alpha \approx 1.275$ radians, and $\gamma = \pi/2 - \phi_2 \approx 0.873$ radians. Thus, by the Law of Sines (6),

$$\angle ACB = \arcsin\left(\frac{\sin(\angle BAC)\sin(\gamma)}{\sin(\alpha)}\right)$$
$$\approx 0.8005 \text{ radians or } 45.866^\circ .$$

Key Applications

The Global Positioning System

Perhaps the single most important aspect of mapping is the question of location. Indeed, a substantial portion of Ptolemy's classic treatise, *Geography*, which contains several of his most famous maps, is taken up with lists of the longitude and latitude coordinates of various places, much of this information having been gleaned from accounts of travelers. The latest, and to date the most accurate, method for determining the coordinates of a point on the earth is the Global Positioning System, or GPS.

Originally introduced by the United States Department of Defense in the 1980s, and made fully functional in 1995, the GPS consists of 24 solar-powered satellites that orbit the earth in nearly circular half-day orbits approximately 20,200 km above the earth's surface. There are four satellites in each of six distinct orbital planes with each plane inclined at an angle of 55° from the plane of the equator. The overall arrangement ensures that every point on earth is nearly always visible from at least four satellites. Each satellite is equipped with a highly accurate atomic clock and continuously transmits, via a microwave radio, the exact time of its internal clock, its precise position, and other secondary information.

A GPS receiver, located somewhere on the earth, can compute its position, as well as the exact time, by determining its distances from four of the satellites. The distance to each satellite is calculated by measuring the time delay between the transmission and the reception of the satellite's signal. In principle, this information locates the receiver at the intersection of four spheres, one centered at each of the four satellites. Since the receiver's position has only three coordinates in space, it would seem to suffice to use three satellites. However, errors in the receiver's clock should be treated as a fourth variable in the equations, thus necessitating the fourth satellite. With only three satellites, it is still possible to obtain a location at sea level.

The key to getting an accurate position is to measure the time delays accurately. The effects of the atmosphere on the propagation of the radio signals are significant in this regard. These effects depend on the transmission frequency, so one solution is to have the satellites broadcast simultaneously on two different frequencies. Using the difference in the time delays in the reception of the two signals, the GPS receiver can adjust for the atmospheric effects. Alternatively, since the effects will be similar for an entire region, a ground station can communicate an appropriate correction factor to GPS receivers in its area. Other factors that influence the transit time of the signals include drifts in the accuracy of the atomic clocks due to aging, radiation in space, power supply fluctuations, and even relativistic effects. For all of these factors, the cumulative effect is known for each satellite and is actually transmitted as part of the signal. Thus, the GPS receiver on the ground is able to account for these factors in its calculations.

So, for each value of the index $i = 1, 2, 3$, and 4, let t_i denote the adjusted measurement by the GPS receiver of the time delay in the reception of the signal from satellite number i. If the receiver's clock were synchronized with those of the satellites, then the distance to satellite i would be $c \cdot t_i$, where c is the speed of light. But, if the receiver's clock were off by b seconds, then even a tiny value of b would result in huge errors in distance measurements. Of course, the error in the receiver clock is not known ahead of time, thus it remains a variable in the calculations. In short, the receiver computes the distance to satellite i as $c(t_i + b)$.

Let (x_i, y_i, z_i) denote the position in space of the ith satellite at the instant it transmits its signal. Then the receiver is located at the point (x, y, z) whose coordinates satisfy the equations

$$(x-x_i)^2+(y-y_i)^2+(z-z_i)^2-c^2(t_i+b)^2 = 0 \tag{7}$$

for $i = 1, 2, 3,$ and 4. This system can be solved, to any desired degree of accuracy, by standard numerical methods such as least squares. Generally, at least ten digits must be computed. The computed value of b allows the GPS receiver to adjust its internal clock accordingly, though future computations will still assume the clock to be in error. The longitude and latitude of the receiver can be recovered from the Cartesian coordinates x, y, and z, as discussed earlier, while the elevation is the difference between $\sqrt{x^2 + y^2 + z^2}$ and sea level at the point (θ, ϕ).

To see how closely this procedure estimates the location of the GPS receiver, observe that, in the system (7), each of the unknowns x, y, and z will vary if the values of the t_i are allowed to vary. For instance, if x is viewed as a function of the t_i s, then, according to multivariable linear approximation, the differential dx is given by

$$\mathrm{d}x = \frac{\partial x}{\partial t_1}\mathrm{d}t_1 + \frac{\partial x}{\partial t_2}\mathrm{d}t_2 + \frac{\partial x}{\partial t_3}\mathrm{d}t_3 + \frac{\partial x}{\partial t_4}\mathrm{d}t_4 . \tag{8}$$

Similar formulas prevail for y and z. If $|\mathrm{d}t_i| < M$ for all i, then it follows that

$$|\mathrm{d}x| < \left(\left|\frac{\partial x}{\partial t_1}\right| + \left|\frac{\partial x}{\partial t_2}\right| + \left|\frac{\partial x}{\partial t_3}\right| + \left|\frac{\partial x}{\partial t_4}\right|\right) M.$$

In a typical real-life scenario, this yields $|\mathrm{d}x| < (3 \cdot 10^9)\,M$. Thus, if the values of all t_i are accurate to within 10^{-9} s (a nanosecond), then the estimated value of x will be within 3 m of its correct value. If the accuracy of the t_i is only $3 \cdot 10^{-8}$ s, then a typical measurement of x will be within 90 m of the actual value.

The estimate of $|\mathrm{d}x|$ just discussed requires that the values of the partial derivatives $\partial x/\partial t_i$ at the solution point are known. To determine these values, first solve the system (7) using the given satellite data. Next, take the partial derivative with respect to t_1, say, on both sides of every equation in (7), treating x, y, z, and b as functions of t_1 and treating the x_i, y_i, and z_i as constants. This yields a system of four linear equations in the unknowns $\partial x/\partial t_1$, $\partial y/\partial t_1$, $\partial z/\partial t_1$, and $\partial b/\partial t_1$. Similar systems obtained by differentiating (7) with respect to t_2, t_3, and t_4 lead to the following matrix equation.

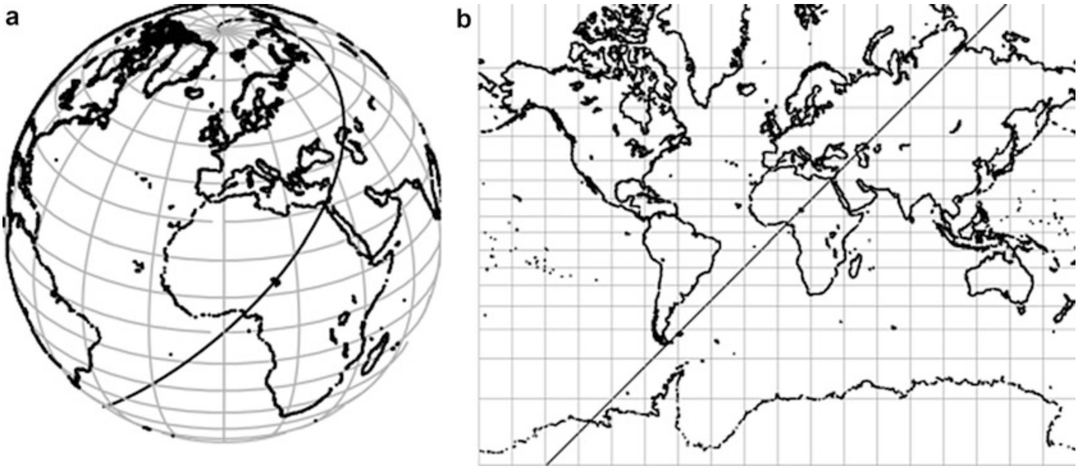
$$\begin{bmatrix} 2(x-x_1) & 2(y-y_1) & 2(z-z_1) & -2c^2(t_1+b) \\ 2(x-x_2) & 2(y-y_2) & 2(z-z_2) & -2c^2(t_2+b) \\ 2(x-x_3) & 2(y-y_3) & 2(z-z_3) & -2c^2(t_3+b) \\ 2(x-x_4) & 2(y-y_4) & 2(z-z_4) & -2c^2(t_4+b) \end{bmatrix} \begin{bmatrix} \partial x/\partial t_1 & \partial x/\partial t_2 & \partial x/\partial t_3 & \partial x/\partial t_4 \\ \partial y/\partial t_1 & \partial y/\partial t_2 & \partial y/\partial t_3 & \partial y/\partial t_4 \\ \partial z/\partial t_1 & \partial z/\partial t_2 & \partial z/\partial t_3 & \partial z/\partial t_4 \\ \partial b/\partial t_1 & \partial b/\partial t_2 & \partial b/\partial t_3 & \partial b/\partial t_4 \end{bmatrix}$$

$$= \begin{bmatrix} 2c^2(t_1+b) & 0 & 0 & 0 \\ 0 & 2c^2(t_2+b) & 0 & 0 \\ 0 & 0 & 2c^2(t_3+b) & 0 \\ 0 & 0 & 0 & 2c^2(t_4+b) \end{bmatrix} . \tag{9}$$

Once the coefficient matrix on the left-hand side and the matrix on the right-hand side of (9) have been evaluated at the solution point, then the values of all of the partial derivatives can be determined by multiplying both sides of (9) by the inverse of the coefficient matrix. Knowing these values, as well as the maximum error M in the accuracy of the t_i, one can then estimate the differentials $|\mathrm{d}x|$, $|\mathrm{d}y|$, and $|\mathrm{d}z|$ (and $|\mathrm{d}b|$, for that matter), and thereby estimate the accuracy of the location calculated by the receiver.

Map Projections

Ptolemy's goal, in *Geography*, was not only to collect the locations of as many places as possible, but to present them in the larger context of a portrait of the earth – a map. Maps go beyond mere location to reveal the many relationships that exist between different peoples and their environments. A host of map projections – specific formats for representing the earth or various parts



Mathematical Foundations of GIS, Fig. 6 A northeast/southwest loxodrome; Mercator's map for latitudes $-4\pi/9 < q\phi < q4\pi/9$

of it – have been created over the millenia. While some projections are essentially artistic, many are designed on mathematical foundations with certain uses in mind. Of particular interest in GIS are maps designed for navigational purposes and those that are amenable to the display of statistical information.

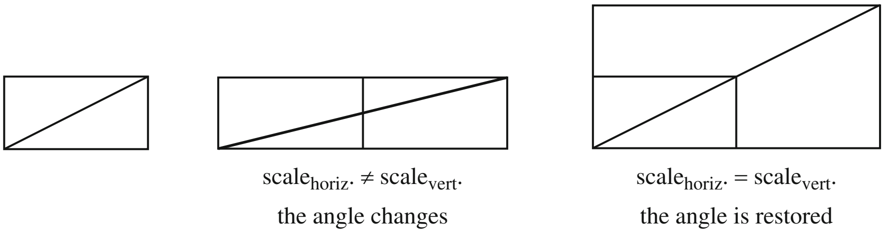
Navigation and Mercator's Map

The gnomonic projection, discussed above, preserves shortest routes, and, thus, enables navigators to plot shortest routes quite easily, provided the points are not too far apart. However, to follow a great circle path generally requires continual changes in compass bearing, which is inconvenient. A more practical approach might be to plot a route that approximates the shortest one but requires only periodic changes in compass bearings. Hence, a map on which paths of constant compass bearing on the sphere were shown as straight lines would be a useful navigational tool. It was just such a map that the Flemish geographer Gerhard Kremer, better known as Mercator, presented in 1569 with the title *Nova et aucta orbis terrae descriptio ad usum navigantium emendate accommodata* (A new and enlarged description of the earth with corrections for use in navigation).

When following a path along the surface of the Earth, one's compass bearing at any given point is represented by the angle between the direction of the path and the meridian through that particular point. Thus, a path of constant compass bearing, called a *loxodrome*, makes the same angle with every meridian it crosses. A loxodrome generally appears as a spiral converging to one of the poles, as illustrated in Fig. 6. Mercator's problem was to figure out how to show all such spirals as straight lines on a map.

To solve this problem, consider first that all parallels and all meridians have constant compass bearings and, so, must be shown as straight lines on the map. Moreover, because the east-west direction is perpendicular to the north-south direction, the images of the parallels should be perpendicular to the images of the meridians. Thus, Mercator chose for the form of his map a rectangular grid in which all parallels of latitude are shown as horizontal lines and the meridians are equally spaced vertical lines. For simplicity, place the equator along the x-axis of a two-dimensional Cartesian coordinate system and the meridian at longitude θ along the vertical line $x = \theta$, for $-\pi < q\theta < q\pi$. Thus, the overall width of the map will be 2π. The parallel at latitude ϕ will be shown as a horizontal line segment at height $y = h(\phi)$, where the function h is to be determined.

A loxodrome on the globe that makes an angle of α with every meridian it crosses should be



scale$_{\text{horiz.}}$ ≠ scale$_{\text{vert.}}$
the angle changes

scale$_{\text{horiz.}}$ = scale$_{\text{vert.}}$
the angle is restored

Mathematical Foundations of GIS, Fig. 7 How scale factors affect angles

shown on the map as a straight line that makes an angle of α with every vertical line it crosses. As Fig. 7 illustrates, this goal will be achieved if, at each point on the map, the scale factor along the parallel, represented by the horizontal edge in the figure, is equal to the scale factor along the meridian, represented by the vertical edge in the figure.

On a reference globe of radius 1 unit, the parallel at latitude ϕ has a circumference of $2\pi\cos(\phi)$, while its image on Mercator's map has length 2π. Since the meridians are evenly spaced, every section of the parallel is stretched by the same amount. Hence, the scale factor of the map along the parallel at latitude ϕ is $M_p(\phi) = \sec(\phi)$.

On the same reference globe of radius 1, the arc of any meridian lying between latitudes ϕ and $(\phi + t)$ has length t while its image on the map has length $h(\phi + t) - h(\phi)$, the gap between the horizontal lines corresponding to the two parallels. Thus, the ratio between the map measurement and the globe measurement is $(h(\phi + t) - h(\phi))/t$. To obtain the exact value of the scale factor, let t approach 0. That is, the scale factor along any meridian at a point at latitude ϕ is given by

$$M_m(\phi) = \lim_{t\to 0} \frac{h(\phi + t) - h(\phi)}{t} = h'(\phi),$$

the derivative of the height function for the parallels.

The solution to Mercator's problem, therefore, is to choose the height function $h(\phi)$ so that $M_m(\phi) = M_p(\phi)$. That is, $h'(\phi) = \sec(\phi)$. Also, $h(0) = 0$ since the equator lies on the x-axis. Together, these conditions imply that

$$h(\phi) = \int_0^{\phi} \sec(t)\, \mathrm{d}t = \ln|\sec(\phi) + \tan(\phi)|. \tag{10}$$

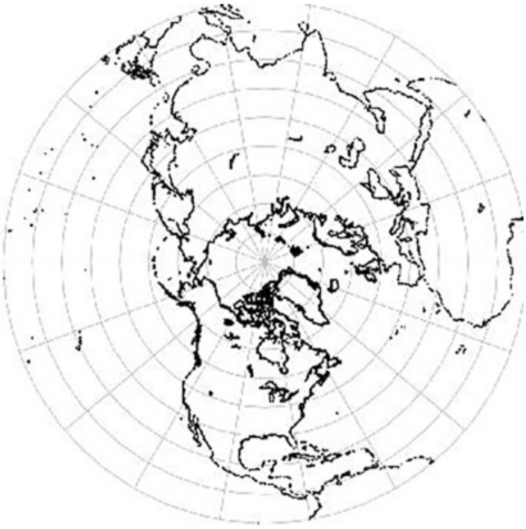
Notice that, as the latitude gets close to $\pm\pi/2$, the scale factor $\sec(\phi)$ tends to infinity. This explains why Mercator's map shows regions in the northern latitudes to be so large compared to equatorial areas.

Equipped with both a gnomonic map and Mercator's map, a navigator can plot a useful route as follows. On the gnomonic map, draw the straight line connecting the starting and ending points. This represents the shortest possible route between the two points. Next, mark some convenient reference points along this route and locate these same reference points on the Mercator map. Now, on the Mercator map, connect the reference points with straight line segments. This is the actual travel route. It follows a constant compass bearing from one reference point to the next while staying reasonably close to the shortest route.

In general, a map projection with the property that the projected images of any two intersecting paths intersect at an angle equal to that between the two paths themselves is said to be *conformal*. Loosely, a conformal map is said to "preserve angles". Thus, Mercator's map is conformal.

An Equidistant Projection

The gnomonic projection is one of the classical projections handed down from ancient Greek geometry. Two others, the stereographic and orthographic projections, are constructed in a similar fashion with the projecting light source located, respectively, at the antipode of the point of tangency of the sphere and the paper and at infinity. A fourth classical projection, the azimuthal equidistant projection, is a strictly mathematical construction that does not utilize a light source. The term *azimuthal* is used to describe any projection that has a central point rather than a central line or lines. All azimuthal maps show great circle paths through the central point as straight lines. Appropriate spacing can also ensure that the distances along these great circle paths are shown in their correct proportions. The resulting map, commonly used for atlas maps of the polar regions, is called an *azimuthal equidistant projection.* For example, on an azimuthal equidistant map with the north pole as its central point, the meridians, being great circle arcs through the pole, are shown on the map as straight lines radiating out from the center. If the prime meridian is mapped onto the positive x-axis, then the image of the meridian at longitude θ, where $-\pi < q\theta < q\pi$, will make an angle of θ with the positive x-axis. As for the parallels, note that all points at latitude ϕ, where $-\pi/2 < q\phi < q\pi/2$, are at a distance of $R(\pi/2 - \phi)$ from the north pole, where R is the radius of the reference globe. Since the map preserves distances to the pole, the image of this parallel on the map will be a circle with radius equal to $R(\pi/2 - \phi)$ centered at the pole. In particular, because the length of one degree of latitude is the same on the entire globe, the images of the parallels will be evenly spaced concentric circles on the map. The base grid for this map can be constructed using only a compass and straight-edge. Figure 8 shows the northern hemisphere.



Mathematical Foundations of GIS, Fig. 8 Azimuthal equidistant projection of the northern hemisphere

Equal-Area Maps

For the display of statistical information or other data-oriented applications, it is preferable to use a base map that shows the areas of all regions of the earth's surface in their correct proportions. Such a map is called an *equal-area* or *equivalent* map by cartographers.

The total surface area of a spherical globe of radius R is $4\pi R^2$. More generally, the area of the portion of the sphere that lies between the equator and the parallel at ϕ radians is equal to $\left|2\pi R^2 \sin(\phi)\right|$. To see this, suppose that $\phi > 0$ and use the equation $z = \sqrt{R^2 - x^2 - y^2}$ for the northern hemisphere. The surface area element is

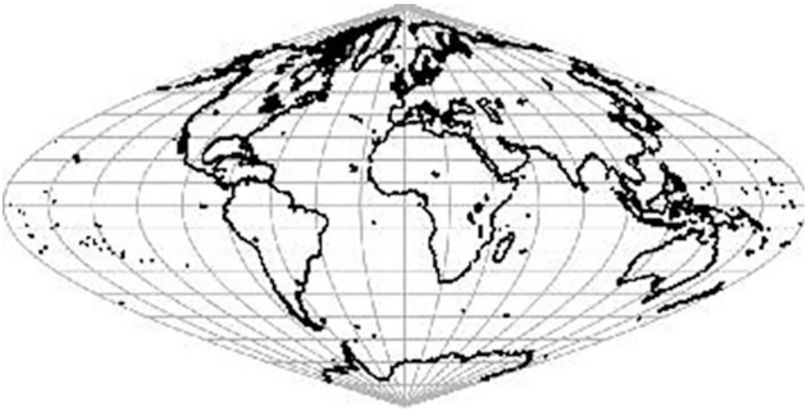
$$\begin{aligned} \mathrm{d}A &= \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1}\ \mathrm{d}x\,\mathrm{d}y \\ &= \frac{R}{\sqrt{R^2 - x^2 - y^2}}\ \mathrm{d}x\,\mathrm{d}y. \end{aligned}$$

Now integrate the surface area element over the region of integration consisting of a ring with inner radius $r = R\cos(\phi)$ and outer radius $r = R$. Switching to cylindrical coordinates yields

$$\begin{aligned} \text{area} &= \int_{\theta=0}^{2\pi} \int_{r=R\cos(\phi)}^{R} \frac{R \cdot r}{\sqrt{R^2 - r^2}}\,\mathrm{d}r\,\mathrm{d}\theta \\ &= 2\pi R^2 \sin(\phi)\,. \end{aligned}$$

Consequently, the area of the strip bounded by two parallels, with latitudes $\phi_1 < \phi_2$, is equal to $2\pi R^2[\sin(\phi_2) - \sin(\phi_1)]$. The portion of this strip that lies between the meridians with longitudes $\theta_1 < \theta_2$ has area

$$A_{\text{block}} = R^2\,(\theta_2 - \theta_1)\,[\sin(\phi_2) - \sin(\phi_1)]\,, \quad (11)$$



Mathematical Foundations of GIS, Fig. 9 Sinusoidal projection of the world

Since every region on the surface of the globe can be filled up, in the limit anyway, by some collection of (possibly infinitesimally small) blocks, each bounded by two parallels and two meridians, it follows that a given map projection is equal-area provided that the area of every such block is shown in its correct proportion. In fact, adding up the areas of many small blocks that fill up a larger region is exactly what an integral represents in calculus. One often-used equal-area projection is the sinusoidal projection, which dates back at least to 1570 when it appeared in the work of Jean Cossin of Dieppe. It was used in some later editions of Mercator's atlases (1606–1609) and also by Nicolas Sanson d'Abbeville beginning in 1650 and by John Flamsteed (1646–1719), the first astronomer royal of England. In addition to its current appellation, the sinusoidal projection has been known variously as the Sanson-Flamsteed, the Mercator-Sanson, or the Mercator equal-area projection (Fig. 9).

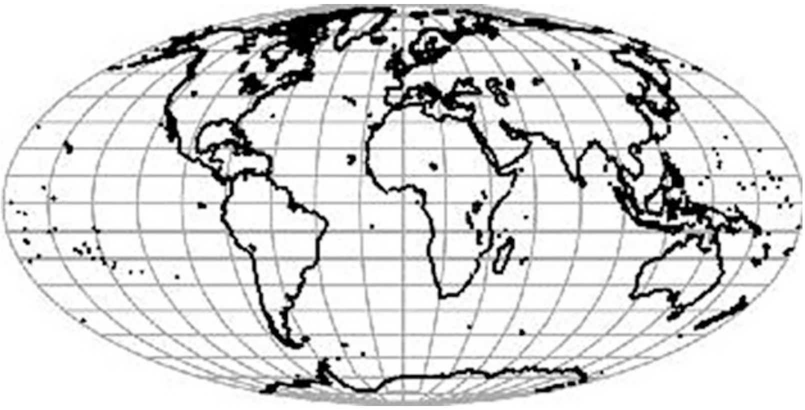
On the sinusoidal map, the meridian at longitude θ is depicted as the graph of the curve $x = \theta \cos(y)$, for $-\pi/2 < y < \pi/2$, while the parallel at latitude ϕ is represented as a segment of the horizontal line $y = \phi$ extending between the map's boundary curves $x = -\pi \cos(y)$ and $x = \pi \cos(y)$. Thus, the lengths of the parallels are portrayed in their correct proportions. Also, the parallels are evenly spaced along the y-axis, just as they are along a meridian on the globe, and the meridians are spaced evenly along the parallels. For any block on the globe described by the conditions $\phi_1 < q\phi < q\phi_2$ and $\theta_1 < q\theta < q\theta_2$, the image of this block on the map has area

$$\int_{y=\phi_1}^{\phi_2} (\theta_2 - \theta_1)\cos(y)\,\mathrm{d}y$$
$$= (\theta_2 - \theta_1)\,[\sin(\phi_2) - \sin(\phi_1)]\,,$$

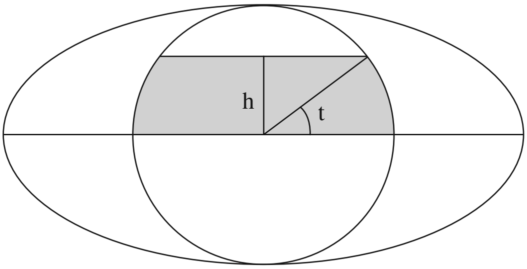
in agreement with Eq. (11).

Another popular equal-area map is the Mollweide map, presented in 1805 by Karl Brandan Mollweide. This projection portrays the whole world in an ellipse whose axes are in a 2:1 ratio. The facing hemisphere is depicted as a central circle, with the diameter of the circle equal to the vertical axis of the overall ellipse. The "dark side" of the earth is split in two with one piece shown on either side of the central circle. To complete the overall structure of the map, the parallels are drawn as horizontal lines on the map, while the two meridians at θ and $-\theta$ will together form an ellipse whose vertical axis coincides with the vertical axis of the overall ellipse. The meridians will be equally spaced along the equator (Fig. 10).

The mathematics involved in the construction of this projection is rather more complicated than that of the sinusoidal. The central circle has radius $\sqrt{2}$ and area 2π, the area of a hemisphere on a reference globe of radius 1. If the parallel at latitude ϕ is placed on the line $y = h(\phi)$, then, to ensure that areas are preserved, the shaded region in Fig. 11 must have area $\pi \sin(\phi)$. Thus, the angle t, shown in the figure, must satisfy $\pi \sin(\phi) = 2t + \sin(2t)$. This equation can be solved only numerically. Then $h(\phi) = \sqrt{2}\sin(t)$.



Mathematical Foundations of GIS, Fig. 10 The Mollweide equal-area projection



Mathematical Foundations of GIS, Fig. 11 For each latitude ϕ, choose t so that the shaded area, $2t + \sin(2t)$, is equal to $\pi \sin(\phi)$

The meridians at $\pm\theta$ form a lune with area 4θ on a globe of radius 1 unit. The ellipse formed by these meridians on the map has a vertical axis of length $2\sqrt{2}$, and, hence, has the equation $\pi^2 x^2 + 4\theta^2 y^2 = 8\theta^2$.

In 1925, J. P. Goode introduced his Homolosine map, an interrupted projection devised by fusing together parts of four sinusoidal maps and seven Mollweide maps, with various central meridians. The Mollweide maps are used in the upper latitudes to smooth out the polar regions.

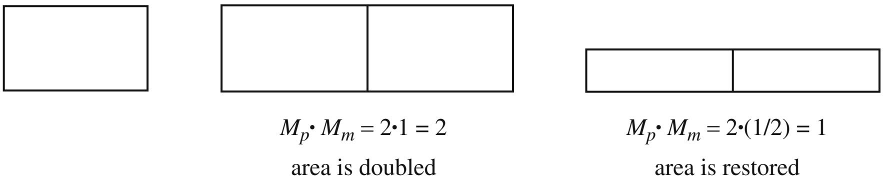
Lambert's azimuthal equal-area projection, presented by Johann Heinrich Lambert in 1772, is widely used for atlas maps today. For a map centered on the north pole, the image of the pole will be taken as the origin in the plane of the map. The parallels will be shown as concentric circles centered at the origin, with the circle corresponding to latitude ϕ having a radius of $r(\phi)$. The function $r(\phi)$, which will be decreasing, is to be determined. The meridian at longitude θ will be portrayed as a radial line segment emanating from the origin and making an angle of θ with the positive x-axis.

While the parallel at ϕ has circumference $2\pi\cos(\phi)$ on a reference globe of radius 1, its image has circumference $2\pi r(\phi)$. Thus, the scale factor of the map along this parallel is $M_p(\phi) = r(\phi)\sec(\phi)$. The arc of any meridian lying between latitudes ϕ and $(\phi + t)$ has length t while its image on the map has length $r(\phi) - r(\phi + t)$, the gap between the circles corresponding to the two parallels. Thus, the ratio between the map measurement and the globe measurement is $(r(\phi) - r(\phi + t))/t$. Let t approach 0 to obtain the scale factor along any meridian at a point at latitude ϕ:

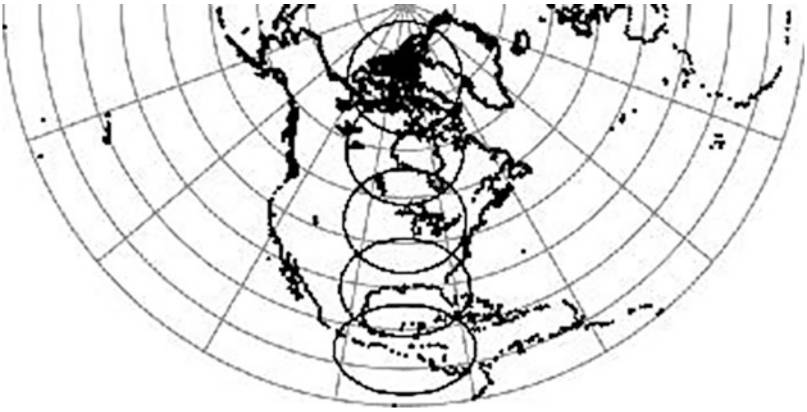
$$M_m(\phi) = \lim_{t \to 0} \frac{r(\phi) - r(\phi + t)}{t} = -r'(\phi).$$

For this projection to preserve areas, the condition $M_p \cdot M_m = 1$ must be met. (See Fig. 12.) Hence, the function $r(\phi)$ must satisfy $-\sec(\phi)r(\phi)r'(\phi) = 1$. This equation can be rewritten as $r\,\mathrm{d}r = -\cos(\phi)\,\mathrm{d}\phi$. Integrating both sides yields the equation $r^2/2 = -\sin(\phi) + C$. Since $r = 0$ when $\phi = \pi/2$, it follows that $C = 1$. Moreover, $r \geq 0$, so that $r = \sqrt{2} \cdot \sqrt{1 - \sin(\phi)} = 2\sin(\pi/4 - \phi/2)$. This formula is uniquely determined by the conditions that led to it. Hence, this projection, of which an example is shown in Fig. 13, is the *only* azimuthal equal-area projection up to overall scale change.

The *conic projections* form an important class of projections that is not discussed here in detail.


$M_p \cdot M_m = 2 \cdot 1 = 2$
area is doubled
$M_p \cdot M_m = 2 \cdot (1/2) = 1$
area is restored

Mathematical Foundations of GIS, Fig. 12 Areas are affected by scale changes



Mathematical Foundations of GIS, Fig. 13 Tissot's indicatrix for Lambert's equal-area azimuthal projection. All *ellipses* have the same area, but are more elongated away from the pole

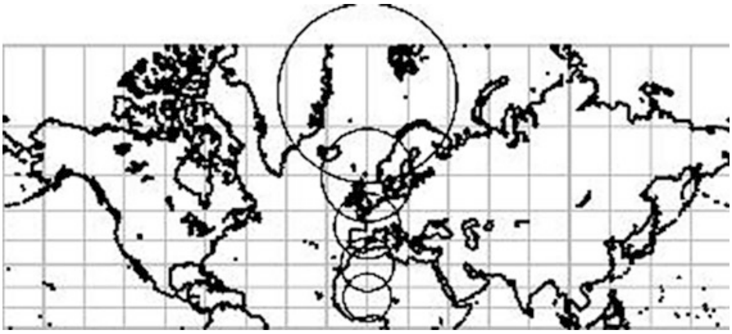
In these, a cone is placed on the globe and the globe somehow projected onto the cone, which is then slit open and laid out to form a sector of a circle. Conic projections were used by Ptolemy *circa* 150 A.D. and are especially useful for mapping portions of the globe that are wide east-to-west but short north-to-south, such as Russia or The United States. The United States Geological Survey uses conic projections extensively for its topographical maps. Albers' equal-area conic projection, presented in 1805, and Lambert's conformal conic, introduced in 1772, are both prevalent today. Both cylindrical and conic projections can be modified to have two standard lines where the sphere and the cylinder or cone meet. The analysis of scale factors for a conic projection is similar to that for an azimuthal projection, with the slight complication that the image fills only a sector of a circle. The angle of the sector depends on the choice of the standard lines for the map.

Map Distortion

Just as the analysis of scale factors was the key to constructing maps that preserved angles or areas, so it is central to understanding the extent to which a given map distorts those measurements. For instance, for a cylindrical, azimuthal, or conic projection, the condition $M_p = M_m$ ensures conformality, while the equation $M_p \cdot M_m = 1$ characterizes an equal-area map. More generally, for these classes of projections, the values of M_p/M_m and $M_p{\cdot}M_m$ can be used as measures of the distortions in angles and areas, respectively. The more these differ from the value 1, the greater are the map's distortions. For example, Mercator's map satisfies $M_p/M_m = 1$ at every point, reflecting the map's conformality. But $M_p \cdot M_m = \sec^2(\phi)$ for this map, indicating a severe distortion of ares as φ approaches $\pm\pi/2$. For Lambert's azimuthal equal-area projection, $M_p \cdot M_m = 1$, but $M_p/M_m = \sec^2(\pi/4 - \phi/2)$. So this map distorts angles increasingly away from the north pole.

Tissot's Indicatrix

In the late nineteenth century, a French mathematician, Tissot, developed a method, called *Tissot's indicatrix*, that has become a standard cartographic tool for quantifying and, especially,



Mathematical Foundations of GIS, Fig. 14 Tissot's indicatrix for Mercator's projection. All *ellipses* are *circles*, but the areas increase away from the equator

visualizing distortions in angles and areas. The starting point for this technique is Tissot's observation that, for any map projection, there is at each point on the sphere a pair of perpendicular directions whose images in the projection are also perpendicular. Tissot called these the *principal directions* at the given point. Schematically, at each point on the map, Tissot constructed an ellipse whose principal axes were aligned with the principal directions and had lengths equal to the scale factors of the map projection in those directions. In practice, a representative selection of points is used and the ellipses are rescaled by a common factor so that they fit on the map and don't interfere with each other too much. In this way, one can make effective visual comparisons between different ellipses in the indicatrix.

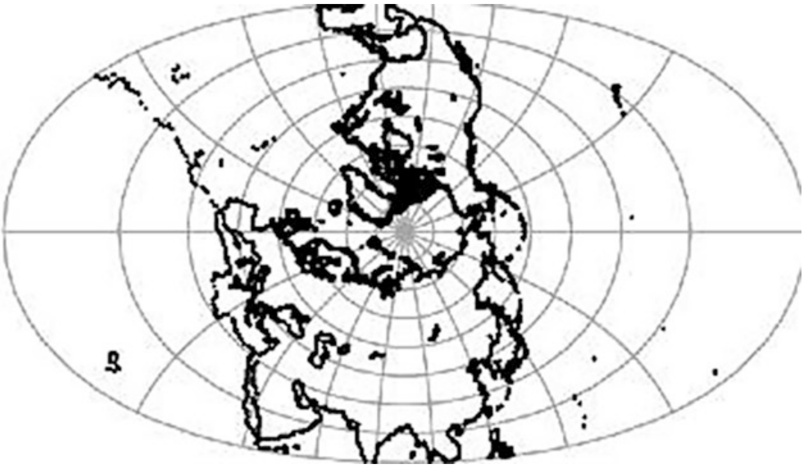
For cylindrical, azimuthal, and conic projections with standard perspective, the principal directions at any point lie along the meridian and the parallel. The corresponding scale factors are simply $M_m(\phi)$ and $M_p(\phi)$. Thus, Tissot's indicatrix consists of a system of ellipses whose principal axes have lengths M_m and M_p. The area of such an ellipse is proportional to $M_p \cdot M_m$. Hence, if $M_p \cdot M_m$ is constant, then the ellipses have the same area over the whole map and the projection is equal-area. In general, the more $M_p \cdot M_m$ varies, the more the map distorts areas. Similarly, the ratio M_p/M_m represents the ratio of the lengths of the principal axes of the ellipses in the indicatrix. If this ratio is equal to 1 at every point, then every ellipse is actually a circle and the projection is conformal. The more this ratio varies from the value 1, the more elongated is the corresponding ellipse and, thus, the more distorted are the angles at that point. Tissot also used the principal scale factors to measure, at each point on the projection, the maximum error between any angle and its image angle (Fig. 14).

Oblique Perspectives

Most maps, in their standard presentations, have either an equatorial or a polar perspective. In applications, however, a different perspective may be preferable. Conceptually, an arbitrary point A can be viewed as taking the place of the north pole and an imaginary set of "meridians" – great circles emanating from A and passing through the point antipodal to A – can be formed. Likewise, an imaginary "equator" lies halfway between A and its antipode, and "parallels" subdivide the imaginary meridians. In this new framework, every point on the globe can be assigned a new pair of relative longitude and latitude coordinates. Mathematically, the usual map equations are applied to these relative longitude and latitude values. The relative coordinates themselves can be computed using linear algebra.

The Transverse Mercator Map

One of the most important maps that Lambert presented in 1772 is the transverse Mercator projection, a version of Mercator's map that is centered at the north pole instead of at the equator. To construct a transverse Mercator projection of the northern latitudes, the north pole, with Cartesian coordinates $N\langle 0, 0, 1\rangle$, will take the place of the point $\langle 1, 0, 0\rangle$ as the center of the map. The point $B\langle -1, 0, 0\rangle$, where the meridian at $\pm\,\pi$ meets the equator, will rotate into the place



Mathematical Foundations of GIS, Fig. 15 Transverse Mercator projection of the northern hemisphere, latitude $\phi \geq \pi/18$

originally occupied by the north pole, $\langle 0, 0, 1\rangle$. Finally, the point $C\langle 0, 1, 0\rangle$ will be fixed by this rotation. The vectors N, B, and C form an orthonormal system. The matrix that converts standard Cartesian coordinates into coordinates relative to the new system is

$$T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

The point at (θ, ϕ) has Cartesian coordinates $\langle \cos(\theta)\cos(\phi), \sin(\theta)\cos(\phi), \sin(\phi)\rangle$. Apply T to this to get the relative Cartesian coordinates $\langle \sin(\phi), \sin(\theta)\cos(\phi), \cos(\theta)\cos(\phi)\rangle$.

For the standard Mercator map, the x-coordinate is equal to the longitude of the point being mapped. For the transverse Mercator, use the relative longitude instead: $\tilde{\theta} = \arctan\left(\frac{\sin(\theta)\cos(\phi)}{\sin(\phi)}\right)$. The y-coordinate on the map is $\ln\left|\sec(\tilde{\phi}) + \tan(\tilde{\phi})\right|$, where $\tilde{\phi} = \arcsin(-\cos(\theta)\cos(\phi))$ is the relative latitude. Figure 15 shows a transverse Mercator projection of the northern hemisphere.

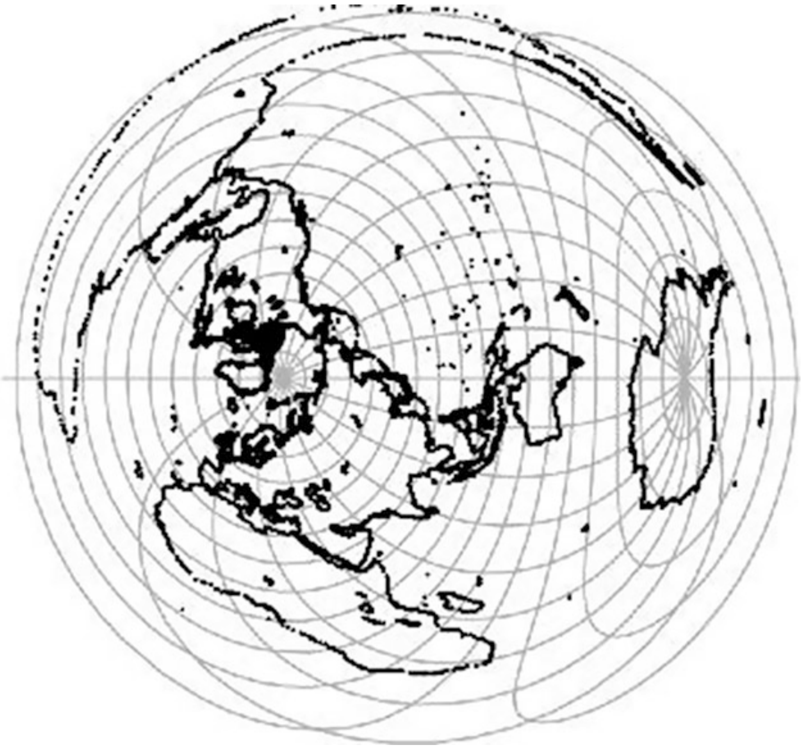
The transverse Mercator map is still conformal, though loxodromes are no longer seen as straight lines. (A path that makes a constant angle with the *relative* meridians emanating from the point B, however, is a straight line!)

In general, a map having an oblique, i.e., nonstandard, perspective can be obtained by imagining the desired centering point A either as a pole or as the point where the equator and prime meridian meet. Let B be the point obtained either by adding or by subtracting $\pi/2$ from the latitude of A. To complete the orthonormal system, use either $C = A \times B$ or $C = -A \times B$, where '×' denotes the vector cross product. (The exact choices of B and C depend on the hemisphere involved and on whether the projection is equatorial or polar in its standard perspective.) Now form the transformation matrix T whose rows are given by the Cartesian coordinates of A, B, and C arranged according to the order in which they take the places of $\langle 1, 0, 0\rangle$, $\langle 0, 1, 0\rangle$, and $\langle 0, 0, 1\rangle$, respectively. The matrix T will transform standard Cartesian coordinates into relative Cartesian coordinates. The standard map equations can then be applied to the corresponding relative values of longitude and latitude.

For example, using A = Tokyo, with Cartesian coordinates $\langle -0.62, 0.52, 0.59\rangle$, in place of the north pole, let $B = \langle -0.45, 0.38, -0.81\rangle$ and $C = A \times B = \langle -0.64, -0.77, 0\rangle$. The transformation matrix is

$$T = \begin{bmatrix} -0.45 & 0.38 & -0.81 \\ -0.64 & -0.77 & 0 \\ -0.62 & 0.52 & 0.59 \end{bmatrix}.$$

For the point with longitude θ and latitude ϕ, the relative Cartesian coordinates are



Mathematical Foundations of GIS, Fig. 16 Azimuthal equidistant projection centered at Tokyo

$$T\begin{bmatrix}\cos(\theta)\cos(\phi)\\ \sin(\theta)\cos(\phi)\\ \sin(\phi)\end{bmatrix} = \begin{bmatrix}\begin{array}{c}-0.45\cos(\theta)\cos(\phi)\\ +0.38\sin(\theta)\cos(\phi)-0.81\sin(\phi)\end{array}\\ -0.64\cos(\theta)\cos(\phi)-0.77\sin(\theta)\cos(\phi)\\ \begin{array}{c}-0.62\cos(\theta)\cos(\phi)\\ +0.52\sin(\theta)\cos(\phi)+0.59\sin(\phi)\end{array}\end{bmatrix}.$$

From these, the relative longitude and latitude of any point can be determined. Applying the standard formulas for an azimuthal equidistant projection to the relative coordinates yields the map in Fig. 16.

Future Directions

Technological advances continue to yield improvements in the accuracy of GPS and other geodetic data. Though literally hundreds of map projections have been developed over the centuries, as long as people continue to find new information to present using maps there will be new ideas for projections on which to present it.

Cross-References

- ▶ Generalization and Symbolization
- ▶ Map Generalization
- ▶ Privacy Preservation of GPS Traces
- ▶ Scale, Effects
- ▶ University of Minnesota (UMN) Map Server
- ▶ Wayfinding, Landmarks
- ▶ Web Mapping and Web Cartography

Recommended Reading

Banerjee S (2004) Revisiting spherical trigonometry with orthoginal projectors. Coll Math J 35(5):375–381

Berggren JL, Jones A (2000) Ptolemy's geography. Princeton University Press, Princeton

Bosowski EF, Feeman TG (1997) The use of scale factors in map analysis: an elementary approach. Cartographica 34(4):35–44

Bugayevskiy LM, Snyder JP (1995) Map projections: a reference manual. Taylor and Francis, London
Cotter CH (1966) The astronomical and mathematical foundations of geography. American Elsevier, New York
Dent BD (1996) Cartography, 4th edn. Wm. C. Brown, Dubuque
Espenshade EB Jr et al (eds) (1995) Goode's world atlas, 19th edn. Rand McNally, Skokie
Feeman TG (2002) Portraits of the earth: a mathematician looks at maps. American Mathematical Society, Providence
Forman S, Steen L (2006) Global positioning system. http://www.stolaf.edu/other/ate/gps.html. Accessed 8 May 2006
Nord G, Jabon D, Nord J (1998) The global positioning system and the implicit function theorem. SIAM Rev 40(3):692–696
Snyder JP (1993) Flattening the earth: two thousand years of map projections. University of Chicago Press, Chicago
Thompson RB (1998) Global positioning system: the mathematics of GPS receivers. Math Mag 71(4): 260–269
Wikipedia (2006) Global positioning system. http://en.wikipedia.org. Accessed 8 May 2006
Wikipedia (2006) Trilateration. http://en.wikipedia.org. Accessed 8 May 2006

Mathematical Programming

▸ Multicriteria Decision-Making, Spatial

Mathematical Theory of Geosensor Networks

▸ Geosensor Networks, Formal Foundations

Matrices, Geographic

▸ Temporal GIS and Applications

Matrix, Inverse

▸ Hurricane Wind Fields, Multivariate Modeling

MAUP

▸ Error Propagation in Spatial Prediction

MaxCount Spatiotemporal Aggregate Operator

Scot Anderson
Southern Adventist University, Collegedale, TN, USA

Synonyms

Range aggregate operators; Spatiotemporal aggregation

Definition

Spatiotemporal aggregation operators answer summary questions about spatiotemporal data maintained in spatiotemporal databases. Aggregation over the two distinctly different domains of time and space requires unique index structures and algorithms. MaxCount finds the largest number of objects intersecting a dynamic query space and the time at which this maximum occurs. Formally we define MaxCount as follows: Let S be a set of moving objects. Given a dynamic query space R defined by two moving points Q_1 and Q_2 as the lower-left and upper-right corners of R, and a time interval T, the MaxCount operator finds the time $t_{\max}$ and maximum number of points $M_{\max}$ in S that R can contain at any time instance within T. Dynamic query space is defined by a continuous time interval T, and a d-dimensional space that can move, and change size or shape over the query time interval (Anderson and Revesz 2009). In the index a dual transform (Kollios et al. 2005), represents the linear motion as a point consisting of a starting position and velocity in each dimension of motion. In *three*-dimensional space, the query space would then occupy *six* dimensions. Changes to any one of the six parameters to a moving object in *three*-dimensional space represent an update to the object and cause an update to the index structure. Buckets contain density functions based on dual transforms and allow extremely efficient approximate calculations of MaxCount based on complex integrals over the query space.

MaxCount and Other Spatiotemporal Aggregation Operators

Background

Research into MaxCount originated in the exploration of aggregation in spatiotemporal domains by Revesz and Chen (2003) and Chen and Revesz (2004) as a natural extension to spatiotemporal indexing. Spatiotemporal aggregation running times led to the examination of estimation methods first by expanding the previous work into *two*-dimensional space (Anderson 2006) and then expanding to d-dimensional space (Anderson and Revesz 2009). Interest in spatiotemporal indexing and aggregation drove the original work to examine safety for air travel and for large numbers of moving objects within a space. This research led to a family of spatiotemporal, threshold, aggregate operators including MaxCount (MinCount), ThresholdRange, ThresholdCount, ThreshouldSum, ThresholdAverage, and CountRange (Anderson 2007). A new spatiotemporal indexing method that allows inserts and deletes to occur in $O(1)$ time (Anderson 2007) supports efficient running times for these operators.

Exact and Approximation Algorithms

Two variations of the MaxCount operator exist: the exact and estimate methods. The estimate algorithm runs in $O(B)$ time and space where the parameter B represents the number of buckets in the index (Anderson 2007). For the exact algorithm, time is $O(N)$ and space is $O(1)$. In comparing the exact and estimate algorithms, running times for the exact algorithm varied widely depending on the size of the query space and the number of objects intersecting it. The estimate algorithm was designed to give accuracy at or above 95% in a short predictable time. Note that B does not depend on the number of objects N. Instead the number of buckets must be chosen high enough to represent the density patterns in the d-dimensional query space. In practice, 20 buckets was enough to achieve better than 95% accuracy even with millions of moving objects (Anderson 2007). MaxCount and the family of threshold aggregate operators now represent a mature set of aggregate operators for spatiotemporal data.

Recommended Reading

Anderson S (2006) Aggregation estimation for 2D moving points. In: Thirteenth international symposium on temporal representation and reasoning, Piscataway. IEEE Computer Society Press, pp 137–144

Anderson S (2007) Software verification and spatiotemporal aggregation in constraint databases. PhD thesis, University of Nebraska, Lincoln

Anderson S, Revesz P (2009) Efficient maxcount and threshold operators of moving objects. Geoinformatica 13(4):355–396

Chen Y, Revesz P (2004) Max-count aggregation estimation for moving points. In: Proceedings of the 11th international symposium on temporal representation and reasoning, Tatihou, pp 103–108

Kollios G, Papadopoulos D, Gunopulos D, Tsotras J (2005) Indexing mobile objects using dual transformations. VLDB J 14(2):238–256

Revesz P, Chen Y (2003) Efficient aggregation over moving objects. In: Proceedings of the 10th international symposium on temporal representation and reasoning, fourth international conference on temporal logic, Cairns, pp 118–127

Max-Enclosing Rectangle Problem

▸ Maximizing Range Sum in Spatial Databases

Maximizing Range Sum in Spatial Databases

Dong-Wan Choi[1], Chin-Wang Chung[2,3], and Yufei Tao[4]
[1]Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea
[2]Chongqing Liangjiang KAIST International Program, Chongqing University of Technology, Chongqing, China
[3]School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, Korea
[4]Chinese University of Hong Kong, Hong Kong, China

Synonyms

Facility location problem; Max-enclosing rectangle problem; Optimal-location query

Definition

Let O be a set of objects (a.k.a. points) in 2D space $\mathbb{R}^2$, where $\mathbb{R}$ represents the real domain. Each object $o \in O$ is associated with a positive value $w(o)$ as its weight. Given non-negative values d_1 and d_2, the goal of the maximizing range sum (MaxRS) problem is to place a $d_1 \times d_2$ rectangle r in $\mathbb{R}^2$ to maximize the covered weight of r, defined as:

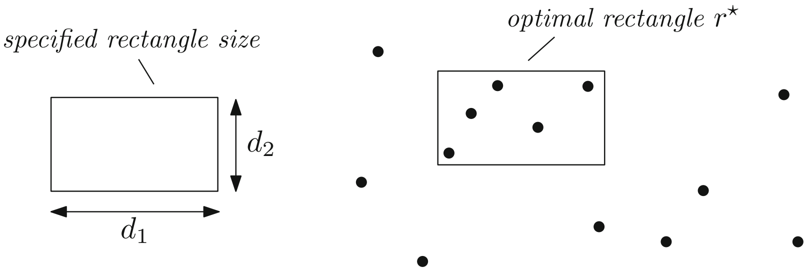
$$covered - weight\ (r) = \sum_{o \in O \cap r} w\,(p).$$

In plain words, *covered-weight*(r) equals the total weight of the objects of O that are covered by r. As a special case, if every object in O has weight 1, then *covered-weight*(r) simply indicates how many objects of O fall in r. Note that the position of r can be anywhere in the data space, namely, there are infinitely many possible rectangles that could have been chosen. To illustrate, consider that O is the set of black points in Fig. 1, and (for simplicity) that all objects have weight 1. Given the lengths of d_1 and d_2 as shown on the figure's left, an optimal solution to the MaxRS problem is the rectangle r^*, which covers 5 objects of O. One can easily verify that no rectangle of size $d_1 \times d_2$ can enclose at least 6 objects.

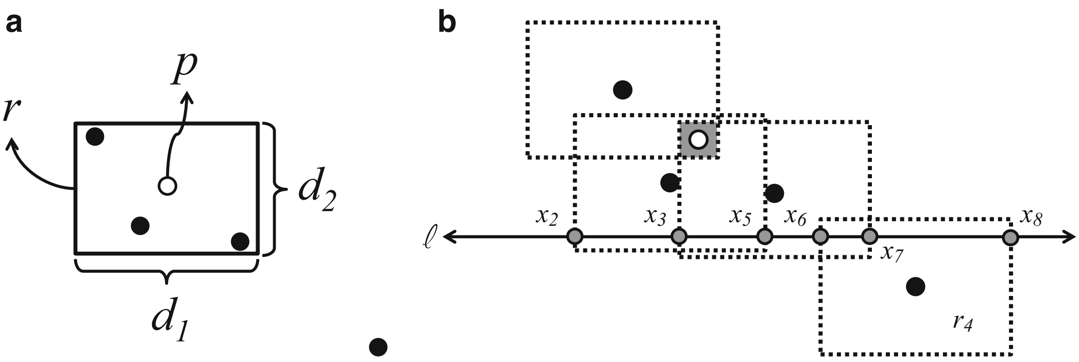
Historical Background

In the theoretical perspective, the MaxRS problem belongs to the *object placement* problems which aim to find an optimal placement of a given geometric object such that the number of points covered by the object is maximized. This kind of problems have been actively studied in computational geometry in the past. When the object is a rectangle, which is the MaxRS problem, it is known that the problem can be solved in $O(n \log n)$ time by Eppstein and Erickson (1994), Imai and Asano (1983), Nandy and Bhattacharya (1995). Chazelle and Lee (1986) presented an $O(n^2)$ algorithm for the case where the object is a circle, which is believed to be optimal. A more general case in which the object is a convex polygon was also studied by Barequet et al. (1997) and Dickerson and Scharstein (1998).

In spatial databases, some relevant problems to the MaxRS problem have been studied, which are the problems of finding the optimal location based on their own requirements. Du et al. (2005) proposed the *optimal-location query* that returns a location in a query region to maximize the total weight of the reverse nearest neighbors in L_1 distance, which is further extended by Zhang et al. (2006). Similar problems have been also studied by Wong et al. (2009) and Zhou et al. (2011), focusing on L_2 distance. Xia et al. (2005) proposed the *top-t most influential site query*, which selects top-t locations among the pre-determined candidate locations based on a given ranking function such as the number of nearest neighbors. Later, the extended version of this problem, called the *top-k spatial preference query*, was proposed in Rocha-Junior et al. (2010) and Yiu et al. (2007) by considering the set of classified objects. Xiao et al. (2011) presented



Maximizing Range Sum in Spatial Databases, Fig. 1 The MaxRS problem



Maximizing Range Sum in Spatial Databases, Fig. 2 An example of transformation. (**a**) The MaxRS problem. (**b**) Transformed MaxRS problem

the optimal-location query processing in road networks.

Scientific Fundamentals

In order to solve the MaxRS problem, the transformation strategy is generally used, which is firstly proposed in Nandy and Bhattacharya (1995). Specifically, the transformation process is as follows: for each object $o \in O$, construct a corresponding weighted rectangle r_o, which is centered at the location of o and has a weight $w(o)$. Let R denote the set of all such weighted rectangles, then the goal of the transformed MaxRS problem is to find the *densest* region with respect to R such that the total weight of rectangles covering the densest region is maximized. Consider a transformation example shown in Fig. 2. Suppose that O has four objects (black points) as shown in Fig. 2a. Given a $d_1 \times d_2$ rectangle, an optimal location can be the center point p of rectangle r. Figure 2b shows the corresponding transformed problem, where there are four weighted rectangles in R, each centering at each object in O. It is not difficult to observe that the optimal point p in the MaxRS problem can be any point in the densest region (gray-filled) with respect to R which is the outcome of the transformed MaxRS problem. Thus, once the densest region is found in the transformed MaxRS problem, the optimal location of the original MaxRS problem can trivially be obtained.

For the transformed MaxRS problem, an *in-memory* algorithm was proposed in Imai and Asano (1983), which is based on the well-known *plane-sweep* algorithm. Basically, the algorithm regards the top and bottom edges of rectangles as horizontal intervals, and maintains a binary tree on the intervals while sweeping a conceptual horizontal line from bottom to top. For example, there will be 7 intervals in the binary tree when the sweeping line is at ℓ in Fig. 2b, namely

$[-\infty, x_2]$, $[x_2, x_3]$, $[x_3, x_5]$, $[x_5, x_6]$, $[x_6, x_7]$, $[x_7, x_8]$, and $[x_8, \infty]$. When the line meets the bottom (top) edge of a rectangle, a corresponding interval is inserted to (deleted from) the binary tree, along with updating the *count*s of intervals currently residing in the tree, where the count of an interval indicates the total weight of intersecting rectangles within the interval. In Fig. 2b, when the line encounters the top edge of the rectangle r_4, the intervals in the range of $[x_6, x_8]$ will be updated or deleted from the binary tree. Thereafter, the resulting intervals in the binary tree will be $[-\infty, x_2]$, $[x_2, x_3]$, $[x_3, x_5]$, $[x_5, x_7]$, and $[x_7, \infty]$, whose counts are 0, 1, 2, 1, and 0, respectively. During the whole sweeping process, an interval with the maximum count is returned as the final result. The time complexity of this algorithm is $O(n \log n)$, where n is the number of rectangles, since n insertions and n deletions are performed during the sweep, and the cost of each tree operation is $O(\log n)$. This is the best efficiency possible in terms of the number of comparisons (Imai and Asano 1983).

As shown in Du et al. (2005), the same strategy also works in external memory, but entails $O(n \log_B n)$ I/Os (B is the size of a disk block), which unfortunately is not scalable in practice. To remedy this, Choi et al. (2012) proposed an external-memory algorithm, namely *ExactMaxRS*, that exactly solves the transformed MaxRS problem in $O((n/B) \log_{M/B} (n/B))$ I/O's, where M is the size of main memory. This is known (Arge et al. 1993; Imai and Asano 1983) to be the lower bound under the comparison model in external memory.

At a high level, the ExactMaxRS algorithm follows the divide-and-conquer strategy, where the entire datset is recursively divided into mutually disjoint subsets, and then the solutions that are locally obtained in the subsets are combined. The overall process of the ExactMaxRS algorithm is as follows:

1. Recursively divide the whole space vertically into m sub-spaces, called *slabs* and denoted as $\gamma_1, \ldots, \gamma_m$, each of which contains roughly the same number of rectangles, until the rectangles belonging to each slab can fit in the main memory.
2. Compute a solution structure for each slab, called *slab-file*, which represents the local solution to the sub-problem with regard to the slab.
3. Merge m slab-files to compute the slab-file for the union of the m slabs until the only one slab-file remains.

In this process, the following questions arise: (1) How to divide the space to guarantee the termination of recursion; (2) how to organize slab-files, and what should be included in a slab-file; (3) how to merge the slab-files without loss of any necessary information for finding the final solution.

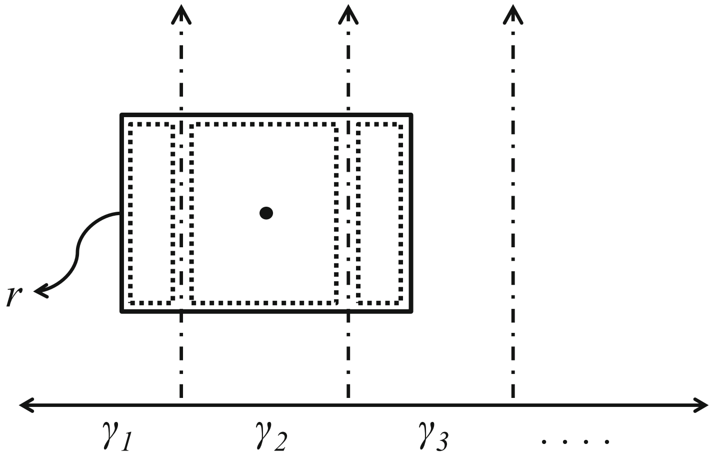
- **Division Phase** Basically, the algorithm recursively divides the space vertically into m slabs along the x-dimension, where $m = \Theta(M/B)$, until the number of rectangles in a slab can fit in the main memory. Since a rectangle in R can be large, it is unavoidable that a rectangle may need to be split into a set of smaller disjoint rectangles as the recursion progresses, which is shown in Fig. 3.

One important issue in the division phase is how to guarantee that the number of rectangles in the sub problem gets gradually smaller. For this purpose, the algorithm does not send the rectangles whose x-ranges cover the entire x-ranges of their slabs (e.g., the middle part of r in Fig. 3), which are called *spanning* rectangles, into the next level of recursion. Instead, spanning rectangles are separately stored in another file and further considered in the merging phase.

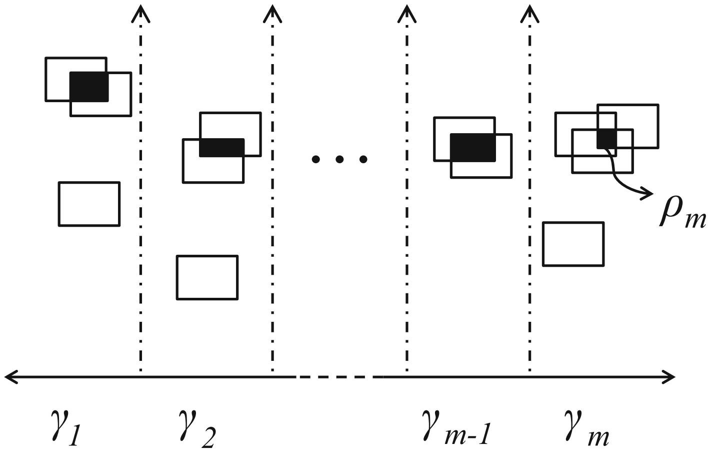
- **Slab-files** The next important question is how to organize a slab-file. What the question truly asks about is what structure should be returned after *conquering* the sub-problem with regard to a slab. Each slab-file should have enough information to find the final solution after all the merging phases.

To get the intuition first, consider an easy scenario where every rectangle has the same weight,

Maximizing Range Sum in Spatial Databases, Fig. 3 An example of splitting a rectangle



Maximizing Range Sum in Spatial Databases, Fig. 4 An easy scenario to illustrate the intuition of slab-files

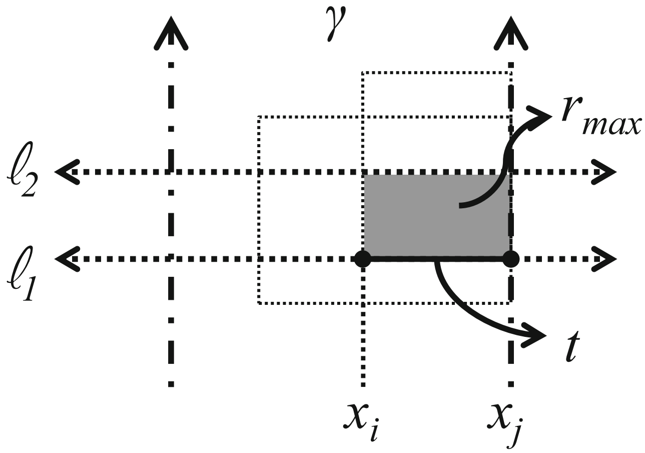


and is small enough to be totally inside a slab, which is shown in Fig. 4. Thus, no spanning rectangle exists. In this case, it is easy to see that it suffices to just maintain the densest region (black-filled in Fig. 4) with regard to rectangles in each slab. Then, in the merging phase, among m densest regions (i.e., one for each slab), choose the best one as the final solution. In Fig. 4, for instance, the best one is ρ_m because it is the intersection of 3 rectangles, whereas the number is 2 for the max regions of the other slabs.

Based on this idea, it is further observed that the horizontal boundaries of the densest region of a slab are laid on the horizontal lines extended from the bottom or top edge of a certain rectangle in the slab. The term *h-line* is used to refer to a horizontal line extended from a horizontal edge of an input rectangle. Therefore, for each h-line in a slab, it suffices to maintain a segment that could belong to the densest region of the slab. To formalize the concept of such a segment, ExactMaxRS defines the notion *max-interval* as follows:

Let (1) $\ell.y$ be the y-coordinate of a h-line ℓ, and ℓ_1 and ℓ_2 be the consecutive h-lines such that $\ell_1.y < \ell_2.y$, (2) $\ell \cap \gamma$ be the part of a h-line ℓ in a slab γ, and (3) r_γ be the rectangle formed by $\ell_1.y, \ell_2.y$, and vertical boundaries of γ. A max-interval is a segment t on $\ell_1 \cap \gamma$ such that, the x-range of t is the x-range of the rectangle $r_{\max}$ bounded by $\ell_1.y, \ell_2.y$, and vertical lines at x_i and x_j, where the total weight of rectangles covering $r_{\max}$ is maximized confined in r_γ.

Figure 5 illustrates the definition of the max-interval.



Maximizing Range Sum in Spatial Databases, Fig. 5 An illustration of the definition of the max-interval

A slab-file is the set of max-intervals defined *only on h-lines*. Each max-interval is represented as a tuple specified as follows:

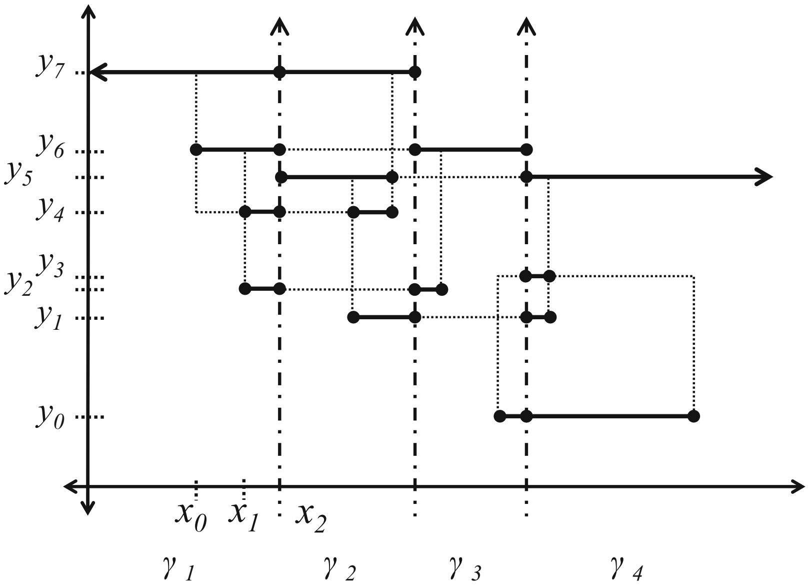
$$t = < y, [x_1, x_2], sum >$$

where y is the y-coordinate of t (hence, also of the h-line that defines it), and $[x_1, x_2]$ is the x-range of t, and *sum* is the total weight of rectangles covering the region corresponding to t. In addition, all the tuples in a slab-file should be sorted in ascending order of y-coordinates.

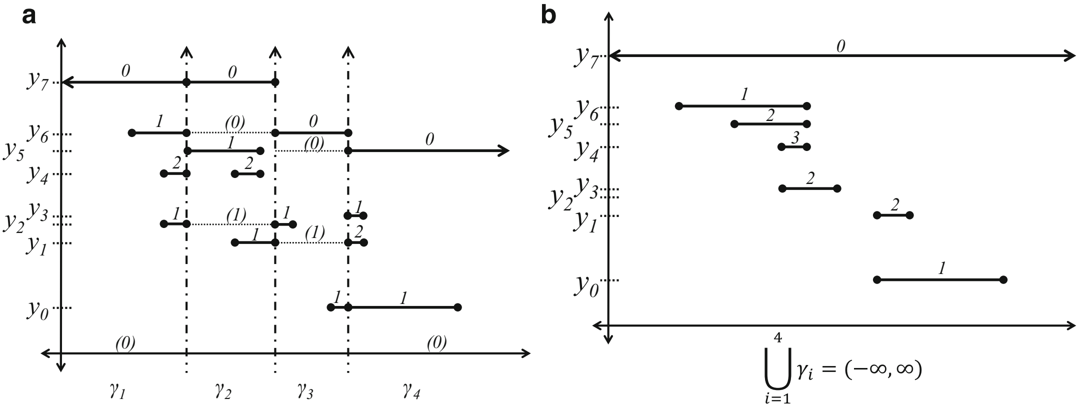
Figure 6 shows the slab-files that are generated from the example in Fig. 2, assuming that $m = 4$ and $\forall o \in O, w(o) = 1$. Max-intervals are represented as solid segments. For instance, the slab-file of slab γ_1 consists of tuples (in this order): $< y_2, [x_1, x_2], 1 >$, $< y_4, [x_1, x_2], 2 >$, $< y_6, [x_0, x_2], 1 >$, $< y_7, [-\infty, x_2], 0 >$. The first tuple $< y_2, [x_1, x_2], 1 >$ implies that, in slab γ_1, on any horizontal line with y-coordinate in (y_2, y_4), the max-interval is always $[x_1, x_2]$, and its *sum* is 1. Similarly, the second tuple $< y_4, [x_1, x_2], 2 >$ indicates that, on any horizontal line with y-coordinate in (y_4, y_6), $[x_1, x_2]$ is always the max-interval, and its *sum* is 2. Note that spanning rectangles have not been counted yet in these slab-files, since (as mentioned earlier) they are not part of the input to the sub-problems with regard to slabs $\gamma_1, \ldots, \gamma_4$.

- **Merging Phase** The merging phase is basically the process of choosing one max-interval for each h-line among all the max-intervals that are at the same h-line but from the different slab-file to be merged. To this end, the algorithm sweeps all the slab-files together with the file of spanning rectangles from bottom to top. Sometimes, max-intervals from adjacent slabs are combined into a longer max-interval.

Figure 7 shows how four slab-files in Fig. 6 are merged into one slab-file. For clarity, rectangles are removed, and the *sum* value of each max-interval is given above the segment representing the max-interval. The number enclosed in a bracket represents the total weight of spanning rectangles that span the corresponding slab at the h-line. For instance, between y_2 and y_6 in slab γ_2, there exists one spanning rectangle, which is why the number in the bracket above the dotted line at y_2 in γ_2 is 1. When the sweeping line ℓ is located at y_0, two max-intervals from γ_3 and γ_4 are merged into a larger max-interval. On the other hand, when ℓ is located at y_1, the max-interval from γ_4 is chosen, since its *sum* value 2 is the maximum among the two max-intervals at y_1. In addition, it is important to note that *sum* values of the max-intervals at y_4 and y_5 in γ_2 are increased by the total weight of corresponding spanning rectangles, which is 1. Figure 7b shows the resulting max-intervals at the end of merging slab-files. From the final slab-file, it turns out that the densest region of the entire data space is between max-intervals at y_4 and y_5, because the max-interval at y_4 has the highest *sum* value 3.



Maximizing Range Sum in Spatial Databases, Fig. 6 An example of slab-files



Maximizing Range Sum in Spatial Databases, Fig. 7 An example to illustrate the merging process. (**a**) Four slab-files before merge. (**b**) A slab-file after merge

Key Applications

An intuitive application of MaxRS is related to many kinds of facilities that should be associated with a region of a particular size. For example, if we open a new pizza franchise store that has a limited delivery range in a downtown area, it is important to maximize the number of residents in a rectangular area around the pizza store. This case is about finding a more profitable place to set up a new service facility.

For an opposite case, the MaxRS problem can be applied to find a more serviceable place for mobile users. Consider a tourist who wants to find the most representative spot in a city. In this case, the tourist will prefer to visit as many attractions as possible around the spot, and at the same time s/he usually does not want to go too far away from the spot.

In addition, MaxRS can also play an important role in spatial data mining where various kinds of massive location log datasets are employed.

Indeed, many works are being reported to mine interesting locations from a large amount of GPS log data of mobile subscribers. These works are mostly involved in extracting the *hot spot* from a massive dataset of points, which can be naturally abstracted as MaxRS. Also, some location datasets are associated with a set of events such as the traffic accident and the crime. For instance, when performing the geographic profiling in crime analysis, it is frequently required to determine the most probable area of offender residence by analysing the set of crime locations. This task is also strongly related to the MaxRS problem.

Future Directions

Choi et al. present an approximate solution for the circle version of the MaxRS problem in Choi et al. (2012), and the output-sensitive algorithm for the problem of finding all the *tied* densest regions, called *AllMaxRS*, in Choi et al. (2014). Tao et al. (2013) investigate the $(1 - \epsilon)$-*approximate* MaxRS problem, which admits the same inputs as MaxRS, but aims instead to return a rectangle whose covered weight is at least $(1-\epsilon)m^*$, where m^* is the optimal covered weight, and $\int$ can be an arbitrarily small constant between 0 and 1. Cao et al. (2014) apply the concept of MaxRS to a road network environment, and thereby propose the *length-constrained maximum-sum region* query.

Some promising directions for future works of MaxRS are as follows. First, a natural variant of MaxRS is minimizing the total weight of covered objects, namely the *MinRS* problem. Another direction can be a continuous version of MaxRS aiming at processing multiple MaxRS queries with different range constraints in real time by utilizing a preprocessed structure.

References

Arge L, Knudsen M, Larsen K (1993) A general lower bound on the I/O-complexity of comparison-based algorithms. In: Proceedings of algorithms and data structures (WADS), Montréal, Canada, pp 83–94

Barequet G, Dickerson M, Pau P (1997) Translating a convex polygon to contain a maximum number of points. Comput Geom 8(4):167–179

Cao X, Cong G, Jensen CS, Yiu ML (2014) Retrieving regions of interest for user exploration. PVLDB 7(9):733–744

Chazelle BM, Lee DT (1986) On a circle placement problem. Computing 36(1):1–16

Choi D-W, Chung C-W, Tao Y (2012) A scalable algorithm for maximizing range sum in spatial databases. PVLDB 5(11):1088–1099

Choi D-W, Chung C-W, Tao Y (2014) Maximizing range sum in external memory. ACM Trans Database Syst (TODS) 39(3):21

Dickerson M, Scharstein D (1998) Optimal placement of convex polygons to maximize point containment. Comput Geom 11(1):1–16

Du Y, Zhang D, Xia T (2005) The optimal-location query. In: International symposium of advances in spatial and temporal databases (SSTD), pp 163–180

Eppstein D, Erickson J (1994) Iterated nearest neighbors and finding minimal polytopes. Discret Comput Geom 11:321–350

Imai H, Asano T (1983) Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. J Algorithms 4(4): 310–323

Nandy SC, Bhattacharya BB (1995) A unified algorithm for finding maximum and minimum object enclosing rectangles and cuboids. Comput Math Appl 29(8): 45–61

Rocha-Junior JB, Vlachou A, Doulkeridis C, Nørvåg K (2010) Efficient processing of top-k spatial preference queries. PVLDB 4(2):93–104

Tao Y, Hu X, Choi D-W, Chung C-W (2013) Approximate maxrs in spatial databases. PVLDB 6(13):1546–1557

Wong RC-W, Tamer Özsu M, Yu PS, Fu AW-C, Liu L (2009) Efficient method for maximizing bichromatic reverse nearest neighbor. PVLDB 2(1):1126–1137

Xia T, Zhang D, Kanoulas E, Du Y (2005) On computing top-t most influential spatial sites. In: Proceedings of international conference on very large data bases (VLDB), pp 946–957

Xiao X, Yao B, Li F (2011) Optimal location queries in road network databases. In: Proceedings of international conference on data engineering (ICDE), pp 804–815

Yiu ML, Dai X, Mamoulis N, Vaitis M (2007) Top-k spatial preference queries. In: Proceedings of international conference on data engineering (ICDE), pp 1076–1085

Zhang D, Du Y, Xia T, Tao Y (2006) Progressive computation of the min-dist optimal-location query. In: Proceedings of international conference on very large data bases (VLDB), pp 643–654

Zhou Z, Wu W, Li X, Lee M-L, Hsu W (2011) Maxfirst for MaxBRkNN. In: Proceedings of international conference on data engineering (ICDE), pp 828–839

Maximum Update Interval

► Maximum Update Interval in Moving Objects Databases

Maximum Update Interval in Moving Objects Databases

Christian S. Jensen[1], Dan Lin[2], and
Beng Chin Ooi[2]
[1]Department of Computer Science, Aalborg University, Aalborg, Denmark
[2]Department of Computer Science, National University of Singapore, Singapore, Singapore

Synonyms

Maximum update interval

Definition

The maximum update interval in moving objects databases denotes the maximum time duration in-between two subsequent updates of the position of any moving object. In some applications, a variation of the maximum update interval denotes the time duration within which a high percentage of objects have been updated.

Main Text

In moving objects databases, there exist a population of moving objects, where each object is usually assumed to be capable of transmitting its current location to a central server. A moving object transmits a new location to the server when the deviation between its real location and its server-side location exceeds a threshold, dictated by the services to be supported. In general, the deviation between the real location and the location predicted by the server tends to increase as time passes. Even the deviation does not increase, it is also necessary to inform the server periodically that the object still exists in the system. In keeping with this, a *maximum update interval* is defined as a problem parameter that denotes the maximum time duration in-between two updates of the position of any moving object.

This definition is very helpful to index and application development especially those with functions of future prediction. Given such a time interval, trajectories of objects beyond their maximum update interval when the objects will definitely be updated usually do not need to be considered. In other words, the maximum update interval gives an idea of a time period of validity of current object information.

Cross-References

► Indexing, BDual Tree
► Indexing of Moving Objects, B^x-Tree
► Indexing the Positions of Continuously Moving Objects

MB-Index

► Indexing Schemes for Multidimensional Moving Objects

MBR

► Minimum Bounding Rectangle

MCMC

► Hurricane Wind Fields, Multivariate Modeling

MDA

► Modeling with Enriched Model-Driven Architecture

MDE

▶ Modeling with Enriched Model-Driven Architecture

Meaning, Multiple

▶ Uncertainty, Semantic

Median Center

▶ Geometric Median

Medical Image Dataset Processing over Cloud/MapReduce with Heterogeneous Architectures

Fusheng Wang[1], Ablimit Aji[2], and
George Teodoro[3]
[1]Stony Brook University, Stony Brook, NY, USA
[2]Hewlett Packard Labs, Palo Alto, CA, USA
[3]University of Brasília, Brasília-GF, Brazil

Synonyms

Cloud; CPU-GPU; Digital pathology; MapReduce; Pathology imaging; Spatial queries

Definition

Digital pathology images or whole slide images (Cooper et al. 2012b) are generated through scanning human tissue specimens with high-resolution microscope scanners. Examination of high-resolution whole slide images enables more effective diagnosis, prognosis, and prediction of cancer and other complex diseases (Cooper et al. 2011; Kong et al. 2013).

Pathology image analysis (Cooper et al. 2011; Kong et al. 2011) segments large number of spatial objects, such as nuclei and blood vessels, from whole slide images, along with many image features from these objects. Extracted spatial objects are represented with their geometric boundaries, and such spatially derived information is used in many analytical queries to support biomedical research (Cooper et al. 2012a; Kong et al. 2013) and exploration.

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. MapReduce (Dean and Ghemawat 2008) is a scalable programming model for processing large data sets with a parallel, distributed algorithm on a cluster.

General-purpose computing on graphics processing unit (GPGPU) refers to utilization of a graphics processing unit (GPU) to perform computation that are traditionally handled by the central processing unit (CPU). Modern computers often follow a heterogeneous architecture which combines both CPUs and GPUs. Spatial analytics mainly study entities and their relationship using their topological, geometric, or geographic properties. Typical spatial queries include spatial range queries, spatial join queries, and K-nearest neighbor queries (You et al. 2013; Puri and Prasad 2014, 2015; Audet et al. 2013). Spatial analytics of pathology images can take advantage of both cloud (Ray et al. 2013) computing supported by MapReduce and GPU computation in hybrid CPU-GPU computing architectures.

Historical Background

Pathology is a major field in modern medicine with a main focus on the causal study of disease. Microscopic examination of tissues helps pathologists to accurately diagnose the disease and guide therapy. While the basic process for pathologists to render diagnoses has remained almost unchanged over the past century, recent

Content is based on the work performed at Emory University as a Ph.D. student.

advances in digital pathology offer significant opportunities in image-based diagnosis and research applications. As high-resolution digital scanners become more affordable, pathology practices will increasingly adopt this technology and produce tremendous amount of whole slide images and analytical results. These advances create new opportunities for improving patient care and promoting biomedical research while posing significant challenges for data storage, management, and processing. Computer-based image analysis is already available in commercial diagnostic systems, but further advances in image analysis algorithms are warranted in order to fully realize the benefits of digital pathology in medical discovery and patient care.

Spatial database management systems (SDBMSs) have been used for managing and querying spatial data, through extended spatial capabilities on top of a relational database management system (RDBMS). While these system can be scaled out with a parallel architecture (Patel et al. 1997; Abouzeid et al. 2009), they have major limitation on managing and querying spatial data at massive scale. Parallel SDBMSs tend to reduce the I/O bottleneck through partitioning of data on multiple parallel disks and are not optimized for computationally intensive operations such as geometric computations. Furthermore, parallel SDBMS architecture lacks effective spatial partitioning mechanism to balance data and task loads across partitions and does not inherently support a way to handle boundary crossing objects. The high data loading overhead is another major bottleneck for SDBMS-based solutions (Pavlo et al. 2009). Scaling out spatial queries through a parallel database infrastructure is studied in Wang et al. (2011, 2013), but the approach is highly expensive and requires sophisticated tuning to achieve optimal performance.

More recently, with the rapid advancement of many hardware and software technologies, such as multi-core and many-core architectures, parallel programming environments, virtualization, and data center architectures, cloud computing has gradually become more mature, affordable, reliable, scalable, and widely available. This allows applications to process large-scale data in parallel at low cost by utilizing a large number of cloud computing resources on demand. In particular, parallel processing of spatial queries on the cloud has become promising for large-scale spatial applications.

Among the plethora of cloud computing technologies, MapReduce has become the prime choice for large-scale parallel data processing tasks. It has many advantages such as easy programmability, fault tolerance, and massive scalability. All the major cloud service providers offer MapReduce-based data processing platforms and services. Therefore, how to utilize MapReduce for scalable spatial query processing is the main focus of many cloud-based spatial query processing systems.

Scientific Fundamentals

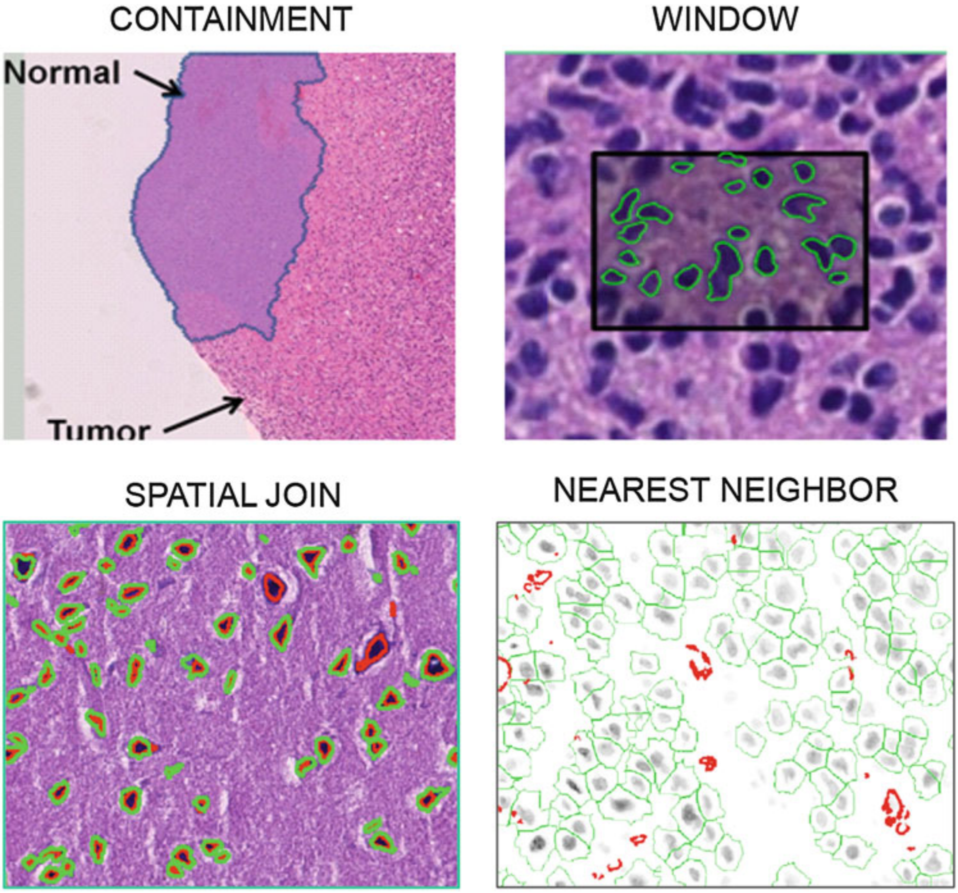
Query Cases

Pathology image analysis produces large-scale spatially derived information and bears resemblance to traditional "GIS" queries. Figure 1 demonstrates a few common queries.

(i) *Containment Query*: find objects contained in certain regions, for example, nuclei in tumor regions.

(ii) *Window Query*: retrieve objects contained in a window from a whole slide image, for example, download and visualize nuclei in current display window.

(iii) *Spatial Join Query (Spatial Cross-Matching)*: compare and evaluate segmentation results from multiple algorithms. For example, a query to compute the distance and intersection ratio of intersected boundaries segmented from an image by different algorithms is a common query type, and it is one of the most expensive query types. To compare two results from a single image, we are cross-matching a million spatial objects with another million spatial objects.



Medical Image Dataset Processing over Cloud/MapReduce with Heterogeneous Architectures, Fig. 1 Example queries of pathology image analytics

(iv) *Nearest Neighbor Queries*: find nearest neighbor objects of one type for objects of another type. An example query can be like this: for each stem cell (green objects in d), find the nearest blood vessel (red objects in d), compute the variation of intensity of each biological property associated with the cell in respect to the distance, and return the density distribution of blood vessels around each cell. This query involves millions of cells for a single image.

MapReduce-Based Spatial Query Processing

Most spatial query processing algorithms employ a two-step query processing strategy- *filter and refine*. During the *filter* step, spatial objects are approximated with simplified representations such as minimum bounding rectangles (MBRs), and an initial query processing is performed on the MBRs to generate a reduced set of candidate objects. Then, during the *refinement* step, the candidate object set is further pruned with accurate geometric operations to generate final query result.

This filter-and-refine strategy also forms the basis of spatial query parallelization at a higher level. Specifically, the underlying dataset is spatially partitioned into *partitions* that can be selectively processed by applying spatial partition-based filtering. Most often the spatial partitions are axis parallel rectangular regions (hyper-rectangles in multidimensional case). While the spatial partitions do not have to be rectangular, such simple representation is easy to compute and has a very small storage footprint. Such rectangular spatial partitioning is also referred as *tiling*.

Meanwhile, geometric computation algorithms such as intersection and union have high computational complexity. Existing CPU algorithms, such as those used in spatial data management systems, are branch intensive with irregular data access patterns. Most of those algorithms run in a single-threaded fashion, and they are hard to parallelize.

Hadoop-GIS (Aji et al. 2012, 2013, 2014; Vo et al. 2014) is a generic MapReduce-based framework for scalable, cost-effective, efficient, and expressive integrated spatial query processing system for data- and compute-intensive spatial applications, including spatial analytics for pathology imaging. Hadoop-GIS provides spatial data processing pipelines and querying methods through spatial partition level parallelism with MapReduce and integrates object level and intra-object level parallelism through GPUs.

Hadoop-GIS provides skew aware data partitioning methods to partition spatial objects such as polygons into buckets (or tiles), and process these buckets in parallel. Thus, generated tiles will become the unit for query processing. The query processing problem then reduces to designing querying methods that can run on these tiles independently while preserving the correct query semantics. A typical spatial query pipeline is shown in Algorithm 1.

In step A, an effective space partitioning is performed to generate tiles (Vo et al. 2014). In step B, spatial objects are assigned tile UIDs, merged and stored into Hadoop distributed file system (HDFS). Step C is for preprocessing queries, which could be queries that perform global index-based filtering, queries that do not need to run in tile-based query processing framework. Step D performs tile-based spatial query processing independently, which are parallelized through MapReduce. Step E provides handling of boundary objects (if needed), which can run as another MapReduce job. Step F does post-query processing, for example, joining spatial query results with feature tables. Step G performs data aggregation on the query results, and final results are written to HDFS.

Algorithm 1 Typical workflow of spatial query processing on MapReduce

```
A. Data/space partitioning;
B. Data storage of partitioned data on HDFS;
C. Pre-query processing (optional);
D. for tile in input_collection do
       Index building for objects in the tile;
       Tile-based spatial querying processing;
E. Boundary object handling;
F. Post-query processing (optional);
G. Data aggregation;
H. Result storage on HDFS;
```

Spatial Data Partitioning

Spatial data partitioning is an essential initial step to define, generate, and represent partitioned data. There are two major considerations for spatial data partitioning. The first consideration is to avoid high-density partitioned tiles. This is mainly due to the high data skew in spatial datasets, which can cause load imbalance in a cluster environment. Another consideration is to handle boundary intersecting objects properly. As MapReduce provides its own job scheduling for balancing tasks, the load imbalance problem can be partially alleviated at the task scheduling level. Therefore, for spatial data partitioning, the main focus is to further partition high-density tiles into smaller ones. For boundary intersecting objects, multiple-assignment, single-join approach (MASJ) (Zhou et al. 1998) is chosen – objects are replicated and assigned to each intersecting tile, processed in a single-join step, and post-processed to filter duplicate objects. There are two major categories of partitioning methods: top-down approach which recursively partitions the dataset into several buckets until the number of objects in each bucket reaches certain threshold and bottom-up approach which starts with very fine granular small partitions and packs the smaller ones to form bigger partitions. Both approaches can also be implemented in MapReduce (Vo et al. 2014).

Real-time Spatial Query Engine

A fundamental component of Hadoop-GIS is a stand-alone spatial query engine called real-time spatial query engine (RESQUE) to support spatial query processing. RESQUE takes advantage of global tile indexes and local indexes created on demand to support efficient spatial queries. Besides, RESQUE supports data compression and comes with very low overhead on data loading. This makes RESQUE a highly efficient spatial query engine compared to a traditional SDBMS engine. RESQUE is compiled as a shared library which can be easily deployed in a cluster environment. Hadoop-GIS takes advantage of spatial access methods for query processing with two approaches. At the higher level, Hadoop-GIS creates global region-based spatial indexes of partitioned tiles for HDFS file split filtering. As a result, for many spatial queries such as containment queries, most irrelevant tiles can be efficiently filtered through this global region index. The global region index is small and can be stored in a binary format in HDFS and shared across cluster nodes through Hadoop distributed cache mechanism. At the tile level, RESQUE supports an on-demand indexing approach by building in-memory tile-based spatial indexes on the fly, mainly for query processing purpose. Since the tile size is relatively small, index building on a single tile is very fast and it significantly improves spatial query performance. With the increasing speed of CPU, indexing building overhead is a very small fraction of overall query processing cost for compute- and data-intensive spatial queries such as cross matching.

MapReduce-Based Parallel Query Execution

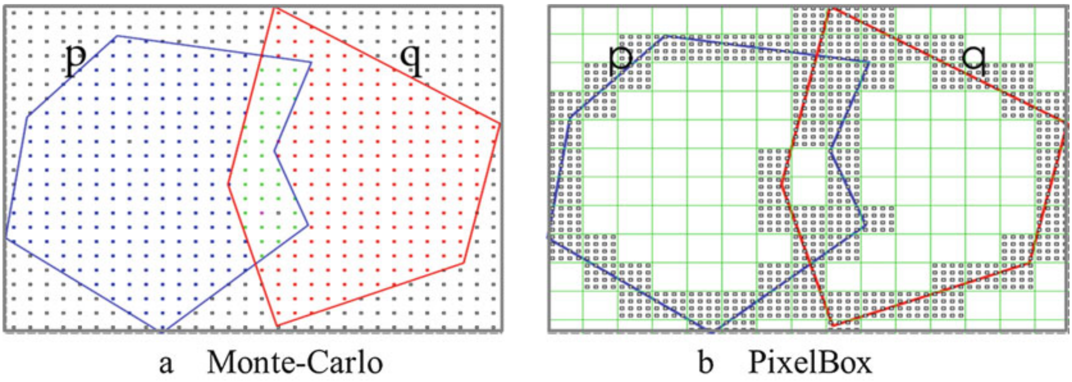
Instead of using explicit spatial query parallelization as summarized in Brinkhoff et al. (1996), we take an implicit parallelization approach by leveraging MapReduce. As data is spatially partitioned, the tile name or UID forms the key for MapReduce, and identifying spatial objects of tiles can be performed in mapping phase. Depending on the query complexity, spatial queries can be implemented as map functions, reduce functions, or combination of both. As many spatial queries involve data-intensive processing and high-complexity geometric computations, query parallelization through MapReduce can significantly reduce query response time.

Boundary Object Handling

In the past, two approaches were proposed to handle boundary objects in a parallel query processing scenario, namely, multiple assignment and multiple matching (Lo and Ravishankar 1996; Zhou et al. 1998). In multiple assignment, the partitioning step replicates boundary-crossing objects and assigns them to multiple tiles. In multiple matching, partitioning step assigns a boundary-crossing object to a single tile, but the object may appear in multiple tile pairs for spatial joins. While the multiple matching approach avoids storage overhead, a single tile may have to be read multiple times for query processing, which could incur increase in both computation and I/O. The multiple-assignment approach is simple to implement and fits nicely with the MapReduce programming model. For example, spatial join on tiles with multiple-assignment-based partitioning can be corrected by eliminating duplicated object pairs from the query result which can be integrated into the query processing pipeline as a MapReduce job.

GPU-Based Geometric Computation

GPUs employ a *SIMD* architecture that executes the same instruction logic on a large number of cores simultaneously. Many spatial algorithms and geometry computations do not naturally fit into such parallelization model. One typical GPU-based approach is through *rasterization*, which transforms spatial space into pixel-based representation. After such transformation, the original vector geometric computation, such as intersection and distance, can now be performed on the pixel-based representation, and it can be easily parallelized on the GPU. For example, in Fig. 2a, two polygons are rasterzied to calculate the intersection. The intersection can be determined by counting number of pixels that belong to both polygons. A common approach



Medical Image Dataset Processing over Cloud/MapReduce with Heterogeneous Architectures, Fig. 2 GPU-based geometric computation. (**a**) Monte-Carlo. (**b**) PixelBox

to check if a pixel lies within a polygon is to use ray tracing (Purcell et al. 2002) for point-in-polygon test. As the operation for each pixel is fully independent from each other, the query can be efficiently executed in parallel by large number of GPU threads. This approach is also called Monte-Carlo-based approach (Wang et al. 2012).

In such pixel-based GPU algorithms, rasterization resolution is critical for achieving best performance. A high-resolution rasterization yields larger number of pixels and consequently increases the compute intensity of the geometry computations. A low-resolution rasterization is computationally efficient, while it is less accurate for geometry calculations. In Wang et al. (2012), a more adaptive approach – named PixelBox – is used to reduce the computation intensity while ensuring the computational accuracy. Specifically, PixelBox (Fig. 2) first partitions the space into cells or boxes. Only for boxes containing edges of polygons, rasterization is performed as in the Monte-Carlo-based approach. In this way, group of pixels in a box could be tested together for the containment relationship with a polygon, and pixel level testing is performed only for edge-crossing areas. Thus, the computational efficiency could be much improved. The experiments in Wang et al. (2012) demonstrate two orders of performance improvement for intersection operation compared to a single-threaded CPU algorithm.

Integration of GPU-Based Geometric Computation into MapReduce

In order to support efficient query execution on GPU-accelerated systems, we have extended Hadoop-GIS to utilize the GPU device to process expensive spatial operations. The MapReduce layer employs a tile-based parallelization approach which distributes data partitions among cluster nodes. Thus, each partition is processed as a single MapReduce task, and each task can be independently scheduled to run on the CPU or GPU device. In our query engine, we have implemented both CPU-based algorithms and GPU-based algorithms for processing spatial queries. The query optimizer is the component responsible for selecting the device to execute a query task. This decision depends on a number of factors such as potential speedup gain from selecting a particular device, device availability, and data movement cost between CPU and GPU devices. The scheduler is critical for overall system performance, and the scheduling decision needs to be optimal. In our system, we take a predictive modeling approach. Specifically, we sample small number of tasks, and we execute them on both CPU and GPU devices. Meanwhile, we profile the tasks and collect information such as runtime, input data size, and characteristics. We feed those information to a polynomial line fitting algorithm to derive the performance model. Given such model, the scheduler assigns tasks to computing devices in a demand-driven basis, such that the task processed by a device is the one that benefits the most from the processor chosen (i.e., the task with the highest acceleration on the GPU).

Hybrid Execution of Core Image Analysis Operations

Spatial analytics is performed based on spatially derived information from whole slide images, through image segmentation algorithms. Such image analysis can also fully take advantage of hybrid CPU-GPU architecture. Pathology image analysis consists of a number of computation steps that include normalization, segmentation, and feature extraction. Additionally, these phases of the applications extract several spatial characteristics of objects and regions of interest, which are used for further analyses via spatial queries. In order to fully optimize the execution of these applications in hybrid systems, equipped with CPUs and GPUs, we have developed a framework for efficient execution of these pipelines with integration with our storage and query layers. In this framework, the pathology applications are described as workflows of coarse-grained computing stages that may be decomposed into a set of fine-grain computing tasks, which are assigned for execution on CPUs and GPUs available in the environment. The concept of function variants is used in order to allow for the execution of a task in multiple devices. In this context, a function variant is allowed to be developed for each available target processors. As a part of the effort to enable large-scale execution of the applications on hybrid systems, core operations have been developed to run on multiple processors: CPUs, GPUs, and Intel Phi (Teodoro et al. 2013a, 2014a, 2012).

The framework includes a number of optimizations for hybrid systems: performance-aware scheduling, data locality conscious task assignment, and data prefetching (Teodoro et al. 2012, 2013b, 2014b). The performance-aware scheduling is motivated by the different performance improvements (speedups) attained by operations when executed in a GPU. This performance gap between different query operators is a direct result of the different processing and data access patterns used in the operations. Therefore, a performance-aware scheduling has been developed to take into account performance variabilities among tasks to maximize the utilization of hybrid system. This scheduler maintains tasks ready for execution ordered according to their expected speedup on each available device and computes a scheduling during the execution in a demand-driven fashion in which tasks are assigned to processor that accelerate them the most. The data locality aware task assignment is designed to avoid unnecessary data transfers among CPU and GPU. If a task is ready for execution, and the input data is already in the GPU memory, it is given a higher priority for execution in that device. This priority change, however, is limited by the impact of the data transfers on task performance. Finally, we employ asynchronous data transfers between devices to improve query performance.

Key Applications

The emergence of pathology analytical imaging, fostered by the advent of cost-effective digital scanners, has enabled large-scale quantitative and integrative scientific investigations which depend on high-throughput analysis of imaging features and annotations derived from two- and three-dimensional datasets. High-performance and scalable computing solutions are critical for analyzing and processing imaging and spatial data at a massive scale. The MapReduce-based spatial analytics framework with integration of CPU computing could facilitate the wide application of microscopic imaging data analysis within the clinical setting and scientific communities.

The prevalence of cost-effective and ubiquitous positioning technologies such as GPS, RFID, and more recently smartphones has enabled enterprises, governments, and scientists to capture spatially oriented data at an unprecedented scale and rate. Large-scale geo-crowdsourcing or volunteered geographical information (VGI) (Goodchild 2007), such as OpenStreetMap, has created high potential for establishing reliable source of information.

Geospatial big data can be roughly classified into the following categories: (1) actively generated spatial data such as social media data; (2) passively generated spatial data, such as call logs and transaction logs; and (3) geo-crowdsourced spatial data, such as OpenStreetMap, Wikimapia, and Waze. In addition, there is also a large collection of traditional spatial data produced through central organizations such as government agencies or private companies, which includes census data and public health data. Analyzing large amounts of geospatial data to derive values and guide decision making has become essential to business success and scientific discoveries. For instance, location-based social networks (LBSNs) are utilizing large amounts of user location information to provide geo-marketing and recommendation services. Social scientists can rely on such data to study the dynamics of social systems and understand human behaviors. Public health researchers can use social media to understand the distribution and propagation of infectious diseases and mental diseases and and discover humans' health behavior (Sadilek and Kautz 2013). Support of efficient and scalable spatial queries and analytics is critical for deriving values from such geospatial data. Hadoop-GIS provides an ideal solution for these applications.

In addition, we plan to extend our query engine system module responsible for coordinating the execution with hybrid systems, equipped with CPUs and GPUs. Our extension to this problem will employ a hierarchical inter-/intra-node parallelization leveraging MapReduce and our smart performance-aware scheduling and runtime systems (Teodoro et al. 2013b). MapReduce will be used for distribution of work among nodes of a distributed environment, whereas the intra-node parallelization will be carried out by our runtime system and scheduling strategies that are able to represent a task as another workflow of fine-grain operations. To efficiently utilize hybrid systems, we plan to exploit intra-/inter-query performance variabilities. Because the execution of a single query is a complex process composed of several fine-grain operations, it is likely that these operations have different computation and data access patterns, and, as a consequence, they will attain different levels of acceleration on a GPU. Additionally, a query execution engine may support the execution of different types of queries in parallel, and the types of queries are also likely to achieve different levels of acceleration on a GPU. Thus, we will use our performance-aware scheduling algorithms to learn and use these performance variability to better utilize the aggregate power of hybrid machines.

Future Directions

Frequently, spatial data analytics require more complex algorithms such as spatial clustering, spatial correlation analysis, and spatial regression. While the compute-intensive nature of those algorithms makes them a natural candidate for GPU acceleration, the recursive nature of those algorithms requires more effort to achieve reasonable speedup. Furthermore, the amount of device memory available in the GPU is often a hard restriction on the complexity of the models that one can build for analysis. Therefore, how to leverage GPU for such complex spatial analytics tasks and efficiently orchestrate data movement between CPU and GPU can be a very promising and challenging future work.

Cross-References

- ▶ Data Warehouses and GIS
- ▶ Polygonal Overlay Computation on Cloud, Hadoop, and MPI
- ▶ Spatial Join with Hadoop

References

Abouzeid A, Bajda-Pawlikowski K, Abadi D, Silberschatz A, Rasin A (2009) HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads. Proc VLDB Endow 2(1):922–933

Aji A, Wang F, Saltz JH (2012) Towards building a high performance spatial query system for large scale medical imaging data. In: SIGSPATIAL/GIS, Redondo Beach, pp 309–318. ACM

Aji A, Wang F, Vo H, Lee R, Liu Q, Zhang X, Saltz J (2013) Hadoop-GIS: a high performance spatial data warehousing system over MapReduce. Proc VLDB Endow 6(11):1009–1020

Aji A, George T, Wang F (2014) Haggis: turbocharge a MapReduce based spatial data warehousing system with GPU engine. In: ACM SIGSPATIAL international workshop on analytics for big geospatial data (BigSpatial'14), Redondo Beach

Audet S, Albertsson C, Murase M, Asahara A (2013) Robust and efficient polygon overlay on parallel stream processors. In: Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems (SIGSPATIAL'13). ACM, New York, pp 304–313

Brinkhoff T, Kriegel H-P, Seeger B (1996) Parallel processing of spatial joins using R-trees. In: ICDE, Redondo Beach

Cooper L, Kong J, Moreno C, Wang F, Kurc T, Saltz J, Brat D (2011) In silico analysis of nuclei in glioblastoma using large-scale microscopy images improves prediction of treatment response. In: EMBC, Redondo Beach

Cooper LAD, Kong J, Gutman DA, Wang F, Gao J, Appin C, Cholleti S, Pan T, Sharma A, Scarpace L, Mikkelsen T, Kurc T, Moreno CS, Brat DJ, Saltz JH (2012a) Integrated morphologic analysis for the identification and characterization of disease subtypes. J Am Med Inform Assoc 19(2):317–323

Cooper LAD, Carter AB, Farris AB, Wang F, Kong J, Gutman DA, Widener P, Pan TC, Cholleti SR, Sharma A et al (2012b) Digital pathology: data-intensive frontier in medical imaging. Proc IEEE 100(4):991–1003

Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211–221

Kong J, Cooper L, Wang F, Chisolm C, Moreno C, Kurc T, Widener P, Brat D, Saltz J (2011) A comprehensive framework for classification of nuclei in digital microscopy imaging: an application to diffuse gliomas. In: ISBI, Redondo Beach

Kong J, Cooper LAD, Wang F, Gao J, Teodoro G, Scarpace L, Mikkelsen T, Schniederjan MJ, Moreno CS, Saltz JH et al (2013) Machine-based morphologic analysis of glioblastoma using whole-slide pathology images uncovers clinically relevant molecular correlates. PLoS One 8(11):e81049

Lo M-L, Ravishankar CV (1996) Spatial hash-joins. In: SIGMOD, Redondo Beach, pp 247–258

Patel J et al (1997) Building a scaleable geo-spatial dbms: technology, implementation, and evaluation. In: SIGMOD, Redondo Beach, pp 336–347

Pavlo A, Paulson E, Rasin A, Abadi DJ, DeWitt DJ, Madden S, Stonebraker M (2009) A comparison of approaches to large-scale data analysis. In: SIGMOD, Redondo Beach, pp 165–178

Purcell TJ, Buck I, Mark WR, Hanrahan P (2002) Ray tracing on programmable graphics hardware. ACM Trans Graph (TOG) 21:703–712. ACM

Puri S, Prasad SK (2014) GIS polygon overlay processing: new parallel algorithm and system prototype

Puri S, Prasad S (2015) A parallel algorithm for clipping polygons with improved bounds and a distributed overlay processing system using mpi. In: 15th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid), Redondo Beach. http://cs.gsu.edu/~spuri2/publications/ParallelGH_PI-GIS.pdf

Ray S, Simion B, Brown AD, Johnson R (2013) A parallel spatial data analysis infrastructure for the cloud. In: Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems (SIGSPATIAL'13). ACM, New York, pp 284–293

Sadilek A, Kautz H (2013) Modeling the impact of lifestyle on health at scale. In: Proceedings of the sixth ACM international conference on Web search and data mining, Redondo Beach. ACM, pp 637–646

Teodoro G, Kurc TM, Pan T, Cooper LAD, Kong J, Widener P, Saltz JH (2012) Accelerating large scale image analyzes on parallel, CPU-GPU equipped systems. In: 26th IEEE international parallel and distributed processing symposium (IPDPS), Redondo Beach, pp 1093–1104

Teodoro G, Pan T, Kurc T, Kong J, Cooper L, Saltz J (2013a) Efficient irregular wavefront propagation algorithms on hybrid CPU-GPU machines. Parallel Comput 39(4):189–211

Teodoro G, Pan T, Kurc TM, Kong J, Cooper LAD, Podhorszki N, Klasky S, Saltz JH (2013b) High-throughput analysis of large microscopy image datasets on CPU-GPU cluster platforms. In: proceedings of the 2013 IEEE international symposium on parallel and distributed Processing (IPDPS'13), Redondo Beach

Teodoro G, Kurc T, Kong J, Cooper L, Saltz J (2014a) Comparative performance analysis of Intel (R) Xeon Phi (TM), GPU, and CPU: a case study from microscopy image analysis. In: 2014 IEEE 28th international parallel and distributed processing symposium (IPDPS '14), Redondo Beach, pp 1063–1072

Teodoro G, Pan T, Kurc T, Kong J, Cooper L, Klasky S, Saltz J (2014b) Region templates: data representation and management for high-throughput image analysis. Parallel Comput 40(10):589–610

Vo H, Aji A, Wang F (2014) SATO: a spatial data partitioning framework for scalable query processing. In: SIGSPATIAL/GIS, Redondo Beach. ACM

Wang F, Kong J, Cooper L, Pan T, Tahsin K, Chen W, Sharma A, Niedermayr C, Oh TW, Brat D, Farris AB, Foran D, Saltz J (2011) A data model and database for high-resolution pathology analytical image informatics. J Pathol Inform 2(1):32

Wang K, Huai Y, Lee R, Wang F, Zhang X, Saltz JH (2012) Accelerating pathology image data cross-comparison on CPU-GPU hybrid systems. Proc VLDB Endow 5(11):1543–1554

Wang F, Kong J, Gao J, Adler D, Cooper L, Vergara-Niedermayr C, Zhou Z, Katigbak B, Kurc T, Brat D, Saltz J (2013) A high-performance spatial database based approach for pathology imaging algorithm evaluation. J Pathol Inf 4(5)

You S, Zhang J, Gruenwald L (2013) Parallel spatial query processing on GPUs using R-trees. In: Proceedings of the 2nd ACM SIGSPATIAL international workshop on analytics for big geospatial data (BigSpatial'13). ACM, New York, pp 23–31

Zhou X, Abel DJ, Truffet D (1998) Data partitioning for parallel spatial join processing. GeoInformatica 2(2):175–204

Memory, External

► Indexing Schemes for Multidimensional Moving Objects

Mereotopology

Anthony G. Cohn
School of Computing, University of Leeds,
Leeds, UK

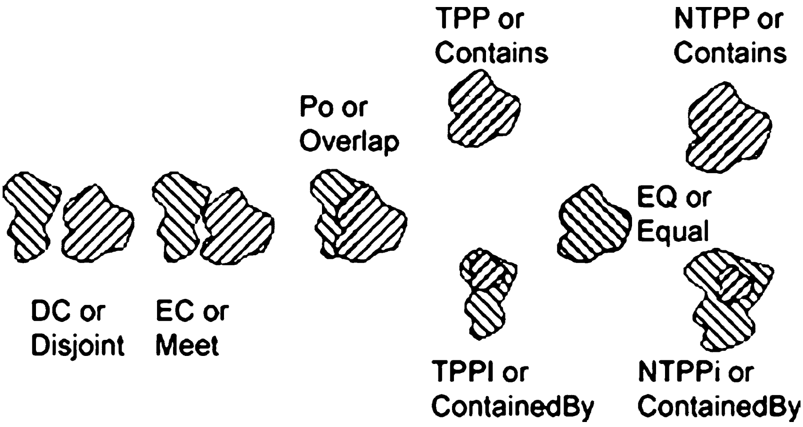
Synonyms

4-Intersection calculus; 9-Intersection calculus; Pointless topology; RCC; Region connection calculus

Definition

Topology, which is founded on the notion of connectedness, is at the heart of many systems of qualitative spatial relations; since it is possible to define a notion of parthood from connection, and theories of parthood are called mereologies, such combined theories are generally called mereotopologies. The best known set of relations based on a primitive notion of connectedness is the Region Connection Calculus (RCC), which defines several sets of *jointly exhaustive and pairwise disjoint, (JEPD)* relations, RCC-5, a purely mereological set, and the more widely used RCC-8 set of eight relations illustrated in Fig. 1. The primitive relation used in RCC (and several related theories) is $\mathsf{C}(x, y)$ – true when region x is connected to region y. A largely equivalent set of relations can be defined in the 4-intersection model in which relations between regions are defined in terms of whether the intersections of their boundaries and interiors are empty or non empty; after taking into account the physical reality of 2D space and some specific assumptions about the nature of regions, it turns out that the there are exactly eight remaining relations, which correspond to the RCC-8 relations. A generalization (the 9-intersection model) also considers the exterior of regions too, and allows further distinctions and larger sets of JEPD relations to be defined. For example, one may derive a calculus for representing and reasoning

Mereotopology, Fig. 1 A 2D depiction of RCC-8 relations or the eight topological relations of the 4 and 9-intersection calculi. The *arrows* show the conceptual neighborhood structure



about regions in $\mathbb{Z}^2$ rather than $\mathbb{R}^2$, or between spatial entities of different dimensions (such as relations between lines and regions).

Cross-References

- ▶ Conceptual Neighborhood
- ▶ Knowledge Representation, Spatial
- ▶ Representing Regions with Indeterminate Boundaries

Recommended Reading

Cohn AG, Hazarika SM (2001) Qualitative spatial representation and reasoning: an overview. Fundam Inf 46(1–2):1–29

Cohn AG, Renz J (2007) Qualitative spatial representation and reasoning. In: Lifschitz V, van Harmelen F, Porter F (eds) Handbook of knowledge representation, chap 13. Elsevier, München

Merge Designs

▶ Contraflow for Evacuation Traffic Management

Message Passing Interface

▶ MPI in GIS

Metadata

Christopher J. Semerjian
Institute for Environmental & Spatial Analysis,
Gainesville State College, Gainesville, GA, USA

Synonyms

Geographic metadata; Geospatial metadata

Definition

A metadata record is a file of information, which captures the basic characteristics of a data or information resource. It represents the who, what, when, where, why, and how of the resource. Metadata is known as "data about data" (Wilson 2004). It describes the content, quality, conditions, location, author, and other characteristics of data. Geospatial metadata is metadata that describes data or objects with geographic attributes, such as a geographic extent or a fixed location on the Earth. This geographic attribute may be a location such as latitude and longitude, a street address, or a geographic position relative to other objects. Geospatial metadata are used to document geographic digital resources such as Geographic Information System (GIS) files, raster and vector alike, and other geospatial databases (Federal Geographic Data Committee 2006a). Metadata makes spatial information more useful to all types of users by making it easier to document and locate data sets. Metadata helps people who use geospatial data to find the data they need and determine how best to use it (Federal Geographic Data Committee 2005). It is also important because it protects the investment in data, it helps the user understand data, and it enables discovery (Metadata Education Project, Wyoming Geographic Information Science Center 2006). The creation and management of metadata is both an essential and required part of GIS functionality.

Historical Background

In Greek epistemology the prefix *meta* means *about*. Thus the term metadata can be literally interpreted as "about data". Metadata is any information that describes data. Historically metadata was seen as additional information to supplement data, but not necessarily an essential part of the data. Recent advances in information technology and the rapid emergence of the digital library have somewhat altered the perception of metadata among information managers; metadata is no longer auxiliary definitions or descriptions of some library resource, but a fundamental dimension of said resource (United States Geologic Survey, Content Metadata Standards for Marine Science). An early use of metadata in the digital world occurred in the 1960s, with the advent of the international Machine-Readable Cataloging (MARC) standards and the Library of Congress Subject Headings (LCSH) (United States Geologic Survey, Content Metadata Standards for Marine Science).

A major factor in the functionality of a Geographic Information System is data interoperability (Danko). Geospatial data comes from a variety of sources in a variety of formats. Metadata is the key to maintaining interoperability by identifying standards and recording the information necessary to ensure information exchange. In the late 1970s, many government agencies in United States (US) started initiating digital mapping programs. In 1983, the Office of Management and Budget established a committee to coordinate digital cartographic activities among the US federal agencies in an effort to keep track of the enormous growth of digital geospatial data. They were to oversee any problems associated with the duplication of effort, lack of standards and inadequate interagency coordination, etc. The Office of Science and Technology Policy study recommended a centralized data base and schema to overcome such problems (Wilson 2004). Thus, in 1983, the Federal Interagency Coordinating Committee on Digital Cartography (FICCDC) was established to coordinate the GIS data development activities (Wilson 2004). In 1990, FICCDC was evolved into Federal Geographic Data Committee (FGDC) (Federal Geographic Data Committee 2006b).

Beginning in 1994, Executive Order 12906 requires federal agencies to produce standardized metadata for all new geospatial data they create. The National Spatial Data Infrastructure (NSDI) was created in the same year to coordinate in collection, sharing, and use of GIS data among federal or non-government agencies (Wilson 2004).

Scientific Fundamentals

The FGDC promoted the development, use, sharing, and dissemination of geospatial data on a national basis (Federal Geographic Data Committee 2006b). The FGDC developed draft content standards for geospatial metadata in the fall of 1992. “The objectives of the standard are to provide a common set of terminology and definitions for the documentation of digital geospatial data. The standard establishes the names of data elements and compound elements (groups of data elements) to be used for these purposes, the definitions of these compound elements and data elements, and information about the values that are to be provided for the data elements” (Federal Geographic Data Committee 2006c). After a public comment period and revision, the FGDC approved the standard on June 8, 1994.

The standard utilizes ten categories to describe a geospatial data set. These categories are:

1. *Identification* – basic information including the title, author, abstract, purpose, geographic extent, data collection dates, status, completion date, publication date, and access constraints.
2. *Data Quality* – data quality information including positional accuracy, attribute accuracy, processing steps, and data lineage.
3. *Spatial Data Organization* – information on the spatial reference method used to depict geographic features (raster or vector) and a count of spatial features.
4. *Spatial Reference* – Horizontal and vertical coordinate system information including projection parameters, coordinate system parameters, and geodetic datum.
5. *Entity and Attribute Information* – a description of each attribute in the data table including the attribute name, data type, syntax, width, precision and domain.
6. *Distribution* – data access information including distribution formats, distribution locations, hyperlinks, and costs.
7. *Metadata Reference* – metadata compilation information including date and author.

8. *Citation Information* – the reference for the dataset including publication information and online linkages.
9. *Time Period Information* – metadata about any temporal attributes of the data set.
10. *Contact Information* – identity of contact persons or organizations associated with the data set.

Geospatial metadata will typically be stored in an XML, TXT, or HTML file with the same filename as the associated dataset. The FGDC does not specify how their ten standard metadata categories should be formatted within the metadata file. However, the FDGC does use several stylesheets for geospatial metadata. Several metadata creation tools and metadata stylesheets are available from the FGDC website or from within various geospatial software packages such as ArcGIS™, AutoDesk™, ERDAS™, and Intergraph™.

The Federal Geographic Data Committee website (http://www.fgdc.gov) is an excellent resource for geospatial metadata standards. The site contains geospatial metadata information including an overview, metadata history, importance of metadata, technical documentation, examples, and standards. A detailed description of the FGDC metadata standard can be found on the site under *Content Standard for Digital Geospatial Metadata*. A glossary and list of metadata elements are also provided.

Other organizations that have developed standards for geospatial metadata include the International Organization for Standardization (ISO) and the Open Geopstial Consortium (OGC). The International Organization for Standardization is an international standard-setting body that produces world-wide standards for commerce and industry. ISO/TC 211 is a standard technical committee formed within ISO, tasked with covering the areas of digital geographic information (such as used by geographic information systems) and geomantic (International Organization for Standardization 2007). Publication ISO 19115:2003 defines the schema required for describing geographic information and services; it provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data (International Organization for Standardization 2007). Publication ISO 19115:2003 can be accessed through the ISO website: (http://www.isotc211.org). The Open Geospatial Consortium, Inc is an international industry consortium of 339 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications (Open Geospatial Consortium, Inc. 2007). The consortium addresses issues and sets standards related to how metadata must be specified in a GIS. These recommendations and standards can be found in the Metadata WG specifications, located on the OGC website (http://www.opengeospatial.org). The geospatial metadata standards developed by the International Organization for Standardization and the Open Geospatial Consortium are tied closely to the standards developed by the Federal Geographic Data Committee and are identical in many regards. The ISO 19115 specifications are quickly becoming the world standard for geospatial metadata.

Other sources for metadata standards include the Digital Geographic Information Working Group (DGIWG). The DGWG was established in 1983 to develop standards to support the exchange of digital geographic information among nations, data producers, and data users (Digital Geographic Information Exchange Standard 1994). The DGIWG published the Digital Geographic Information Exchange Standard (DIGEST) in 1994. The DIGEST can be accessed from the DGIWG website (https://www.dgiwg.org/digest/). The Massachusetts Institute of Technology Libraries Metadata Advisory Group maintains links to many other metadata standards documents. These can be accessed through the MIT Libraries website (http://libraries.mit.edu/guides/subjects/metadata/standards.html).

From a data management perspective, metadata is important for maintaining an organization's investment in spatial data. Metadata is a summary document providing content, quality, type, creation, and spatial information about a dataset. Therefore, metadata benefits an organization in the following ways:

1. Provides an inventory of data assets.
2. Helps determine and maintain the value of data.
3. Helps users and creators to determine the reliability and currentness of data.
4. Supports decision making.
5. Documents legal issues.
6. Helps keep data accurate and helps verify accuracy to support good decision making and cost savings.
7. Helps determine budgets because it provides a clearer understanding of when or if data needs to be updated or repurchased.

ESRI's GIS internet mapping service, ArcIMS^{TM} also hosts a metadata service. This allows companies and organizations to serve geospatial metadata through web for easier public viewing. The ESRI supported spatial database engine, ArcSDE^{TM} is the interface to the relational database that stores geospatial metadata documents for organizations. The ArcIMS metadata service uses the ArcSDE database as repository. ArcCatalog^{TM}, Metadata Explorer, Web browsers, or Z39.50 clients can access metadata stored in a metadata service.

ArcGIS has been designed to create metadata for any data set supported/created by ArcGIS as well as any other data set identified and cataloged by the user (e.g., text, CAD files, scripts). Metadata can be created for several different datasets, such as ArcInfo coverages, ESRI shape files, CAD drawings, images, grids, TINs, ArcSDE geodatabases, personal Geodatabase, maps, workspaces, folders, layers, INFO, dBASE, and DBMS tables, projections, text files, programming scripts, etc.

Key Applications

Metadata can be created for any data or information resource and is routinely used to describe or provide functional information for all types of digital data. Metadata allows more efficient query and filter applications for any digital data source. Therefore it is commonly used by libraries, corporate networks, and internet search engines for faster data retrieval and processing. Here are some of the key non-geospatial application areas for metadata.

Traditional Databases

Metadata is used in traditional database management systems such as relational database systems to store information on the size, structure, location, modification dates, and number of tables in the database system.

Operating Systems

Operating systems such as Windows or Linux store metadata on all files and folders. This includes permissions, security, display, and time stamp information for files and folders.

Digital Documents and Images

Metadata is utilized by word processors, spreadsheets, imaging software, etc. to describe key elements of the document such as creation date, modification date, author, font, spacing, size, security, etc. Metadata is commonly generated by web-browsers, peer-to-peer software, and multimedia indexing software.

Future Directions

Most available metadata creation tools are designed to produce metadata for one data element at a time. The need exists to develop tools to manage metadata for numerous objects more effectively and efficiently (Idaho Geospatial Data Clearinghouse, Interactive Numeric & Spatial Information Data Engine 2006). Furthermore, the use of metadata is becoming more widespread and standardized and there is an increasing demand for automated metadata creation tools. Some of these tools have already been developed and are already freely available on the internet (Idaho Geospatial Data Clearinghouse, Interactive Numeric & Spatial Information Data Engine 2006). The United States Geologic Survey provides several tools, tips and tricks on their metadata information and software page (United States Geologic Survey, Formal metadata 2006). Another tool used for the batch creation and maintenance of metadata within ArcGIS is available for download from the Idaho Geospatial Data Clearinghouse (Idaho Geospatial Data Clearinghouse, Interactive Numeric & Spatial Information Data Engine 2006). It is anticipated that many other metadata tools will become available in the future. By the printing of this book, many other metadata tools will be available online.

Cross-References

- ▶ Data Warehouses and GIS
- ▶ Metadata and Interoperability, Geospatial

References

Danko D, Senior consultant, GIS Standards, Environmental Systems Research Institute. ISO Metadata Workshop

Digital Geographic Information Exchange Standard (DIGEST), Version 1.2 (1994). https://www.dgiwg.org/digest. Digital Geographic Information Working Group, Jan

Federal Geographic Data Committee (2005) Business case for metadata. http://www.fgdc.gov/metadata/metadata-business-case

Federal Geographic Data Committee (2006a) Geospatial metadata. http://www.fgdc.gov/metadata

Federal Geographic Data Committee (2006b) The Federal Geographic Data Committee. http://www.fgdc.gov

Federal Geographic Data Committee (2006c) Content standard for digital geospatial metadata. http://www.fgdc.gov/standards/projects/FGDC-standards-projects/metadata/base-metadata/index_html

Idaho Geospatial Data Clearinghouse, Interactive Numeric & Spatial Information Data Engine (2006). http://inside.uidaho.edu/whatsnew/whatsnew.htm

International Organization for Standardization (2007). http://www.isotc211.org/

Metadata Education Project, Wyoming Geographic Information Science Center (2006). http://www.sdvc.uwyo.edu/metadata/education.html

Open Geospatial Consortium, Inc. (2007) "About OGC". http://www.opengeospatial.org/ogc

United States Geologic Survey, Formal metadata (2006) Information and software. http://geology.usgs.gov/tools/metadata/

United States Geologic Survey, Content Metadata Standards for Marine Science: A case study, USGS open-file report 2004-1002. http://pubs.usgs.gov/of/2004/1002/html/evol.html

Wilson R (2004) What is metadata? TNGIC metadata outreach trainers and power point modules. http://www.tnmetadata.org/training.html

Recommended Reading

Federal Geographic Data Committee (2000) Content standard for digital geospatial metadata workbook. http://www.fgdc.gov/metadata/metadata-publications-list

Federal Geographic Data Committee (2005a) Geospatial metadata quick guide. http://www.fgdc.gov/metadata/metadata-publications-list

Federal Geographic Data Committee (2005b) Geospatial metadata factsheet. http://www.fgdc.gov/metadata/metadata-publications-list

Federal Geographic Data Committee and the National Metadata Cadre (2006) Top ten metadata errors. http://www.fgdc.gov/metadata/metadata-publications-list

Geospatial One-Stop (2004) Creating and publishing in support of geospatial one-stop and the national spatial data infrastructure. http://www.fgdc.gov/metadata/metadata-publications-list

Kang-tsung Chang (2008) Introduction to geographic information systems, vol 101, 4th edn.

Metadata Reference Guide for Federal Geographic Data Committee Metadata, Metadata Advisory Group, Massachusetts Institute of Technology Libraries (2004). http://libraries.mit.edu/guides/subjects/metadata/standards/fgdc.html

Wayne L (2005a) Federal Geographic Data Committee. Institutionalize metadata before it institutionalizes you. http://www.fgdc.gov/metadata/metadata-publications-list

Wayne L (2005b) Metadata in action: expanding the utility of geospatial metadata. http://www.fgdc.gov/metadata/metadata-publications-list

Wilson R (2004) Tennessee geographic information council, introduction to metadata

Metadata and Interoperability, Geospatial

David M. Danko
ESRI, Vienna, VA, USA

Synonyms

Catalog entry; Interoperability; Interoperability, technical; Interoperability, XML schema; ISO 19115; Legend; Marginalia; Semantic; Summary information; Supplementary material; Standards

Definition

Geospatial metadata is metadata about spatial information concerning objects or phenomena that are directly or indirectly associated with a location relative to the Earth; auxiliary information which provides a better understanding and utilization of spatial information. Metadata is a primary interoperability enabler.

ISO 19115 Geographic Information – Metadata defines metadata as "*data about data*." The Techweb Encyclopedia http://www.techweb.com/encyclopedia defines "data" as "*any form of information whether on paper or in electronic form. Data may refer to any electronic file no matter what the format: database data, text, images, audio and video. Everything read and written by the computer can be considered data except for instructions in a program that are executed (software).*"

Therefore, metadata is data/information in any form, paper or electronic, about data/information in any form, including computer/web service applications, no matter what format.

Historical Background

Interoperability has helped humans advance to a position as the dominate species in the world today. Interoperability becomes more complex and important as the world becomes more integrated and cultures become more interdependent. Two important forms of interoperability are technical and semantic interoperability. Geospatially humans have attained interoperability over the centuries using maps, charts, and in written and verbal descriptions. The need for geospatial interoperability is increasing as geographic information systems move into mainstream information technology (IT) applications and with the increased use of web services. There are many factors that are required to make interoperability happen; two major factors are standards and metadata. Standards: criteria which document agreement between a provider and a consumer; enable both technical and semantic interoperability. In the past standards for geographic information included those for languages, of course, and standards for consistency of scale, level of detail, geometric layout, symbology, and accuracy. With the exception of the aeronautical and hydrographic navigation fields these typically have been set by the national and commercial organizations producing the maps a charts. Metadata has always played an important role in cartography; for centuries it has provided users with an understanding of maps. Mention the word "metadata" and many think of something complex that applies only to information technology and computer science. However, metadata is not new; it is used every day in library card catalogs, Compact Disc (CD) jackets, user's manuals, and in many other ways. The field of cartography has a long history using metadata; it has been used for centuries in the margins of maps and charts. The title, source, scale, accuracy, producer, symbols, navigation notices, warnings, all of the information found in the borders of maps and charts is metadata. This metadata is very user oriented; just about anyone can pick up a map, understand the metadata, and use the map.

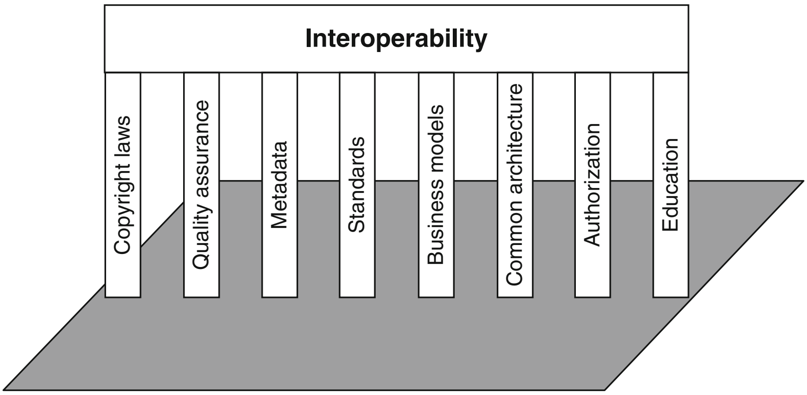
Geographic information systems (GIS) have always required interoperability. GIS uses data from multiple sources and from multiple distributed organizations within a community. For years GIS has been merging different information types: raster, vector, text, and tables. As the use of GIS grows and moves into varied disciplines the need for interoperability increases; GIS interoperates with a broad array of IT applications and is applied across diverse information communities. Web Services carry this need to new heights with loosely coupled, distributed networks.

Moving into the digital environment, metadata is equally important. Because digital data is an imperfect representation of the real world, and with the proliferation of data from an ever-widening array of sources and producers, it is important to have knowledge provided by metadata to understand, control and manage geographic information. Metadata adhering to international standards will expand interoperability enabling global networks, provide a common global understanding of geographic data, and promote global interoperability.

Moving into the world of global spatial data infrastructures, the need for internationally standardized metadata across communities was realized. In 2003 *ISO 19115 Geographic Information – Metadata* was established as an international standard it defines and standardizes a comprehensive set of metadata elements and their characteristics, along with the schema necessary to fully, and extensively, document geographic data. The standard applies to all types of geographic data and services. Since the development of the ISO metadata standard and with the wide expansion in the number of datasets and services available in the world the need for metadata focused on discovery became apparent resulting in the development of profiles of the ISO standard which support on-line catalogs, clearinghouses, and web portals.

Scientific Fundamentals

There are many things that are needed to make interoperability happen. It is necessary to have an infrastructure to support interoperability, a common architecture, and compatible technologies. Authorization (both authorization to share data and services with others, and authorization to uses other's data and services) is crucial. Ensuring that individual's and organization's intellectual property rights are not infringed is essential; therefore good copyright laws are needed. Also needed are business agreements and a business model; there must be a mutual benefit to both sides or there is no need to exchange information, no need for interoperability. Of course quality assurance helps; if the information in an exchange is not fit for purpose then there is no reason to be interoperable. **Standards** are required; standards allow us to communicate both technically – hardware and software working together; and semantically – understanding the same term for the same concept. The International Organization for Standardization Technical Committee for Geographic Information Standards (ISO/TC211) is developing an integrated suite of standards to address both technical and semantic interoperability. Of course, first and foremost is the understanding of data and services; for true interoperability, **metadata** is needed (Fig. 1). Metadata is an important part of the ISO TC 211 standards. Metadata provides a vehicle to locate and understand geospatial data which may be produced by one community and applied by another. As humans move into the age of global spatial data infrastructures, knowledge about widely distributed and dissimilar geographic data is essential to universally allow users to locate, evaluate, extract, and employ the data. Varied and wide spread



Concepts supporting interoperability

Metadata and Interoperability, Geospatial, Fig. 1

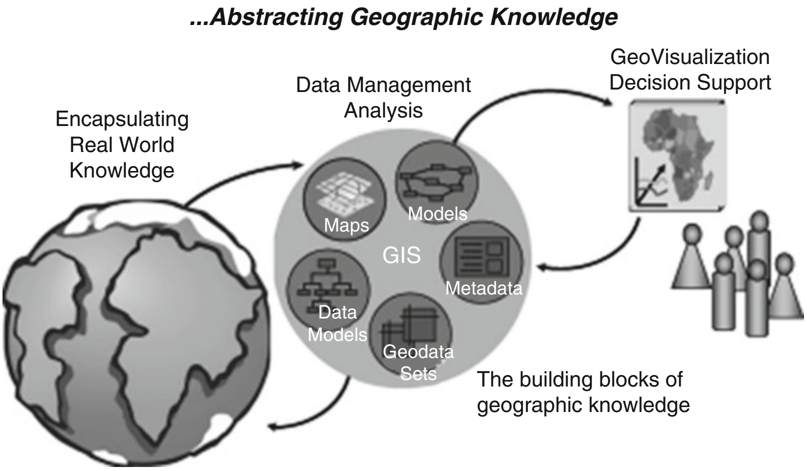
communities with a common understanding of metadata will be able to manage, share, and reuse each other's geographic data, making global interoperability a reality. An international metadata standard provides this common understanding worldwide. Metadata standards provide pick-lists of metadata elements so that producers will know what metadata to collect and users will know what metadata to look for. The pick-list includes a vocabulary fully defining the metadata elements so that producers and users around the globe can understand the metadata.

The ISO 19115 Metadata standard defines and standardizes a comprehensive set of metadata elements and their characteristics, along with the schema necessary to fully, and extensively, document geographic data. The standard applies to all geographic data – it is applicable to datasets in series, datasets, individual geographic features, and their attributes. The standard defines the minimum set of metadata required to serve the wide range of metadata applications, as well as optional metadata elements to support a more extensive description of geographic data. Because of the diversity of geographic data, no single set of metadata elements will satisfy all requirements; for this reason the ISO metadata standard provides a standardized way for users to extend their metadata and still ensure interoperability allowing other users to comprehend and exploit this extended metadata.

Many geographic metadata standards have been in existence prior to the development of this ISO standard. In many cases these separate information community, regional, and national standards evolved in separate niches and are incompatible. Several general metadata standards that do provide minimal global interoperability do not adequately support geographic information. This incompatibility and insufficiency was the motivation for the development of ISO 19115. The ISO metadata standard was designed:

- to support geographic information;
- to work with wider information technology standards and practices;
- to serve the global community, in a multi-national, multi-language environment;
- based on a foundation of national, regional, and special information community standards and experiences and a thorough requirements analysis, and implementation testing.

Geographic information systems encapsulate real world geographic knowledge into an information system environment by abstracting geographic knowledge into five basic building blocks allowing manipulation, data management, and analysis to support geo-visualization and decision making. These five building block element are: *data models, geodata sets, processes and workflows, maps and globes, and metadata*. *Data models* use spatial schemas, methods for defining/encapsulating geometry, topology, and networks, typically using a standardized modeling language or following rules for application schemas, to produce a template defining the relationships, rules, object definitions, and behavior of an abstraction of a universe of discourse for a specific user's conceptual view of geographic reality. *Geodata sets* are an instantiation of these models with digital data, typically in raster or vector form. Not all geographic phenomena can be abstracted using data models some are the result of a process and must be modeled using *process and workflow models*. *Maps and globes* are of course the oldest form of abstracting real world geography through the graphical display of geometry, topology, and attribution on paper, computer monitors, physical and virtual globes, and other display technology. All other aspects of real world geographic knowledge that cannot be modeled using the above four elements must be described using *metadata* (Fig. 2). Any description or abstraction of reality is always partial and always just one of many possible "views." This view, or model, of the real world is not an exact duplication; some things are approximated, others are simplified, and some things are ignored – there is no such thing as perfect, complete, and correct data. To insure that data is not misused, the assumptions and limitations affecting the collection of the


...Abstracting Geographic Knowledge
Encapsulating Real World Knowledge
Data Management Analysis
GeoVisualization Decision Support
Maps
Models
GIS
Metadata
Data Models
Geodata Sets
The building blocks of geographic knowledge

Metadata and Interoperability, Geospatial, Fig. 2

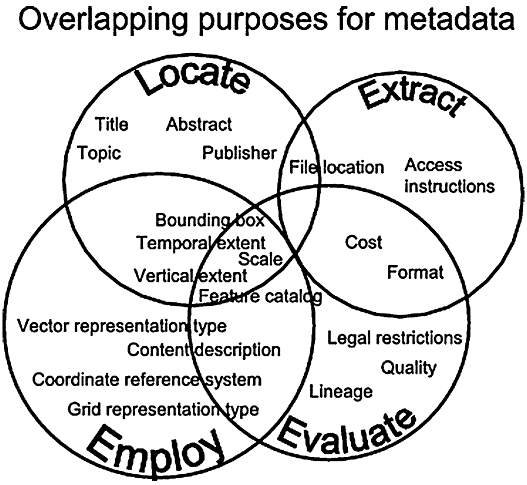
data must be fully documented. Metadata allows a producer to fully describe their geospatial data; users can understand the assumptions and limitations and evaluate the dataset's applicability for their intended use.

Metadata serves four purposes:

Locate: Metadata enables users to locate geospatial information and allows producers to "advertise" their data or service. Metadata helps organizations locate data outside their organization and find partners to share in data collection and maintenance.

Evaluate: By having proper metadata elements describing a dataset or service, users are able to determine if it will be suitable for their intended use. Understanding the quality and accuracy, the spatial and temporal schema, the content, and the spatial reference system used, allows users to determine if a dataset fills their needs. Metadata also provides the size, format, distribution media, price, and restrictions on use, which are also evaluation factors.

Extract: After locating a dataset and determining if it meets user's needs, metadata is used to describe how to access a dataset and transfer it to a specific site. Once it has been transferred, users need to know how to process and interpret the data and incorporate it into their holdings.



Metadata and Interoperability, Geospatial, Fig. 3

Employ: Metadata is needed to support the processing and the application of a dataset. Metadata facilitates proper utilization of data, allowing users to merge and combine data with their own, apply it properly, and have a full understanding of its properties and limitations (Fig. 3).

Metadata must be collected on all products (geospatial and non-geospatial) and should be produced – when knowledge of the products is fully understood – at the time of data production.

Key Applications

Metadata is required in at least four different circumstances and perhaps in different forms to facilitate its use: in a catalog for data discovery purposes; embedded within a dataset for direct use by application software; in a historical archive; and in a human readable form to allow users to understand and get a "feel" for the data they are using.

Catalogs: Metadata for cataloging purposes should be in a form not unlike a library card catalog or on-line catalog. Metadata in a catalog should support searches by subject matter/theme, area coverage/location, author/producer, detail/resolution/scale, currency/date, data structure/form, and physical form/media.

Historical Records: Metadata should support the documentation of data holdings to facilitate storage, updates, production management, and maintenance of geospatial data. Historical records provide legal documentation to protect an organization if conflicts arise over the use or misuse of geospatial data.

Within a geospatial dataset: Metadata should accompany a dataset and be in a form to support the proper application of geospatial data. GIS and other application software using data need to evaluate data as it applies to a situation. In this form the metadata may be incorporated into the structure of the data itself.

In a human readable form: Metadata in a form in which a computer can locate, sort, and automatically process geospatial data greatly enhance its use, but eventually a human must understand the data. One person's, or organization's, geospatial data is a subjective abstract view of the real world, it must be understood by others to ensure the data is used correctly. Metadata needs to be in a form which can be readily and thoroughly understood by users.

Non-geographers using geospatial data: A revival in the awareness of the importance of geography and how things relate spatially, combined with the advancement in the use of electronic technology, have caused an expansion in the use of digital geospatial information and geographic information systems (GIS) worldwide. Increasingly, individuals from a wide range of disciplines outside of the geographic sciences and information technologies are capable of producing, enhancing, and modifying digital geospatial information. As the number, complexity, and diversity of geospatial datasets grow, the use of metadata providing an understanding of all aspects of this data grows in importance.

Increasingly, the producer is not the user: Most geospatial data is used multiple times, perhaps by more than one person. Typically, it is produced by one individual or organization and used by another. Proper documentation provides those not involved with data production with a better understanding of the data and enable them to use it properly. As geospatial data producers and users handle more and more data, proper metadata documentation provides them with a keener knowledge of their holdings and allows them to better manage data production, storage, updating, and reuse.

Future Directions

As stated above, metadata is a primary interoperability enabler by making it possible for geospatial information users to better understand their information. Although metadata in the past played a key role in the production and use of paper maps and navigation charts, with the advent of the digital age, metadata has often been overlooked. People have been more concerned with the process of encapsulating real world knowledge into an information system. Now that this process of modeling one's "universe of discourse" has become routine and easily achievable using geographic information systems and metadata standards have been produced which guide producers and users on the importance, the definition, the concepts, and the utilization of geospatial metadata, it is now increasingly being produced and utilized. With the standardization of metadata many GIS and data collection systems are developing tools that automate metadata collection and management. The scale, coordinate reference system, language, character-set, keywords, data dictionary, and other information available in the data or the information system can be automatically collated into a metadata file.

With the recent development of *ISO/TS 19139 Metadata – XML Schema* technical interoperability has been achieved, enabling the exchange and machine parsing of metadata. ISO/TS 19139 also provides the capability to furnish multi-lingual metadata and enables the use of pre-defined code-lists standardizing information fields with vocabularies tailored for specific cultures or disciplines.

Since the development of an international geospatial metadata standard nations, regions, and scientific, defense, and commercial disciplines have been establishing profiles of this standard. Profiles allow information communities to tailor the standard to be "tuned" to meet the needs of a specific society or discipline. Profiles tailored for language, culture and specific vocabularies are being developed for Europe, North America, Latin America, and for information communities such as defense, environment, biology, navigation communities, and others. This development of profiles of ISO 19115 will continue – refining the understanding of metadata and the data it is describing – increasing interoperability across the globe and across information communities.

In the past, geospatial datasets – typically a full range of themes and/or feature types – were collected from a common source, scanning maps or data extraction from a single mono image or stereo model; metadata covering a specific dataset was fully adequate. Increasingly, now and in the future, geographic information is being gathered from a wide variety of sources and geospatial datasets and data bases are being incrementally updated necessitating the use of feature level and hierarchical levels of metadata; metadata about specific datasets as well as metadata about specific features within the datasets. As database and storage techniques improve feature level metadata will become more common.

Cross-References

▶ Metadata
▶ OGC's Open Standards for Geospatial Interoperability

Recommended Reading

Chan LM, Zeng ML (2006) Metadata interoperability and standardization – a study of methodology part 1. D-Lib Mag 12(6). ISSN 1082-9873. http://www.dlib.org/dlib/june06/chan/06chan.html

Dangermond J (2004) Speaking the language of geography – GIS, ArcNews. Fall. http://www.esri.com/news/arcnews/fall04articles/speaking-the-language1of2.html

Federal Geographic Data Committee (FGDC), Geospatial metadata. http://www.fgdc.gov/metadata

Longhorn R (2005) Geospatial standards, interoperability, metadata semantics and spatial data infrastructure, background paper for NIEeS workshop on activating metadata, Cambridge, 6–7 July 2005. http://archive.niees.ac.uk/talks/activating_metadata/Standards_Overview_and_Semantics.pdf

Moellering H (ed) (2005) World spatial metadata standards. Elsevier, Oxford

National Aeronautics and Space Administration Geospatial Interoperability Office (2005) Geospatial interoperability return on investment study, Greenbelt. http://gio.gsfc.nasa.gov/docs/ROI%20Study.pdf

National Information Standards Organization (2004) Understanding metadata. NISO Press, Bethesda. ISBN: 1-880124-62-9. http://www.niso.org/standards/resources/UnderstandingMetadata.pdf

Methods of Photogrammetry

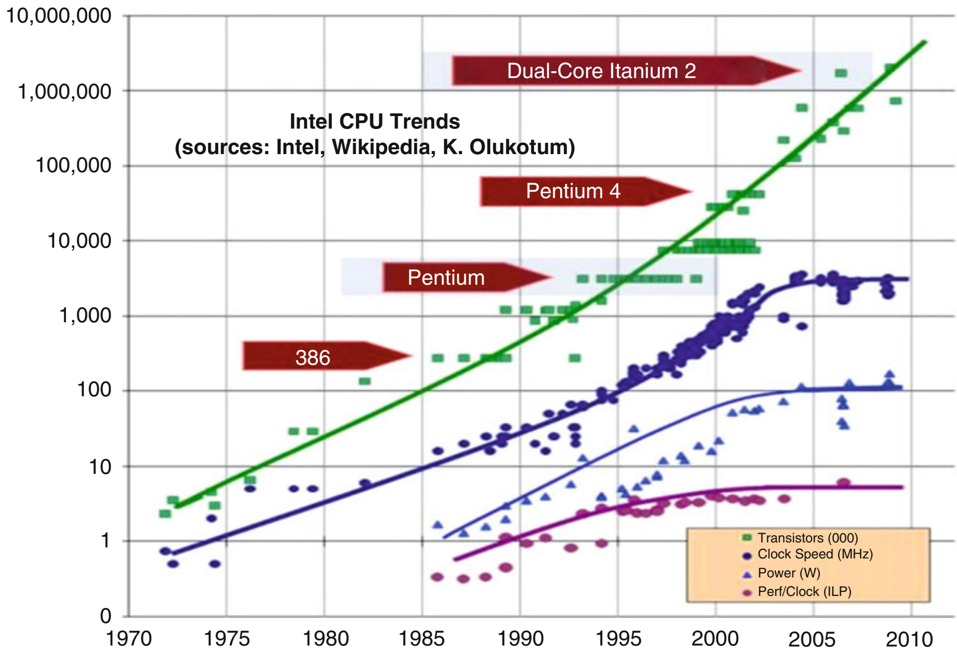
▶ Photogrammetric Methods

MIC in GIS

Xuan Shi[1] and Miaoqing Huang[2]
[1]Department of Geosciences, University of Arkansas, Fayetteville, AR, USA
[2]Department of CSCE, University of Arkansas, Fayetteville, AR, USA

Historical Background

Computer hardware architecture and technology have been changing rapidly in the past few



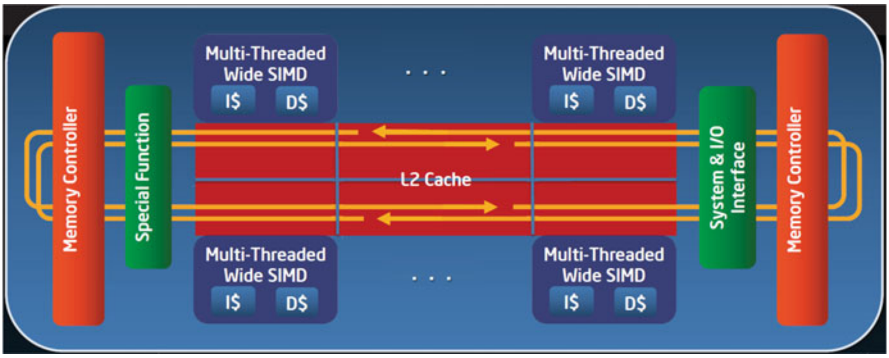
MIC in GIS, Fig. 1 CPU scaling showing transistor density, power consumption, and efficiency (Sutter 2005)

decades. In 1965, Gordon E. Moore, the cofounder of Intel Corporation, observed the trend that the number of transistors in a dense integrated circuit doubles approximately every 2 years (Moore 1965). It is envisioned that the overall computing power of a computer, or the clock speed of the processor in the central processing unit (CPU), would double every 2 years, since the increasing number of transistors could promote better performance in computation. Such a trend sustains for several decades till mid-2000s as described in Fig. 1.

Due to the physical barrier, however, it could be remarkably difficult to achieve significant performance improvements by further increasing the clock frequency on the uniprocessors. Clock frequency of the chip is the number of clock cycles repeated per second. Due to the heat dissipation and power consumption, when the clock frequencies reach around 4 GHz, which means the computer completes four billion cycles per second, it is difficult to achieve higher clock frequencies. At 90 nm level, the transistor gates became already too thin to prevent current from leaking out into the substrate (Geppert 2002; Borkar et al. 2011).

When the clock frequency on a single core had met such a physical bottleneck, multicore processors, or chip multiprocessors, have been developed to improve the overall system performance. Multicore processors contain two or more cores on one chip. In 2005, Intel and AMD released the dual-core processors into the market. Since then, quad-core processors were released in 2007, while octo-core processors were released in 2009. While multicore technology has been evolving incrementally, in 2012, Intel announced that Xeon Phi would be the brand name for all products based on the Many Integrated Core (MIC) architecture. Details about MIC are introduced in the following section.

A few pioneering works on geocomputation have been accomplished (Lai et al. 2014; Shi et al. 2014) over the Intel MIC-based supercomputer Beacon hosted at the National Institute for Computational Sciences (NICS) in the University of Tennessee. Three representative geocomputation applications were implemented on Beacon (Shi et al. 2014), including (1) Kriging interpolation, which is a use case of embarrassingly parallel computing, (2) ISODATA for unsupervised image classification,



MIC in GIS, Fig. 2 Internal architecture of Intel's Xeon Phi coprocessor

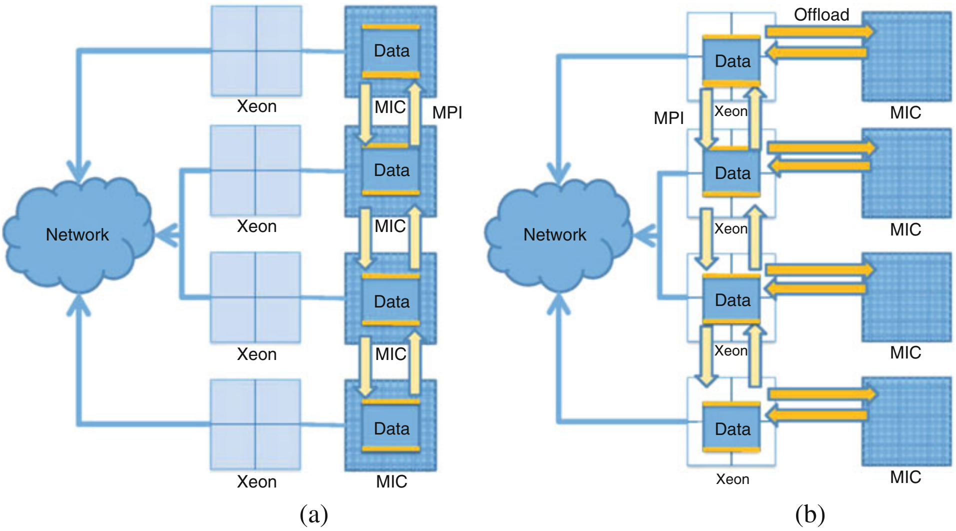
which has loose communication between distributed computer nodes for global reduction, and (3) Game of life, which is a cellular automata-based simulation and has intensive communication between the distributed computer nodes for both neighborhood-based calculation and global reduction.

Scientific Fundamentals

Xeon Phi is the first commercially available hardware product based on Intel's Many Integrated Core (MIC) architecture. Multicore CPUs, such as Intel Xeon *processors*, typically coexist with Xeon Phi *coprocessors* in a hybrid computer node. The current Intel Xeon Phi *coprocessor* contains up to 61 scalar processing cores. Each core on the coprocessor can run four threads in parallel. These cores are connected through a high-speed bidirectional, 1024-bit-wide ring bus (512 bits in each direction). In addition to the scalar unit inside each core, there is a vector processing unit to support wide vector processing operations. As shown in Fig. 2, each core has its own private L1 instruction cache and L1 data cache. Cache coherence among L1 caches is supported by hardware. There is also an onboard L2 cache shared by all the cores on the MIC card. This cache architecture is close to the traditional multicore CPU and, however, is quite different from the cache architecture on graphics processing units (GPUs). On GPUs, there is no direct communication between streaming multiprocessors. Therefore, cache coherence is not supported. On a MIC card, there is typically an off-chip global memory shared by all the cores in the same card. This global memory is separate from the main memory on the host. Therefore, data needs to be explicitly transferred to the global memory on the MIC card for efficient data processing.

Because each core alone is a classic processor, traditional parallel programming models, such as MPI and OpenMP, are supported by each core. The communications between the cores can be realized through the shared memory programming models, e.g., OpenMP. Additionally, each core can run MPI to realize communication. When multiple MIC cards are integrated into a cluster, direct communication between MIC processors across different nodes in the cluster is also supported through MPI.

When massively computing resources are available, different approaches can be deployed to parallelize scientific computation on computer clusters equipped with MIC processors. Figure 3 displays the two most commonly used approaches. In the native model, the MPI process is directly run on each MIC. Each MIC directly hosts one MPI process. In this way, the 60 cores on the Xeon Phi 5110P are treated as 60 independent processors while sharing the 8 GB onboard memory. To take advantage of the parallelism on each MIC, the multithreading approach can run four threads in each MPI process using OpenMP. In the offload mode, the MPI processes will be hosted by CPUs, which



MIC in GIS, Fig. 3 Two basic parallel approaches on MIC clusters. (**a**) Native model. (**b**) Offload model

will offload the data and computation to the MIC coprocessors using OpenMP. Besides the native model and the offload model, a third model, i.e., hybrid model, is also supported. In a hybrid MPI processing model, data and computation can be distributed by MPI onto both the CPU cores on Xeon processors and the MICs on the Xeon Phi coprocessors.

Three representative geocomputation applications, i.e., Kriging interpolation, ISODATA, and Game of life, have been accomplished (Lai et al. 2014; Shi et al. 2014) over supercomputer Beacon. Beacon is a Cray CS300-AC cluster supercomputer that offers access to 48 compute nodes and 6 I/O nodes joined by FDR InfiniBand interconnect providing 56 Gb/s of bidirectional bandwidth. Each compute node is equipped with two Intel Xeon E5-2670 8-core 2.6 GHz processors, four Intel Xeon Phi (MIC) coprocessors 5110P, 256 GB of RAM, and 960 GB of SSD storage. Each I/O node provides access to an additional 4.8 TB of SSD storage. Each Xeon Phi 5110P coprocessor contains 60 1.053 GHz MICs and 8 GB GDDR5 onboard memory. Thus, Beacon provides 768 conventional cores and 11,520 accelerator cores that provide over 210 TFLOP/s of combined computational performance, 12 TB of system memory, 1.5 TB of coprocessor memory, and over 73 TB of SSD storage, in aggregate. The compiler used in this work is Intel 64 Compiler XE, Version 14.0.0.080 Build 20130728.

Kriging is a geostatistical estimator that infers the value of a random field at an unobserved location. Kriging is based on the idea that the value at an unknown point should be the average of the known values at its neighbors. The algorithm itself reads input data and returns a raster grid with calculated estimations for each cell. No matter using the native model or the offload model, MPI is used for distributing the computation among computer nodes. For Kriging, the output raster grid is evenly distributed among multiple MPI processes. Each MPI process also receives the whole input data. Then each MPI process works on its own sub-grid by interpolating the value of unknown points using their neighbors.

The Iterative Self-Organizing Data Analysis Technique (ISODATA) algorithm is one of the most frequently used algorithms for unsupervised image classification in remote sensing applications (Ball and Hall 1965). The objective of this benchmark is to classify the image into n classes. In general, ISODATA can be implemented in three steps: (1) calculate the initial mean value of each class; (2) classify each pixel to the nearest class; and (3) calculate the new class means based on all pixels in one class. The second and third steps are repeated until the change between two iterations is small enough. In order to parallelize the computation, the whole image is partitioned into blocks of the same size. Each block is sent to a different MPI process. During each iteration, each MPI process first calculates the local means of n classes. Then all MPI processes send their local means to the head MPI process. After the head MPI process collects all the local means, it calculates the global means for the n classes and returns them to all other MPI processes. Then all the MPI processes start the computation of the next iteration.

Cellular automata (CA) are the foundation for geospatial modeling and simulation. Game of Life (GOL) (Gardner 1970), invented by British mathematician John Conway, is a well-known generic cellular automaton that consists of a collection of cells that can live, die, or multiply based on a few mathematical rules. The universe of the Game of Life is a two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive ("1") or dead ("0"). Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. In each iteration, the statuses of all cells are updated simultaneously. In order to parallelize the updating process, the cells in the square grid are partitioned into stripes along the row-wise order. Each stripe is handled by one MPI process. At the beginning of each iteration, each MPI process needs to send the statuses of the cells along the boundaries of each stripe to its neighbor MPI processes and receive the statuses of the cells of two adjacent rows.

On supercomputer Beacon, MPI was used for distributing workloads among allocated MICs. On Intel MIC coprocessors, the native model, i.e., MPI processes directly run on MICs, was first applied. For all three representative benchmarks, the strong scalability of Beacon cluster was examined until the performance reached a plateau due to the introduced communication overhead. Furthermore, a pilot study was conducted to compare different programming models on the Intel MIC coprocessor with detailed report on the scalability and performance comparison (Lai et al. 2014). In summary, when the native model was applied, OpenMP was further used to increase the number of threads running on each MIC. On the offload model in which the MPI process runs on the host CPU, various numbers of threads were deployed to the MIC coprocessor for performance comparison. Further a third programming model was implemented, i.e., the hybrid model, in which the workloads are scheduled onto both the CPUs and the MICs. Experiments demonstrated that the native model is typically better than the offload model. The hybrid model can provide extra performance improvement.

Key Applications

Jeffers and Reinders concluded (2013) that "Most applications in the world have not been structured to exploit parallelism. This leaves a wealth of capabilities untapped on nearly every computer system." In the research and development of geographic information system (GIS) and science (GIScience), the scalability and performance of geospatial computation are severely limited when massive datasets are processed by serial program over desktop computer or a single processor. Heterogeneous geospatial data integration and analytics obviously magnify the complexity and operational time frame. Many large-scale geospatial problems may not be processible at all if the computer system does not have sufficient memory or computational power.

Emerging computer architectures and advanced computing technologies, such as Intel's Many Integrated Core (MIC) architecture, provide a promising solution to employ massive parallelism to achieve scalability with high performance for data-intensive computing over large spatial data. From a perspective of efficiently utilizing the MIC architecture and different approaches for parallelism on MICs, geocomputation can be classified into two general categories depending on whether data communication has to be implemented over MIC processors. Embarrassingly parallel approach is the first category since geocomputation is implemented on MIC processors without communication between nodes. In spatial data processing and analytics, many GIS functions can be done through embarrassing parallelism, such as map algebra calculation over raster datasets, spatial interpolation by IDW and Kriging (Shi and Ye 2013), data conversion from one data type or projection to other types or projections that can be another use case (Li et al. 2010), and many more.

In the second category, data exchange and reduction are necessary in distributed computation. Handling data communication between MICs increases the difficulty and challenge to complete geocomputation tasks, such as neighborhood-based calculation in focal and zonal statistics, surface analysis, and aspect or slope calculation. In such scenarios, data at the boundary of the tiles has to be exchanged before the geospatial computation is executed. In spatial statistic calculation, reduction has to be implemented over distributed MIC processors. In spatial modeling and simulation, data exchange and reduction may have to be executed multiple times in order to complete the simulation. While data exchange is performed by regular forms if the same amount of data, such as a given number of columns or rows, is exchanged between multiple MICs, irregular data exchange will increase the difficulty in handling load balance.

In summary, many geocomputation problems can be solved by running the GIS operations on MIC or clusters of MICs, such as those three categories of geocomputation discussed in scientific fundamentals. In practice, however, several issues may have significant impact on expanding the usage of MICs in the geospatial applications and software engineering. First of all, each MIC has 60 independent processors while sharing 8 GB onboard memory. For this reason, each individual processor is weak and has limited memory. When large-scale geospatial data are involved in the computational tasks, a single MIC may not be able to process the data or to complete the task efficiently. Consequently a cluster of MICs has to be utilized. If the price of MIC is too high, it may not be affordable to general users. Considering most software products are designed and developed for desktop users, if such software products only use a single processor to do something, MIC may not be a cost-effective equipment since most of those 60 independent processors on MIC are not utilized efficiently. Finally, software redesign and reengineering are required to implement the abovementioned native, offload, or hybrid programming modes to deploy MIC in GIS software or application development. Commercial GIS software companies may not have the intention for investment if MIC is not the dominant computer equipment in the major market and user community.

Future Directions

Hardware aspect: As the size of the basic component of silicon devices, i.e., the transistor, keeps shrinking, chip vendors are able to accommodate more and more processing cores on a single processor. It can be envisioned that future many-core processors will contain hundreds to thousands of cores, which will provide a massive parallelism to support many parallel applications. Another trend is the union of the host CPU and the coprocessor into a single hybrid device. Currently, the Intel MIC coprocessor is designed as a separate device with its own memory hierarchy in its own package. Because the memory space of the coprocessor is separated from the main memory space of the host CPU, data have to be transferred to the MIC side for efficient data processing. In the near future, both the multicore CPU and the many-core coprocessor may be integrated into a single chip. Naturally these two memory hierarchies will be merged into a single space so that the workloads can be distributed on these two types of cores flexibly and seamlessly.

Software aspect: Currently, application development on Intel MIC coprocessors typically has to write computer programs to control and utilize the host CPUs and the coprocessors. These two parts of programs are written separately in most cases. Specific capability and skills are required to write the MPI and OpenMP code on the coprocessors. It is expected that new programming languages, such as OpenACC, and compilation techniques are proposed to express data parallelism more explicitly in the computer programs so that the compilers can generate parallel code more easily. This will reduce or remove the hurdles for domain scientists to deploy high-performance computing solutions and resources and to facilitate the process of porting domain applications to the massively parallel platforms.

Cross-References

► MPI in GIS

References

Ball GH, Hall DJ (1965) ISODATA: a method of data analysis and pattern classification. Technical report, Stanford Research Institute, Menlo Park

Borkar S, Chien AA (2011) The future of microprocessors. Commun ACM 54(5):67–77. doi:10.1145/1941487.1941507

Gardner M (1970) Mathematical games – the fantastic combinations of John Conway's new solitaire game of life. Sci Am 223:120–123

Geppert L (2002) The amazing vanishing transistor act. IEEE Spectrum 39(10):28–33

Jeffers J, Reinders J (2013) Intel Xeon Phi coprocessor high-performance programming. Morgan Kaufmann/Elsevier, Amsterdam/Boston

Lai C, Hao Z, Huang M, Shi X, You H (2014) Comparison of parallel programming models on Intel MIC computer cluster. In: Proceedings of fourth international workshop on accelerators and hybrid exascale systems (AsHES) as part of IPDPS, Phoenix, 19 May, pp 925–932

Li J, Humphrey M, Agarwal D, Jackson K, van Ingen C, Ryu Y (2010) eScience in the cloud: a MODIS satellite data reprojection and reduction pipeline in the Windows Azure platform. In: Processing of 2010 IEEE international parallel & distributed processing symposium (IPDPS), Atlanta

Moore GE (1965) Cramming more components onto integrated circuits. Electronics 38(8), pp 114-117

Shi X, Ye F (2013) Kriging interpolation over heterogeneous computer architectures and systems. GISci Remote Sens 50(2):196–211

Shi X, Lai C, Huang M, You H (2014) Geocomputation over the emerging heterogeneous computing infrastructure. Trans. GIS. doi:10.1111/tgis.12108

Sutter H (2005) The free lunch is over: a fundamental turn toward concurrency in software. http://www.gotw.ca/publications/concurrency-ddj.htm

Microgeomatics

► Indoor Positioning

Minimum Aggregate Travel Point

► Geometric Median

Minimum Bounding Rectangle

Jordan Wood
Department of Computer Science, University of Minnesota, Minneapolis, MN, USA

Synonyms

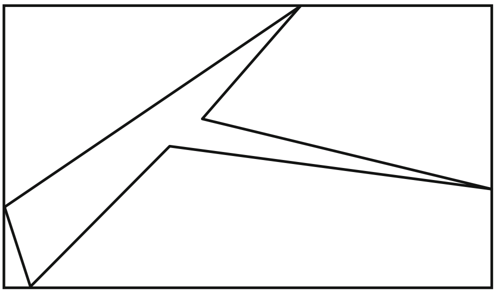
MBR; Minimum orthogonal bounding rectangle; MOBR

Definition

A minimum bounding rectangle is used to approximate a more complex shape. It is a rectangle whose sides are parallel to the x and y axises and minimally enclose the more complex shape.

Main Text

Spatial objects can take a significant amount of memory to represent. For example, a polygon which represents the borders of a country could have tens of thousands of vertices. A polyline which represents a complex linear feature such as a river would also have many vertices. Doing geometric operations such as finding objects which overlap such a complex object would be very computationally expensive, since the location of every vertex would have to be considered. There are times when we only need to know the approx-



Minimum Bounding Rectangle, Fig. 1

imate geometrical features of an object, such as during the filter step of a filter and refine strategy. In these cases, the minimum bounding rectangle (MBR) is used to approximate the shape in a simpler manner. The sides of an MBR are always parallel to the *x* and *y* axises of the space in question. Also, it is the smallest rectangle with this property which completely encloses the original shape. It can be calculated and stored as the minimum and maximum x and y values of the original shape. An example of an MBR is shown in Fig. 1. The rectangle is the MBR of the polygon.

Cross-References

- ▶ Plane Sweep Algorithm
- ▶ Spatial Constraint Databases, Indexing

Recommended Reading

Shekhar S, Chawla S (2003) Spatial databases: a tour. Pearson Education, Upper Saddle River. ISBN:0-13-017480-7

Minimum Bounding Rectangles

▶ Oracle Spatial, Geometries

Minimum Orthogonal Bounding Rectangle

▶ Minimum Bounding Rectangle

Mining Collocation Patterns

▶ Co-location Patterns, Algorithms

Mining Sequential Influence for Personalized Location Recommendations

▶ Exploiting Sequential Influence for Personalized Location-Based Recommendation Systems

Mining Sequential Patterns from Spatiotemporal Databases

▶ Sequential Patterns, Spatiotemporal

Mining Spatial Association Patterns

▶ Co-location Patterns, Algorithms

Mining Spatiotemporal Datasets

▶ Trajectories, Discovering Similar

Mitigation

▶ Climate Change and Developmental Economies

MLPQ Spatial Constraint Database System

Peter Z. Revesz
Department of Computer Science and
Engineering, University of Nebraska-Lincoln,
Lincoln, NE, USA

Synonyms

Management of linear programming queries; MLPQ system

Definition

The MLPQ system (Kanjamala et al. 1998; Revesz et al. 2000; Revesz and Li 1997; Revesz 2010) is a spatial constraint database system developed at the University of Nebraska-Lincoln. The name of the system is an acronym for *Management of Linear Programming Queries*. The special feature of the MLPQ system is that it can run both SQL and Datalog queries on spatial constraint databases with linear inequality constraints. The MLPQ system is applied in the area of geographic information systems, moving objects databases and operations research. The MLPQ system has an advanced graphical user interface that displays maps and animates changes over time. Using a library of special routines, MLPQ databases can be easily made web-accessible.

Scientific Fundamentals

The MLPQ system accepts input textfiles specified as a set of constraint tuples or facts. Each atomic constraint must written in a way that all variables are on the left hand side of the >= comparison operator meaning "greater than or equal." For example, the Lincoln town area can be represented as follows.

```
begin %Lincoln-area%
Lincoln (id, x, y, t) :- id = 1, y − x >= 8, y >= 14,
x >= 2, −y >= − 18,
−y − z >= −24.
Lincoln (id, x, y, t) :- id = 2, x − y >= −8, 0.5 x − y
>= −12,
−y >= −18, x >= −14, y >= 8, 3x + y >= 32.
end %Lincoln-area%
```

The MLPQ allows browsing of the directory of input data files. Once an input datafile is selected, then the system immediately displays it on the computer screen. For queries and advanced visualization tools a number of graphical icons are available. For example, when clicking on the icon that contains the letters *SQL* a dialog box will be called. The dialog box allows selection of the type of SQL query that the user would like to enter. The types available are basic, aggregate, nested, set, and recursive SQL queries. Suppose one clicks on *AGGREGATE*. The a new dialog box will prompt for all parts of an aggregate SQL query, including the part that creates an output file with a specific name as the result of executing the query. For example, one can enter either of the SQL queries that one saw in the entry on Spatial Constraint Databases. When the SQL query is executed and the output relation is created, it is shown on the left side of the computer screen. Clicking on the name of the output relation prompts MLPQ to display it as a spatial constraint database table and/or as a map if that is possible. MLPQ requires each constraint tuple to have an id as its first field (this helps internal storage and indexing); however, any number of spatiotemporal and non-spatiotemporal fields can follow the id field. If there are more than two spatial fields, then the projections to the first two spatial fields (assumed to be the second and third fields) are displayed only. To visualize moving objects with a growing or shrinking spatial area, special animation algorithms need to be called. The system is available free from the author's webpage at cse.unl.edu/~revesz.

Cross-References

▸ Constraint Database Queries
▸ Spatial Constraint Databases, Indexing
▸ Visualization of Spatial Constraint Databases

References

Kanjamala P, Revesz P, Wang Y (1998) MLPQ/GIS: a GIS using linear constraint databases. In: Proceedings of 9th COMAD international conference on management of data. McGraw-Hill, New Delhi, pp 389–393

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

Revesz P, Li Y (1997) MLPQ: a linear constraint database system with aggregate operators. In: Proceedings of 1st international database engineering and applications symposium. IEEE Press, Washington, pp 132–7

Revesz P, Chen R, Kanjamala P, Li Y, Liu Y, Wang Y (2000) The MLPQ/GIS constraint database system. In: Proceedings of ACM SIGMOD international conference on management of data. ACM Press, New York

MLPQ System

▶ MLPQ Spatial Constraint Database System

Mobile Advertising

▶ Mobile GIS Solutions for Retail and Advertising

Mobile Check-In Recommendation

Defu Lian[1] and Nicholas Jing Yuan[1,2]
[1]Big Data Research Center, University of Electronic Science and Technology of China, Chengdu, China
[2]Microsoft Research, Beijing, China

Synonyms

Point-of-Interest Recommendation; Venue Discovery; Venue Recommendation

Definition

Check-in service is a feature of location-based social networks, such as Foursquare, that is used for announcing a person's arrival at a point of interest with precise coordinates and rich semantic and content information. Due to the growing popularity of location-based social networks, a vast number of user check-ins have been accumulated. Based on this data, users' preferences can be learned and it is possible to predict or change future visiting locations for users. Mobile check-in recommendation is one such technique that places an emphasis on helping users to change their routines for discovering novel locations. Therefore, it is an important method for helping people to speed up their familiarization with their surroundings, especially when they arrive at new places. From the more technical perspective, it is a specific type of location recommendation, which includes a subclass of information filtering algorithms that seek to predict the "rating" or "preference" a user might give to a location. However, in contrast to classical location recommendation, mobile check-in recommendation targets recommending points of interest and mainly makes use of check-in data from location-based social networks. This check-in data is large scale yet noisy and spread all over the world. This data can also be shared with friends on social networks, such as Twitter and Weibo, and thus includes friend links and interest information.

Historical Background

Although mobile check-in recommendation has its roots in location recommendation, they have so many differences that it is difficult for the algorithms in classical location recommendation to be directly applied to mobile check-in recommendation. For example, using multiple users' real-world location histories, some recommender systems (Horozov et al. 2006; Zheng et al. 2009, 2010; Takeuchi and Sugimoto 2006; Park et al. 2007) have been designed to recommend geographic locations such as shops, restaurants, hot spots, or general locations to users. These real-world location histories are collected from a small number of volunteers and are located in a limited number of geographic regions. Moreover, although these histories may include lots of geographical coordinates, they show strong regularity and redundancy. These characteristics of real-world location histories are different from large-scale yet noisy and wide-ranging areas in mobile check-in history. In addition, check-ins are produced by users on location-based social networks, so they are accompanied by rich and diverse social network information. All of these features characterize the research of mobile check-in recommendation.

Scientific Fundamentals

Recommender Systems

Recommender systems help people to sift through a large number of movies (Miller et al. 2003; Bennett and Lanning 2007), music (Celma 2010), restaurants (Park et al. 2007), products (Linden et al. 2003), and so forth to find the most interesting and valuable information tailored for them. Generally, recommendation problems explicitly rely on a rating structure. In the most common formulation, it is reduced to a problem of estimating (extrapolating) ratings for unrated items of a user based on all the rated ones. If in the rating history there are M users $U = \{u_1, \cdots, u_M\}$ and N items $I = \{l_1, \cdots, l_N\}$, where each user and item can be associated with a set of characteristics, they can be organized as an $M \times N$ matrix, where question marks in a row show the unrated items of a user:

$$R = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_{M-1} \\ u_M \end{array} \begin{array}{c} \begin{array}{ccccc} l_1 & l_2 & \cdots & l_{N-1} & l_N \end{array} \\ \begin{pmatrix} r_{1,1} & ? & \cdots & r_{1,N-1} & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & ? & r_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \\ ? & r_{M-1,2} & \cdots & r_{M-1,N-1} & ? \\ r_{M,1} & ? & \cdots & r_{M,N-1} & r_{M,N} \end{pmatrix} \end{array}.$$

In this way, estimating ratings for unseen items is formulated as a matrix completion problem. Once we can estimate ratings for unrated items, we can recommend the items with the highest estimated ratings to the user. Based on how recommendations are made, recommender systems are usually classified into the following three categories. The first is content-based recommendation, recommending items whose content is similar to ones the users have preferred in the past. Thus it extrapolates the ratings for unrated items of a user based on the rated ones by means of content information. The second category is collaborative recommendation, which is based on collaborative filtering. It recommends items that people with similar tastes and preferences chose in the past. In other words, it extrapolates the ratings for unrated items of a user based on the rated ones of other people with similar appetites. And the last category is hybrid recommendation, which combines content-based recommendation and collaborative recommendation.

Mobile Check-In Recommender Systems

Mobile check-in recommender systems provide personalized locations for users according to their check-in history. Due to its importance in helping people become familiar with their surroundings, mobile check-in recommendation has become an important research topic since the appearance of the first paper on this topic (Ye et al. 2010, 2011). The key research point of check-in recommendation that is different from classical recommendation (such as movie, music, news, etc.) is the existence of geographical information of POIs (items) and social networks. In other words, mobile check-in recommendation mainly studies how to exploit this information to recommend points of interest. The characteristics of check-in history indicate only a part of collaborative filtering algorithms that are more suitable for check-in recommendation. Therefore, the research of mobile check-in recommendation is organized according to the following three aspects.

Geographical Modeling

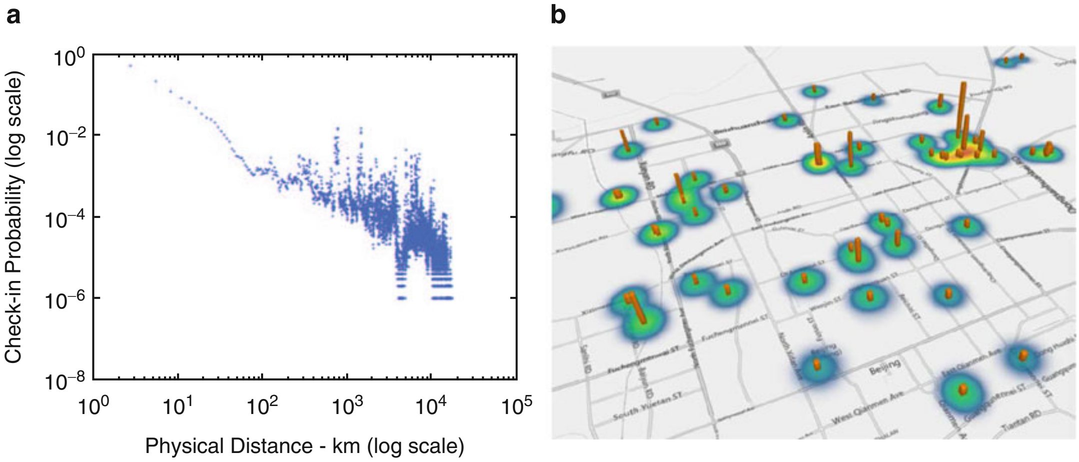
The geographic information of POIs requires physical interactions with users so that it fosters the study and modeling of Tobler's first law of geography: "Everything is related to everything else, but near things are more related than distant things." For example, Ye et al. (2011) observed a significant percentage of POI pairs checked in by the same user appeared to be within a short distance, as shown in Fig. 1a, and assumed the distance between pairs of POIs followed power-law distribution. In particular, the distance between any pair of POIs is first calculated, and then over these distances, a histogram is plotted. Based on the histogram, a power-law distribution is fitted, i.e.,

$$p(d_{j,k}) = a \times d_{j,k}^{b}$$

where $d_{j,k}$ is the distance between POI l_j and l_k and $p(d_{j,k})$ is the probability of checking in at these two POIs simultaneously. After taking a logarithm on both sides of the equation, it becomes a linear equation so that parameters a and b can be determined by the linear regression optimizer. When assuming the independence of the distances between different pairs of POIs (Ye et al. 2011), the probability a user i will check in at another POI l_j, given his historical check-in POIs L_i, is

$$P(l_j|L_i) = \prod_{l_k \in L_i} p(d_{j,k})$$

However, as observed by Zhang and Chow (2013), though all users' distance information between POI pairs in an aggregated level shows the power-law distribution, individual distribution of distance is varied from person to person. Therefore, they suggested using kernel density estimation to estimate the individual distribution of distances between POI pairs at which the same user checked in. In spite of this, concentrating on modeling the distance distribution may still ignore the multicenter characteristics of an individual visited location according to Cheng et al. (2012), as shown in Fig. 1b; thus, Cheng et al. (2012) and Liu et al. (2013) tried to apply clustering technology on individual visited locations to capture the first law of geography. The number of clusters in clustering technology is determined heuristically by cross validation. However, the number of geo-clusters is usually different from person to person. It is insufficiently appropriate to set the same number of geo-clusters for all users. Therefore, Lian et al. (2013, 2014) proposed leveraging a two-dimensional kernel density estimation on individual check-in POIs. Specifically, the probability of checking in at a POI l_j for a user u_i is determined by



Mobile Check-In Recommendation, Fig. 1 Geographic distribution illustration. (**a**) Power-law distributed distance on Foursquare. (**b**) The density plot of location dist. on Jiepang

$$P(l_j) = \frac{1}{|L_i|h} \sum_{l_k \in L_i} K(\frac{d_{j,k}}{h})$$

where $K(\cdot)$ is a kernel function. Based on this formulation, we see that the key difference from a power-law solution is that a power-law assumption slows down the decrease of check-in probability at some POI with the increase of its distances from all the check-in POIs of one user. More importantly, the benefit of leveraging kernel density estimation is reducing the complexity of computation by means of approximation techniques. The basic intuition is that the check-in probability of a user at a specific POI l_j is almost not affected by his/her distant check-in POIs. If one POI is distant from the nearest check-in POI of a user, the check-in probability at this POI is approaching zero so that its computation could be eliminated. For this reason, in Lian et al. (2013), a propagation-based algorithm is proposed to select candidate POIs and the check-in probability is then calculated. In particular, for each check-in POI l_j, its geographical influence is only propagated to nearby POIs within d km if it is assumed that the influence radius of POIs is d km. In other words, only these nearby POIs are affected by this POI. Then, each l_k of these nearby POIs receives $\frac{1}{h}K(\frac{d_{j,k}}{h})$ influence from the POI l_j. After performing propagation for all the check-in POIs of one user, the received geographical influence of one candidate POI from them is summed together and then divided by the number of them; the check-in probability at this candidate POI is obtained. During this process, the setting of bandwidth h in the kernel function is determined by the requirement that the influence of candidate locations on the border of the influence circle is close to zero. If the probability on the border is at most ϵ times smaller than the maximum possible check-in probability, it is subject to $K(\frac{d}{h}) < \epsilon K(0)$. The dominate running time of computing these probabilities is used to perform range query for retrieving nearby POIs within d km from each check-in POI. With the help of a range tree, it is linearly proportional to the logarithm of the total number of POIs and the number of retrieved POIs. Therefore, using this approximation technique can greatly reduce the time complexity. Actually, the improvement in time complexity can be further boosted by splitting the whole world into grids of approximately the same size and assuming the received geographical inference of POIs in the same grid is equal to each other, as introduced in Lian et al. (2014). Moreover, given the same influence radius of all POIs, when precomputing the received influence of each grid from all the POIs, two-dimensional kernel density estimation can be converted to the following optimization problem:

$$\min_{x_i} \sum_j \ell(x_i^T y_j, r_{i,j}) + \lambda\Omega(x_i)$$

$$\text{subject to } x_i \geq 0$$

where y_j is an influence vector of a POI l_j, and each element corresponds to a grid's influence received from this POI and x_i is an activity area vector of user u_i, in which every element corresponds to nonnegative possibility that this user will appear in a certain grid. Thus their dot product can be considered as the possibility that user u_i will check in at POI l_j. $\ell(\cdot,\cdot)$ is a loss function, measuring the deviation between $x_i^T y_j$ and $r_{i,j}$, where $r_{i,j}$ is a Boolean variable indicating whether user u_i has checked in at POI l_j or an integer variable indicating the times that user u_i has checked in at POI l_j. $\Omega(x_i)$ is a regularized term avoiding over-fitting. As long as $\ell(\cdot,\cdot)$ and $\Omega(x_i)$ are convex with respect to x_i, the optimizer can be obtained by convex optimization algorithms. The general loss function in recommendation is weighted squared loss:

$$\ell(x_i^T y_j, r_{i,j}) = w_{i,j}(x_i^T y_i - r_{i,j})^2$$

where $w_{i,j}$ is a weight of rating of user u_i on POI l_j. The details will be discussed bellow since it is directly related to the state-of-the-art collaborative filtering for mobile check-in recommendation.

Collaborative Filtering

Since geographical information can be considered to be the content of POIs, geographical modeling for recommendation is considered to be a content-based filtering method. Although it has played an important role in recommendation, it suffers from overspecialization (Adomavicius and Tuzhilin 2005) since it always recommends items of similar content and nearby POIs by considering geographical information as content. This problem can be alleviated by collaborative filtering, which tries to predict the utility of items for a particular user based on the items previously rated by other users. In the problem of check-in recommendation, a series of collaborative filtering algorithms have been exploited. For example, user-based collaborative filtering, involving collaboration between users, has been applied to mobile check-in recommendation (Ye et al. 2011). In particular, assuming $r_{i,j}$ is a Boolean variable indicating whether a user u_i has checked in at POI l_j, the possibility $f_i(l_j)$ of this user checking in at POI l_j can be measured as

$$f_i(l_j) = \frac{\sum_{u_l \in U} s_{i,l} \cdot r_{l,j}}{\sum_{u_l \in U} s_{i,l}}$$

where $s_{i,l}$ is the similarity weight between users u_i and u_l. The similarity weight between two users is related to the number of their common check-in locations and is commonly set as a Jaccard similarity coefficient or cosine similarity between their POI vectors, which are defined as

$$s_{i,l}^{\text{Jac}} = \frac{|L_i \cap L_l|}{|L_i \cup L_l|}, s_{i,l}^{\cos} = \frac{|L_i \cap L_l|}{\sqrt{|L_i||L_l|}}.$$

However, from these formulations, we can see that this way of modeling ignores the check-in frequency at different POIs. According to the study in Lian et al. (2013), check-in frequency plays an important part in check-in recommendation since it reflects the extent or confidence of user preference. A larger check-in frequency indicates a higher confidence in the users' preferences for POIs. The simple strategy for this problem is to replace the Boolean variable indicating check-in with frequency of check-in, which has been exploited in Noulas et al. (2012). Nevertheless, based on the analysis in Lian et al. (2013), such a strategy is not as good as the one above, i.e., using the Boolean value, since the recommendation list of POIs is dominated by regular locations, such as homes or office buildings, of similar users. For example, assuming that users A and B regularly visit their distinct residences and workplaces and A has also been to two restaurants x and y occasionally, while B has enjoyed lunches at restaurants x and z at times, recommending algorithms using check-in frequency tends to recommend the residence and workplace of user B other than the restaurant z to user A, which may be unacceptable for the latter one. In other words, check-in frequency information is not well exploited. Another point not taken into account in this approach is that each check-in may provide a favorable POI, whereas unchecked-in POIs with respect to a specific user may be either undiscovered or really unattractive to her. Thus, directly discarding unchecked-in POIs, just like in user-based collaborative filtering, may miss some important and useful information. These two points actually share the same characteristics as implicit feedback datasets, which only provide positive items for each user, in contrast to explicit feedback datasets with both positive items (high rating) and negative items (low rating). Thus, based on these two characteristics, in Lian et al. (2014), proposed applying weighted matrix factorization (Hu et al. 2008; Pan et al. 2008) tailored for implicit feedback for check-in recommendation and empirically showed the superiority of weighted matrix factorization to user-based collaborative filtering. Matrix factorization is a model-based collaborative filtering algorithm, which develops a model for estimating a user's ratings first and then makes a recommendation based on the estimated ratings. According to Adomavicius and Tuzhilin (2005), its recommendation performance, such as cutoff recall and precision, NDCG, MAP (Manning et al. 2008), and so on, is usually better than user-based collaborative filtering, which is a memory-based recommendation algorithm and uses the entire user-item (POI) data to generate a prediction.

Matrix factorization involves mapping users and POIs into a joint latent space with a dimension $K \ll \min(M, N)$, such that a user's preference for a POI is modeled as an inner product between them in the latent space. The mapping is achieved by the following optimization problem:

$$\min_{P,Q} \sum_{i,j} w_{i,j}(p_i^T q_j - r_{i,j})^2 + \lambda(\|P\|_F^2 + \|Q\|_F^2)$$

where $p_i \in \mathbb{R}^K$ and $q_i \in \mathbb{R}^K$ are latent factors of user u_i and POI l_j. P and Q are matrices whose rows are latent factors of users and POIs, respectively. $\|\cdot\|_F$ is Frobenius norm of matrices, simply the square root of sum of squared values in matrices, to avoid over-fitting. A common configuration of weight $w_{i,j}$ is set as 1 if user u_i has checked in at POI l_j and 0 otherwise, as used in Yang et al. (2013), Zheng et al. (2010), and Noulas et al. (2012). However, such a setting still doesn't use unchecked-in POIs of users so that it is difficult to discriminate the most favorable POIs from the least favorable ones. One effective strategy for this problem is to randomly sample unchecked-in POIs as negative samples first and then assigning lower weights to them than check-in POIs (Pan et al. 2008). Another effective and more efficient strategy for this problem is considering all unchecked-in POIs as negative samples and then assigning the same but lower weight to them than check-in POIs (Hu et al. 2008). One setting of weight $w_{i,j}$ is

$$w_{i,j} = \begin{cases} \alpha(c_{i,j}) + 1 & \text{if } c_{i,j} > 0 \\ 1 & \text{otherwise} \end{cases}$$

In this case, although loss function is defined over a large number of samples, it can be efficiently optimized by means of the alternative least square algorithm. This algorithm consists of a series of iterative steps of optimizing P given Q fixed and optimizing Q given P fixed. When fixing Q, the objective function decouples between different users. In other words, latent vectors of different users can be optimized separately and in parallel. For user u_i, the updated formulation of its latent vector is defined as follows:

$$p_i = \left(Q^T(W^i - I)Q + Q^T Q + \lambda I\right)^{-1}$$
$$Q^T W^i r_i \text{ , } i \in \{1, 2, \cdots, M\}$$

where W^i is an $N \times N$ diagonal matrix, subject to $W^i_{j,j} = w_{i,j}$, and r_i is a column rating vector of user i. Here the efficiency comes from an extremely sparse matrix $(W^i - I)$ and $Q^T Q$, which is shared across different users and thus can be precomputed before optimizing each user's latent factors. Similarly, the latent factor of POI l_j is updated as follows:

$$q_j = (P^T(W^j - I)P + P^T P + \lambda I)^{-1}$$
$$P^T W^j \tilde{r}_j \text{ , } i \in \{1, 2, \cdots, N\}$$

where W^j is an $M \times M$ diagonal matrix, subject to $W^j_{i,i} = w_{i,j}$, and $\tilde{r}_j$ is a column rating vector of the POI l_j. When setting each element of the rating matrix R as a Boolean value indicating whether a user has checked in at the corresponding POI, it achieves state-of-the-art mobile check-in recommendation among a range of collaborative filtering algorithms according to Lian et al. (2014). In addition to showing the characteristics of implicit feedback, the check-in history also exhibits a skewed distributed check-in frequency. In other words, most POIs are only checked in just a few times, while only a few POIs have been checked in many times. In addition, check-in frequency is in an integer domain so that it can be well modeled by a Poisson distribution:

$$P(c_{i,j} = k) = \frac{\lambda_{i,j}^k}{k!} e^{-\lambda_{i,j}}.$$

This distribution can be used in a matrix factorization by assuming the mean $\lambda_{i,j}$ as the interaction between latent vectors of user u_i and POIs l_j, i.e., $\lambda_{i,j} = p_i^T q_j$. Due to the positiveness of $\lambda_{i,j}$, nonnegative constraints should be imposed on p_i and q_j. Assuming the independence and identical distribution of check-in frequency, the log-likelihood can be written as

$$\min_{P \geq 0, Q \geq 0} - \sum_{i,j} c_{i,j} \log(p_i^T q_j) + p_i^T q_j + \text{const}$$

Actually, this objective function is equivalent to a nonnegative matrix factorization (Seung and Lee 2001), whose loss function is taken as divergence from PQ^T to check-in frequency matrix C, i.e., $D(C|PQ^T) = \sum_{i,j} c_{i,j} \log \frac{c_{i,j}}{p_i^T q_j} - c_{i,j} + p_i^T q_j$. The objective function can be optimized by multiplicative update rules:

$$p_{i,f} = p_{i,f} \frac{\sum_j q_{j,f} c_{i,j} / p_i^T q_j}{\sum_j q_{j,f}},$$

$$q_{j,f} = q_{j,f} \frac{\sum_i p_{i,f} c_{i,j} / p_i^T q_j}{\sum_i p_{i,f}}$$

Since each user checks in at only a few POIs, the check-in frequency matrix is extremely sparse and such update rules can be completed efficiently. For the sake of avoiding the risk of dividing zero, the latent factors of users and items are usually assumed to follow gamma distribution, as used in Liu et al. (2013) and Cheng et al. (2012). Although these update rules can guarantee the decrease of the objective function, they actually optimize a lower bound of the original objective function so that it usually converges to a local optimal. Additionally, according to the comparison performed in Lian et al. (2014), nonnegative matrix factorization does not perform as well as implicit feedback-based weighted matrix factorization. The underlying reason is still related to discarding unchecked-in POIs directly in the multiplicative update rules due to multiplying $c_{i,j}$ with $q_{j,f}/p_i^T q_j$ or $p_{i,f}/p_i^T q_j$.

Social Filtering

Collaborative filtering involves collaboration between agents. However, there are not any constraints that these agents have a friendship with each other. Intuitively, check-ins are usually shared with friends, so when friends have seen the check-in information, if they are interested in the points of interest, they may also visit them in the near future. Therefore, social networks can be important for mobile check-in recommendation and have been exploited recently (Ye et al. 2011; Noulas et al. 2012; Lian and Xie 2014). Currently, there are two strategies for exploring social network information. The first one is social-based collaborative filtering, which is similar to user-based collaborative filtering, except defining user similarity based on social network information. The simplistic similarity between two users u_i and u_l is defined as 1 if they are friends and 0 otherwise, as used in Noulas et al. (2012). In this case, the score to a POI from a user can be expressed as the number of her friends who have checked in there. Another strategy is related to the proportion of their common friends, as used in Ye et al. (2011), simply defined as

$$s_{i,l} = \frac{|F_i \cap F_l|}{|F_i \cup F_l|},$$

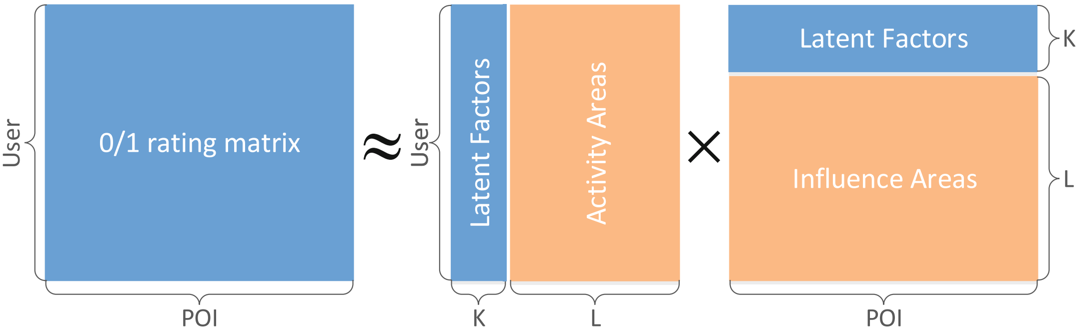
where F_i and F_l represent the friend sets of users u_i and u_l, respectively. In addition to being exploited in memory-based collaborative filtering, social network information has also been leveraged in the matrix factorization framework, since matrix factorization works better than memory-based collaborative filtering. Social network information has been taken into matrix factorization in many ways. One solution is based on the Laplacian regularizer, as used in Lian and Xie (2014) and defined as follows given the symmetric similarities between users based on social network information:

$$\Omega(S) = \frac{1}{2} \sum_{i,l} s_{i,l} \|p_i - p_l\|^2 = tr(P^T(D-S)P)$$

where $D_{i,i} = \sum_l s_{i,l}$. Actually $L = D - S$ is a Laplacian matrix, which is positive semidefinite. Therefore, the Laplacian regularizer is convex with respect to matrix P, i.e., users' latent factors.

Hybrid Recommendation

Since geographical clustering incurred from the physical interaction between users and POIs plays an important role in mobile check-in recommendation, its integration with collaborative filtering as well as social filtering is a popular research topic. For example, in Ye et al. (2011), Ye et al. proposed linearly combining them. That is,



Mobile Check-In Recommendation, Fig. 2 The augmented model for weighted matrix factorization, where the dimension of latent space is K and the number of grid areas is L

$$f_{i,j} = (1-\alpha-\beta)\frac{f_i(l_j)}{\max_k f_i(l_k)} + \alpha\frac{f_i^s(l_j)}{\max_k f_i^s(l_k)} + \beta\frac{P(l_j|L_i)}{\max_k P(l_k|L_i)},$$

where $f_i^s(l_j)$ is the score of social filtering, i.e., $f_i^s(l_j) = \frac{\sum_{u_l \in U} s_{i,l} \cdot r_{l,j}}{\sum_{u_l \in U} s_{i,l}}$, and the two weighting parameters α and β $(0 \leq \alpha, \beta \leq 1, 0 \leq \alpha + \beta \leq 1)$ denote the relative importance of the social filtering score and geographical check-in probability score compared to the score of user-based collaborative filtering. However, these parameters are manually tuned so that it is a time-consuming process; Cheng et al. (2012) proposed an empirically log-linear model to integrate a geo-clustering model with a nonnegative matrix factorization model with gamma prior placed on latent factors of both users and items. Although it doesn't require learning for integration, it lacks a probabilistic explanation. Liu et al. (2013) also proposed an empirical model, defined as

$$f_{i,j} \propto (p_i^T q_j + x_i^T W x_j)\rho(i)(d_0 + d(i,j))^{-\tau}.$$

In this equation, x_i and x_j are properties of user i and POI j. W is a matrix parameter mapping users and items into a joint latent space to capture the affinity between them. $\rho(i)$ is a factor of popularity of items. A power-law-like parametric term $(d_0 + d(i,j))^{-\tau}$ is to model the distance factor in the decision-making process. Lian et al. (2014) proposed a GeoMF model for seamlessly integrating weighted matrix factorization for implicit feedbacks with geographical modeling, as shown in Fig. 2. In this model, the influence area of a POI is considered as an extra part of POI's latent factors and activity area of a user is considered as an extra part of user's latent factors.

Their dot product just corresponds to two-dimensional kernel density estimation. In particular, it minimizes the following objective function:

$$\min_{P,Q,X} \|W \odot (R - PQ^T - XY^T)\|_F^2 + \gamma(\|P\|_F^2 + \|Q\|_F^2) + \lambda\|X\|_1$$
$$\text{subject to } X \geq 0$$

where X is a matrix stacking a user's activity area by columns and Y is a matrix stacking the items' influence area vector by columns. ℓ_1 norm of matrix $\|X\|_1$ constrains that users usually stay around server important locations, such as home or workplaces. The optimization procedure consists of one procedure of learning latent factors when X is fixed and another procedure of nonnegative weighted least square with sparse constraints when latent factors are fixed. Given fixed X, updating P and Q is similar to weighted matrix factorization except for substituting R with $R - XY^T$ since the interaction between P and Q only captures the residual rating without being captured by the geographical influence. When fixing P and Q, since the learning activity area vector of different users is independent from each other, the objective function with respect to user u_i is expressed as follows:

$$\min_{x_i} \|W^i(r_i - Qp_u - Yx_i)\|_F^2 + \lambda\|x_i\|_1$$
$$\text{subject to } \mathbf{x}_i \geq 0$$

This can be done by projected gradient descent method, as shown in Lian et al. (2014). The general idea of a projected gradient descent algorithm is to update parameters by gradient descent and then to project them into feasible regions defined by bound constraints (nonnegativeness). Nevertheless, the choice of learning rate in the gradient descent should guarantee that the projected parameters can sufficiently decrease the objective function. When only integrating collaborative filtering with a social network, Noulas et al. (2012) and Ye et al. (2011) proposed a random walk-based solution. Specifically, they constructed a graph whose nodes consist of users and POIs, whose links between users indicate friendships, and whose links between users and items are related to physical interactions between them and then performed a random walk with a restart for POI recommendation. The problem of this method is, as analyzed in Ye et al. (2011), the importance of two pieces of information cannot be discriminated from each other. More perfect integration of social network information with collaborative filtering can be achieved by leveraging matrix factorization and graph Laplacian regularization, as used in Lian and Xie (2014).

Key Applications

Mobile check-in recommendation could be useful in many applications. It is absolutely necessary to help people speed up familiarizing themselves with their surroundings when they arrive at new places and to discover novel and serendipitous venues. Due to this benefit, it has been used in recognizing which POIs users are staying (Lian and Xie 2014), which is an essential component in indoor localization and has been exploited for predicting future locations (Lian et al. 2013) since users may occasionally explore new places and visit new and attractive POIs. Additionally, since where users go could reflect their underlying interest and be related to who the users are, mobile check-in recommendation can be useful for inferring user demographics (Zhong et al. 2015).

Future Directions

Mobile check-in recommendation has been studied from various perspectives. There are still many promising research directions. For example, although weighted matrix factorization for implicit feedback has been empirically shown in mobile check-in recommendation, a more general framework incorporating content information of users and items is still missing. Although weighted matrix factorization works well for the general setting of a regularized coefficient and weighted matrix, there is still no automatic method, e.g., a Bayesian framework, for learning them. When more check-in data is accumulated, how to deal with large-scale learning problem is a big issue based on these existing algorithms. Given the location information, the behaviors of users could be observed in a different geographical level, e.g., in a region level, in a city level, and in a country level. How to recommend points of interest in this situation is an interesting problem.

References

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749

Bennett J, Lanning S (2007) The netflix prize. In: Proceedings of KDD cup and workshop, San Jose, vol 2007, p 35

Celma O (2010) Music recommendation and discovery: the long tail, long fail, and long play in the digital music space. Springer, Berlin/Heidelberg

Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks. In: Proceedings of AAAI'12, Toronto

Horozov T, Narasimhan N, Vasudevan V (2006) Using location for personalized poi recommendations in mobile environments. In: Proceedings of SAINT'06. IEEE Computer Society

Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings of ICDM'08. IEEE, pp 263–272

Lian D, Xie X (2014) Mining check-in history for personalized location naming. ACM Trans Intell Syst Technol 5(2):32:1–32:25

Lian D, Xie X, Zheng VW, Yuan NJ, Zhang F, Chen E (2015) Cepr: a collaborative exploration and periodically returning model for location prediction. ACM Trans Intell Syst Technol, 6(1): 8:1–8:27

Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 831–840

Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. Internet Comput IEEE 7(1):76–80

Liu B, Fu Y, Yao Z, Xiong H (2013) Learning geographical preferences for point-of-interest recommendation. In: Proceedings of KDD'13. ACM, pp 1043–1051

Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York

Miller BN, Albert I, Lam SK, Konstan JA, Riedl J (2003) Movielens unplugged: experiences with an occasionally connected recommender system. In: Proceedings of the 8th international conference on intelligent user interfaces. ACM, pp 263–266

Noulas A, Scellato S, Lathia N, Mascolo C (2012) A random walk around the city: new venue recommendation in location-based social networks. In: Proceedings of SocialCom'12. IEEE, pp 144–153

Pan R, Zhou Y, Cao B, Liu NN, Lukose R, Scholz M, Yang Q (2008) One-class collaborative filtering. In: Proceedings of ICDM'08. IEEE, pp 502–511

Park MH, Hong JH, Cho SB (2007) Location-based recommendation system using bayesian user's preference model in mobile devices. In: Ubiquitous intelligence and computing. Springer, Berlin/Heidelberg, pp 1130–1139

Seung D, Lee L (2001) Algorithms for non-negative matrix factorization. Adv Neural Inf Process Syst 13:556–562

Takeuchi Y, Sugimoto M (2006) Cityvoyager: an outdoor recommendation system based on user location history. In: Ubiquitous intelligence and computing. Springer, Berlin/Heidelberg, pp 625–636

Yang D, Zhang D, Yu Z, Wang Z (2013) A sentiment-enhanced personalized location recommendation system. In: Proceedings of the 24th ACM conference on hypertext and social media (HT'13). ACM, pp 119–128

Ye M, Yin P, Lee W-C (2010) Location recommendation for location-based social networks. In: Proceedings of GIS'10. ACM, pp 458–461

Ye M, Yin P, Lee W-C, Lee D-L (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of SIGIR'11. ACM, pp 325–334

Zhong W, Zhang F, Xie X, Zhong Y, Yuan NJ (2015) You are where you go: inferring demographic attributes from location check-ins. In: Proceedings of the 8th ACM international conference on web search and data mining (WSDM), Shanghai

Zhang J-D, Chow C-Y (2013) igslr: personalized geo-social location recommendation-a kernel density estimation approach. In: Proceedings of GIS'13, Orlando

Zheng Y, Zhang L, Xie X, Ma WY (2009) Mining interesting locations and travel sequences from gps trajectories. In: Proceedings of WWW'09. ACM, pp 791–800

Zheng VW, Cao B, Zheng Y, Xie X, Yang Q (2010) Collaborative filtering meets mobile recommendation: a user-centered approach. In: Proceedings of AAAI'10. AAAI Press

Zheng VW, Zheng Y, Xie X, Yang Q (2010) Collaborative location and activity recommendations with gps history data. In: Proceedings of WWW'10. ACM, pp 1029–1038

Mobile GIS Solutions for Retail and Advertising

Jingyuan Yang
Rutgers University, New Brunswick, NJ, USA

Synonyms

GIS advertising; GIS retailing; Mobile advertising

Definition

Integrated with GPS-enabled smart mobile devices, mobile geographic information system (GIS) involves location-based services (LBS) to provide richer spatial information and serve as an excellent analytical tool for business analytics. Location intelligence is very significant for many aspects of retail business, such as market analysis, store location selection, marketing or advertising, distribution, delivery, and facilities management. Mobile GIS offers retailers and marketers a better solution to integrate and visualize the geographic intelligence and enhance effectiveness, efficiency, and decision-making in these processes.

Historical Background

As the number of mobile devices, such as smart phones and tablets that are GPS enabled, drastically increased in the last decade, more and more geographic or spatial data is generated and collected. With this increase in the availability of large amount of geospatial data for business, it is crucial that business decision-makers integrate their domain business knowledge with the location intelligence to better reveal the current trends, patterns, and opportunities that may not be discovered without the help of location intelligence. With the ubiquity mobile devices have and the development of wireless networking technology, GIS evolves into the mobile GIS, which is becoming a powerful analytic tool to manage, visualize, and analyze business information.

As the competitive environment gets more intense, retailers and marketers must spare no effort to acquire new customers, keep customers from churning out, select profitable store locations, schedule market expansion, and move further ahead of the competitors. In fact, GIS has been applied in real-world geographic business problems to achieve competitive advantages since the 1960s (ESRI 2007). Today more companies and organizations are using GIS to analyze the company's performance and to make strategic plans. In recent years, various mobile GIS applications and packages on business, such as retailing (Roig-Tierno et al. 2013; Cheng et al. 2007) and marketing (especially advertising) (Kölmel and Alexakis 2002; Hristova and O'Hare 2004), have been innovated and implemented by many companies to gain sustainable competitive advantages in market.

For example, retail site selection is one of the most popular GIS applications. Retail site selection is very crucial to the success of a business; Finn and Louviere (1990) analyze the customer segments regarding the geographical region to select the best shopping mall location; Harris and Batty (1993) and Birkin et al. (2002) discuss the possibilities of geographical information and location models to solve retail outlets planning and locating. Another example of mobile GIS application is mobile advertising. Mobile advertising is a dynamic young field but is growing very quickly. There are many frameworks of mobile GIS that can extract surrounding information based on customers' current location to decide what and how to best advertise to them, which will provide a deeper understanding of customers, improve the marketing effectiveness, and ultimately increase ROI. For instance, Hristova and O'Hare (2004) developed a mobile advertising framework adapted to the user location and mobile devices. This framework is location based and context aware and can generate personalized context for mobile customers.

Scientific Fundamentals

The Concept of Mobile GIS

Mobile GIS is a recently developed technology for the access of geographic data and location-based services through mobile devices, such as laptops, tablet PCs, pocket PCs, personal digital assistants (PDA), and mobile phones. In fact, mobile GIS is an integrated framework to obtain, store, and manage spatial data and process services through wireless devices by using network connections to distributed systems (Peng and Tsou 2003). By using mobile GIS with GPS, Internet, and wireless communication technologies, mobile GIS has great potential to play an important role in business fields such as retailing, marketing, financing, and so on.

There are two major application categories of mobile GIS (Tsou 2004): (*a*) field-based GIS focuses on data collection and needs to edit or change the original GIS data or modify their attributes, and (*b*) location-based services focus on business-related location management functions, such as site location, navigation, routing, or vehicle tracking, usually only referring to the original GIS dataset.

The Properties of Mobile GIS

Mobility, distributiveness, and egocentric awareness are the principal properties of the mobile GIS (Frank et al. 2004).

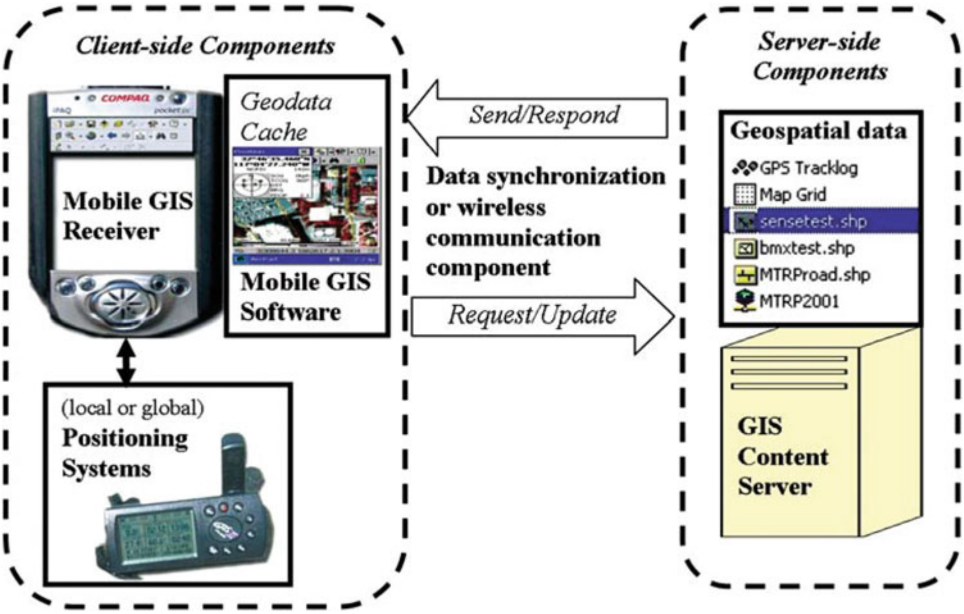
- Mobility: Mobile GIS can support users to interact with not only the abstract map representation system of the real world but also their surrounding objects such as buildings or people. Complex structure of GISs and customized interfaces are needed to be designed to meet the requirements of mobile environment.
- Distributiveness: To provide comparative processing power, a distributed system is needed. The mobile devices will include a thin client-side component, an Internet connection, a server-side component, and a geographic database management system. Therefore, the data storage and management and computer processing occur in different places. In this way, it will enable the mobile device to process the manageable tasks of coordinating the users' interactions.
- Egocentric Awareness: Users usually relate their surrounding objects to their positions in different perspectives. Therefore, mobile GIS needs to adapt the way of presentation from birds-eye view to egocentric view. To achieve this goal, personalized map content is needed to be created based on context-awareness of the user's position, orientation, and task at hand.

The Architecture of Mobile GIS

Different architectures have been developed for mobile GIS implementation, such as stand-alone architecture, client/server architecture, distributed client-server architecture, and so on. The simplest mobile GIS architecture is the stand-alone architecture, which include the mobile GIS application and geospatial data entirely in the mobile device. To tackle the low mobile device memory problem and the need of communication with any other applications, client/server architecture is widely adopted. The architecture of the mobile GIS follows the traditional Internet GIS client/server architecture. The client-side mobile GIS components are the end-user hardware devices that can send geographic task requests and provide digital maps or analytical results of GIS operations. The server-side components manage comprehensive geospatial data and process GIS operations based on task requests from the client-side components. Between the client and the server, there are wired/wireless networks to support the communications of exchange data and services.

There are six basic components of mobile GIS (Tsou 2004) as presented in Fig. 1:

- *Positioning systems* are the tools that provide geographic coordinate information to mobile GIS receivers. There are two types of positioning systems: one is local positioning systems, which use triangulation of the radio signals or cellular phone signals from multiple base stations, and the other one is global positioning systems (GPS), which use satellite signals to calculate the position of GPS units.
- *Mobile GIS receivers* are handheld computers that can display maps and geographic information to end users, for example, smart phones, tablet PCs, or PDAs are the most common mobile GIS receivers. However, the factors of small screen resolutions (240 × 300), limited storage memory, and slow CPU processing speed compared to desktop personal computer limit the functionality and/or performance of mobile GIS.
- *Mobile GIS software* refers to the specific GIS software packages developed for mobile GIS applications. Most mobile GIS software packages are compact and customizable and focus on specific GIS operations, for example, location matching, routing services, navigation, or map display.
- *Geospatial data* is personalized GIS layers used in mobile GIS applications. Due to the limited storage space in mobile GIS receivers, mobile GIS receivers will store compressed geospatial data in a temporary GIS storage space. The customized dataset can be accessed in two ways: one way is to download or synchronize from GIS content server and the other way is to use wireless network communications to retrieve the most up-to-date geospatial information from the content server directly.



Mobile GIS Solutions for Retail and Advertising, Fig. 1 A general architecture of mobile GIS (Tsou 2004)

- *The data synchronization/wireless communication component* is the communication mechanism connecting the mobile GIS receivers with GIS content servers. The communications could be two-way communications and through either real-time wireless communications (via Wi-Fi or cellular phone signals) or cable-based data synchronization communications (via USB or serial ports). Other middlewares or data synchronization software packages (such as Microsoft ActiveSync or Web Services) are needed for mobile GIS applications.
- *GIS content servers* are server stains which provide geospatial data or map services to mobile GIS receivers. The content servers could be stand-alone workstation or web-based server stations. Stand-alone GIS workstations are appropriate for the majority of cable-based mobile GIS receivers. Wireless-based mobile GIS receivers may require more advanced web or map servers for accessing multilayer geospatial data. A single GIS content server can serve for multiple mobile GIS receivers at the same time.

The Mobile GIS Solutions for Retail and Advertising

In general, mobile GIS provides four functions to address the needs of business such as retail and advertising (Azaz 2011; Cheng et al. 2007):

(*a*) *Graphic Display*: the fundamental GIS capability of representing displays of data and information using a coordinate system to emphasize the geographic relationships among map elements;
(*b*) *Database Management and Integration*: the capability of GIS to store, edit, integrate, and provide access to data;
(*c*) *Data Modeling*: providing support for analysis and decision-making based on spatial relationships; and
(*d*) *Design and Planning*: based on spatial information and relationships to design and create strategic planning.

Specially, mobile GIS can assist retailers and marketing in the following ways:

- *Retail* Location is very crucial for the success or failure of a retail business. Therefore, retailers can leverage the location intelligence provided by mobile GIS to:

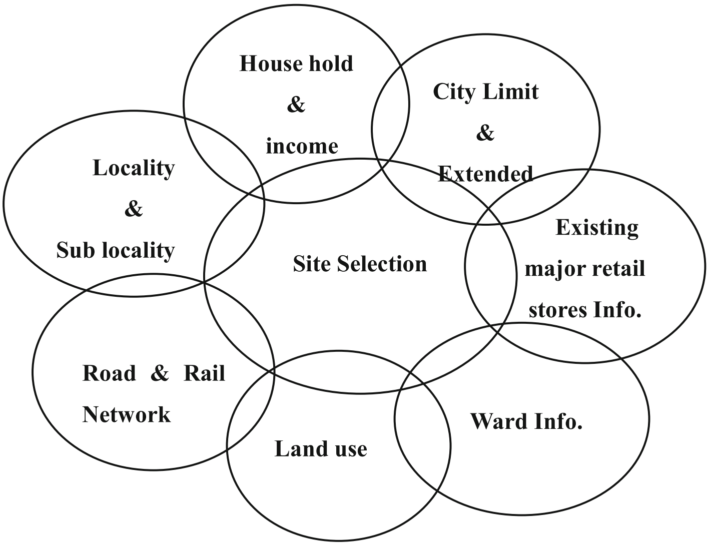
- Determine a specific forecast projection for a new store
- Analyze and interpret the market characteristics of different market regions
- Select profitable store locations
- Maximize market share and improve store performance
- Develop local media or marketing campaign to targeted customer segments
- Design optimal delivery routes from stores to customers
- Analyze and recommend solutions for underperforming stores

- *Advertising* Besides the important business decision of choosing the best locations for business, a successful advertising strategy is also needed to increase the effectiveness of advertising by helping businesses reach their target consumers as well as increasing marketing ROI. Using mobile GIS, the marketers can develop customized advertising campaigns, which are tailored for different types of customers. To realize the mobile context-sensitive adverting, context-aware computing technique is needed and applied to identify the set of environmental states and settings that is interesting to the user. Next, in order to deliver more targeted and personalized advertisement content, the collection of personal profiles is required to show the characteristics and preferences of a particular user. Then, user geographic location is integrated with the previous information because the context-aware advertising is highly relevant to the current user locations.

Key Applications

Retail Application Mobile GIS

Finding the right site locations is very important to many retail organizations. In recent years, GIS has been applied in retail industry to select an optimal site for a new store or a service department. In general, to quickly analyze the potential of future sites, retailers will consider the size and characteristics of the consumer population within a specific retail area. In particular, geodemographic information, that is, the combination of geographic and demographic information, is used to reach an insightful understanding of potential market. Figure 2 illustrated associated information considered for new retail store site selection. Geodemographic segmentation systems are commonly used to assess the penetration rate of different market segments targeted by the retailer. Next, for the specific retail area, a geographic region is established within a predefined drive



Mobile GIS Solutions for Retail and Advertising, Fig. 2 Associated data layers considered for new retail store site selection (Sreekanth et al. 2013) time from the store location. Sites should be located where maximum number of households belonging to the specific groups were the most historically profitable and have high potentials to demand the particular good or service, therefore deserving a stronger focus and higher marketing budget. Further, customer-modeling applications can integrate geodemographic data to relate existing information to surrounding neighborhood demographic data. This provides an ideal understanding of where the most profitable customer clusters are located and a fine-tuned product line they would prefer in terms of demographics, characteristics, and lifestyle habits. In summary, with the location intelligence provided by GIS, retailers can visualize market penetration, market share, and trade areas and ultimately increase profitability and realize greater ROI.

Compared to traditional GIS solutions, mobile GIS provides more attractive features for retailers. First of all, besides the common paper format analysis report containing multilayer map descriptions, demographics, and so on, interactive mobile GIS apps are used to view and interact with the services, such as seeing a map with the prospective sites marked and even querying and viewing all related site documents. Mobile GIS is a far more efficient tool to view and find information about potential new retail locations. In addition, feedback in notes or video format can also be provided back to adjust the analysis.

Furthermore, the established mobile GIS can help customers find not only the list of the retail locations but also more detailed products or services information that customers are interested in. An interactive store locator services can integrate inventory and pricing information of the product the customers are looking for. Therefore, the retailer can increase store traffic and improve the purchase possibility of customer.

Mobile Personalized Advertising Using Mobile GIS

Mobile GIS not only helps consumers research and compare retailers and wanted products for making smart buying decisions but also helps marketer to be aware of the presence of potential customers and send them real-time advertisements triggered by their physical location. Analyzing and relating to user's geographic location is very important to create context-sensitive advertisement. In general, mobile GIS can help marketers design context-aware and personalized advertisement for the mobile users based on (Hristova and O'Hare 2004):

(*a*) *User context*: about user's identity, users' profile, location relationships, and orientation;

(*b*) *Computing context*: network connectivity and bandwidth, wireless device information, and operating system;

(*c*) *Physical context*: the surrounding environment, nearby objects, and history context (if available);

(*d*) *Time context*: such as time of the day, day of the week, month, and season of the year.

More specifically, marketers can integrate more information-based products or services that may interest potential customers in terms of their location using mobile GIS. The context-sensitive advertisements may include description of the products that are of high interest to the customers or show affordable products of mobile users or more focus on products and services within the immediate user locality or available at a given time or in stock.

Future Directions

In summary, mobile GIS is a very promising technology with strong demands from both retail and marketing industry. With the progress of new mobile GIS technologies, many future trends are happening and developing for mobile GIS. For example, mobile GIS can integrate with social media networks. Social media networks (such as Facebook and Twitter) have become popular places for users to share location and sentiment. This information can greatly help provide the context needed for retailers and marketer to gain insights about the characteristics of customers and to create precisely personalized marketing offers. Combining social media information and location intelligence can produce richer insight into customer and market dynamics. Another possible trend may be indoor location technology. With the high-quality indoor location data, it is possible for indoor positioning, indoor maps, indoor navigation and routing, indoor tracking, and locating for advertisement and retails. Related to the trend described above, the precise location could be used for predicting user' intent, to better design the real-time business strategy and marketing plans.

Cross-References

- ▶ Location-based Recommendation Systems
- ▶ Mobile check-in Recommendation
- ▶ Mobile Marketing
- ▶ Time-Aware Personalized Location Recommendation

References

Azaz L (2011) The use of geographic information systems (GIS) in business. In: International conference on humanities, geography an economics–ICHGE, Pattaya, pp 299–303

Birkin M, Clarke G, Clarke MP (2002) Retail geography and intelligent network planning. Wiley, Chichester

Cheng EWL, Li H, Yu L (2007) A GIS approach to shopping mall location selection. Build Environ 42(2):884–892

ESRI (2007) GIS for retail business. Available via: http://www.esri.com/library/bestpractices/retail-business.pdf. Accessed 19 May 2016

Finn A, Louviere J (1990) Shopping-center patronage models: fashioning a consideration set segmentation solution. J Bus Res 21(3):259–275

Frank C, Caduff D, Wuersch M (2004) From GIS to LBS–an intelligent mobile GIS. IfGI Prints 22:261–274

Harris B, Batty M (1993) Locational models, geographic information and planning support systems. J Plan Educ Res 12(3):184–198

Hristova N, O'Hare GMP (2004) Ad-me: wireless advertising adapted to the user location, device and emotions. In: Proceedings of the 37th annual Hawaii international conference on system sciences, Big Island. IEEE, 10pp

Kölmel B, Alexakis S (2002) Location based advertising. In: Proceedings of the first international conference on mobile business, Athens

Peng Z-R, Tsou M-H (2003) Internet GIS: distributed geographic information services for the internet and wireless networks. Wiley, Hoboken

Roig-Tierno N, Baviera-Puig A, Buitrago-Vera J (2013) Business opportunities analysis using GIS: the retail distribution sector. Glob Bus Perspect 1(3):226–238

Sreekanth PD, Kumar KV, Soam SK, Rao NH, Bhaskar K (2013) GIS-based decision support system (DSS) for recommending retail outlet locations. Inf Knowl Manage 3(4):57–66

Tsou M-H (2004) Integrated mobile GIS and wireless internet map servers for environmental monitoring and management. Cartogr Geogr Inf Sci 31(3):153–165

Mobile Maps

▶ Web Mapping and Web Cartography

Mobile Marketing

Xiaolin Li, Jiang Wu, and Yue Sun
School of Management, Nanjing University,
Nanjing, China

Synonyms

Market intelligence; Marketing information system

Definition

Mobile marketing refers to "the two-way or multi-way communication and promotion of an offer between a firm and its customers using a mobile medium, device, or technology" (Shankar and Balasubramanian 2009). With mobile marketing, the marketers communicate directly with consumers at any time or place. Key features of mobile marketing are location specificity, portability, and untethered/wireless features (Shankar and Balasubramanian 2009). Mobile devices are usually compact and easily portable. When device users move about, significant amounts of personal and location-specific information is generated. Marketers can use geolocation technologies to target the consumer with location-sensitive marketing promotions.

Historical Background

Following widespread mobile device market penetration and recent developments in high-speed wireless network technologies, mobile marketing has become increasingly important since the mid-1990s. Push notification was one of the most popular early mobile marketing applications. Marketers used short message service (SMS) and later multimedia messaging service (MMS) to push texts, pictures, mobile links, or other rich contents to consumers. The services could be used for mobile couponing, customer relationship management, and entertainment.

Push notification was but one mobile marketing application. The development of geo-location technologies helped mobile marketing reach its potential. Most mobile devices have location capabilities such as the Global Positioning System (GPS) to identify physical location. When consumers use mobile phones, significant quantities of geographic information are generated and collected by mobile service providers helping to originate the firm's geographic mobile information system (GMIS). These data enable marketers to increase their knowledge of consumer's mobile usage behaviors and in turn give full play to mobile marketing techniques. By utilizing GMIS, marketers not only target and segment markets more precisely but also formulate multiple approaches to marketing strategies. For instance, location-based service (LBS) is now used in a variety of contexts and regarded as the mobile marketing "killer application." Starbucks customers, for example, will receive messages highlighting offers or coupons as they cross a company geofence. Conversely, when prospective customers want a cup of coffee, they can ask the LBS to locate the nearest Starbucks. Marketers acknowledge mobile marketing's great potential. In 2015, mobile marketing generated $400 billion in sales with marketers spending $19.8 billion on mobile marketing campaigns (Hsu).

Push notification is only one of the applications of mobile marketing. The development of geo-location technologies help mobile marketing reach its potential. Most mobile devices have location capacities such as Global Positioning System (GPS) to identify their physical location. When consumers are using mobile phones, huge massive geographic information is generated, which could be collected by the mobile service providers to form the original source of firms' geographic mobile information system (GMIS). It enables marketers to increase their insights into the consumers' mobile using behaviors and gives full play to advantages of the mobile marketing. By utilizing the GMIS, marketers can not only make a more precise market segmentation and targeting but provide various approaches for marketing strategies. For instance, location-based service (LBS) is now used in a variety of contexts and regarded as a "killer application" of mobile marketing. Taking Starbucks as an example, people will receive a message highlighting an offer or a coupon when they cross a Starbucks geofence. In other cases, when people want to have a cup of coffee, they can ask the LBS to tell them where the nearest Starbucks is. The high potential of mobile marketing is well acknowledged by marketers. In 2015, mobile marketing will generate $400 billion in sales, and marketers will spend $19.8 billion on mobile marketing (Hsu).

Scientific Fundamentals

Mobile Geographic Information Systems

The definition of geographic information system (GIS) as found in peer-reviewed literature relates this technology to any tool that associates databases and digitized maps. A complete GIS has at least five components: software, hardware, geographical data, people, and organization. It is therefore a framework to facilitate decision making involving the use of geo-referencing information within an organization. GIS is becoming a key component of business systems where it was previously restricted to geospatial analysis.

Mobile GIS can be considered an extension of existing commercial GIS. Several factors motivated mobile computing in general and mobile GIS in particular. First was the expansion of wireless communication. Second, mobile networks were commonplace worldwide. Third was the exponential advances in hardware design and finally geographical database availability. Consequently, the PC and Internet revolution dissolved into the mobile revolution and replaced it. Now, as the technology becomes increasingly robust, there are sound business reasons to encourage commercial mobile workforce expansion. Since mobile GIS can be handheld anywhere, anytime, an entire workforce may access data and operations previously reserved for organizational decision making.

Mobile Data Collection

Data concerning our daily lives along with location information collected in mobile devices has triggered in-depth research:

- Radio signal localization has attracted considerable attention in telecommunication and navigation recently. The best known positioning system is GPS (Kaplan and Hegarty 2005) which is satellite based and optimal for tracking outdoor users. However, GPS satellite signals do not easily penetrate the built environment and so fail indoors. Indoor radio waves are characteristically site specific, exhibiting severe multipath effects with low probability of line-of-sight (LOS) signal propagation between transmitter and receiver. This makes accurate indoor positioning challenging. Several wireless technologies have been proposed for indoor location sensing like infrared, ultrawide band (UWB), WiFi, and more recently the Radio Frequency Identification (RFID) (Papapostolou and Chaouchi 2011).
- RFID technology stores and retrieves data via electromagnetic transmissions to a radio frequency compatible integrated circuit. It is now considered a radical means of enhancing data-handling processes (Burdet 2004). It identifies RFID tag positions using installed RFID readers and servers (Bouet and Santos 2008). RFID offers promise for accurate fast tracking (Papapostolou and Chaouchi 2011) and is suitable for dense environments.
- As many buildings and structures have existing WLAN infrastructures, wireless systems are easier and cheaper options compared to other indoor positioning techniques. GPS integration into today's wireless technology is regarded as key complementary to location-aware systems (Sharaf and Noureldin 2007). Some common methodologies developed over the years include angle of arrival (AoA), time of arrival (ToA), time difference of arrival (TDOA), and time of flight (ToF). Currently, WiFi (WLAN based on 802.11 standards) is widely used in public areas and buildings such as airports, offices, and hospitals. With WiFi receivers commonly embedded in consumer devices like mobile phones, implementing WiFi locating systems via existing WLAN infrastructures in public areas is relatively straightforward (Cheng et al. 2014).

Data Mining in Mobile GIS

Data mining is an analytical process which examines mega data specifically for business, market, or scientific research. Although GIS and data mining are two powerful techniques, they are normally used independently in marketing. Nowadays, this is considered incorrect as the large amount of business data organization collect can also be analyzed by data mining techniques under the geographical location umbrella. According to the specific application, location information can include customer history and retail sale data series along with business, traffic, demographic, and market research.

LBS have become more widely used by business to gain a competitive edge. LBS will boom in coming years as mobile geographical location technology improves. For instance, customers can be easily located and traced inside shopping centers while using mobile payment applications. By analyzing this, data companies can better focus marketing and advertising campaigns. These improvements have changed the marketing sciences. The new levels of knowledge and accuracy regarding market segmentation and market mix make for dramatic improvements in decision making.

Marketing Information Systems

Marketing has many definitions. All are related to customer satisfaction. It may be said that marketing is a science that identifies, creates, and satisfies customer's needs. Marketing looks for best possible combinations between segments, supply, and demand. Kotler defined marketing information system (MKIS) as a structure consisting of people, equipment, and procedures to gather, sort, analyze, evaluate, and distribute needed, timely, and accurate information to marketing decision makers (Kotler 1997). This definition is independent of any specific computer technology. Other components in a MKIS can be added, deleted, or enhanced to suit the decision maker's purposes.

Both Kotler (1997) and Burns and Bush (2000) present nearly identical MKIS models which show relationships between managerial tasks, uses of the system, system information development, and decisions within the marketing environment (Fig. 1). The model shows a system with several components each with a number of interrelationships. Kotler's definition covers the infrastructure and procedures needed to implement the model. The GIS server and the MKIS system can also be integrated as shown in Fig. 2.

Key Applications

Market Segmentation Using Mobile GIS

Mobile users are seldom without their devices. This means marketers can reach prospects appropriately and effectively within their campaigns. This personal attachment to devices therefore offers marketers easy targeting, accountability, and, potentially, high interactivity. As geographic information has always been an important segmentation and targeting tool, retailers have used it for store location information since the early twentieth century. Advertisers now seek customer's prior permission based on the location data to send marketing messages via electronic channels, while marketers use consumers' location data for direct marketing. A basic assumption when using geographic information in market segmentation is that people within a certain region may have similar psychological or behavioral consumption patterns. However, due to the absence of an individual's behavioral data, geographic information used in traditional market segmentation is not sufficient to profile consumers' specific characteristics in any segment market. Mobile device market penetration has changed mass marketing. Mobile devices exhibit some important characteristics: location specificity, portability, untetheredness, and personalization (Shankar and Balasubramanian 2009). Marketers can collect not only individual spatiotemporal locations but also product needs and consumption habits. As such, mobile marketing targets customers more precisely at specific locations and times than ever before (Shankar et al. 2010).

Product Strategy Using Mobile GIS

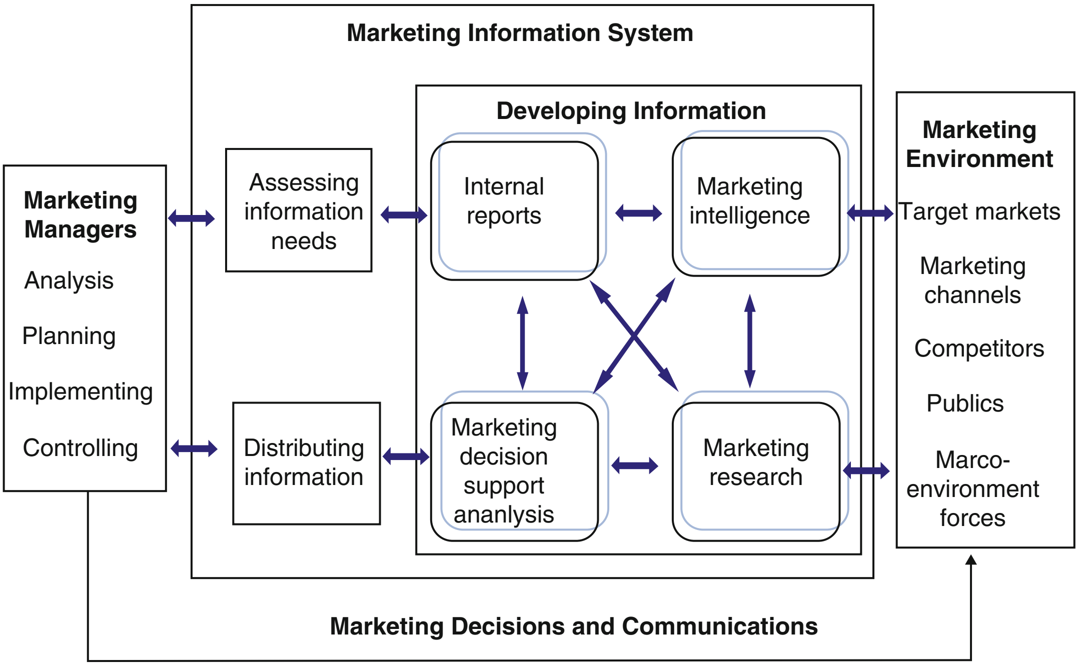
Mobile geographic information systems help marketers understand customers' needs by providing customized, location-based products or services. Location information itself can also be an entirely new product or can add value to current services.

Below are typical location-based services (Dhar and Varshney 2011):

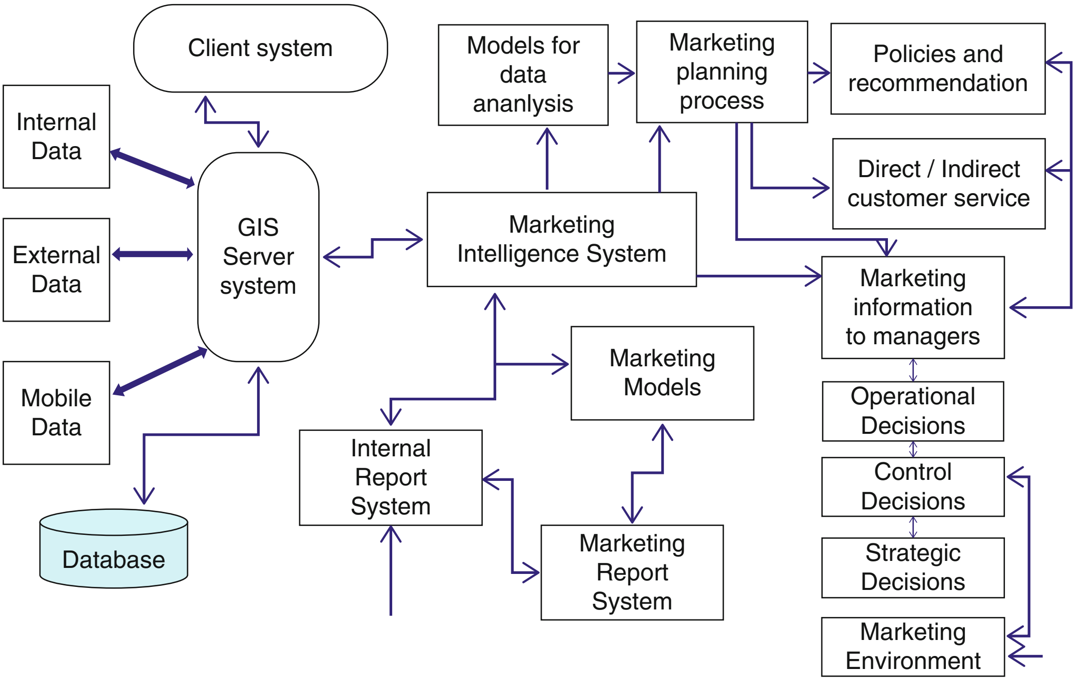
- *Information/directory services:* e.g., dynamic yellow pages informing consumer of location of nearest restaurants or shopping malls
- *Tracking and navigation services:* e.g., dynamic navigation guidance or locating lost pets and mobile advertising
- *Emergency services:* e.g., E911, emergency medical ambulance, and roadside assistance.

Price Strategy Using Mobile GIS

Mobile payment refers to a mobile device being used to authorize, initiate, or realize a commercial transaction (Schierz and Schilke 2010). Kim et al. (2010) suggested that mobile payment systems have four characteristics: mobility, reachability, compatibility, and convenience. Mobile payment technologies enable consumers to pay for goods or services using mobile devices via SMS, WAP billing, mobile Web, direct-to-subscriber billing, and direct to credit cards. Cur-



Mobile Marketing, Fig. 1 The Kotler (1997)/Burns and Bush (2000) model for marketing information systems



Mobile Marketing, Fig. 2 GIS and MKIS integration model rently, some consumers can also make payments through more advanced technologies like near field communication (NFC). According to Strategy Analytics research, payments made through mobile NFC will exceed $130 billion worldwide in consumer retail sales by 2020.

Place Strategy Using Mobile GIS

Retailers can use indoor navigation to guide consumers to where they want them to go in shopping malls. In-store navigation uses interrelated building representations including semantic, geometric, as well as property information and spatial relationships. Information from various sensors is acquired and then transformed for users into enhanced images or acoustic maps via mobile phones (Serrao et al. 2012). Users receive vocal prompts which provide ambulatory guidance. For example, in 2012, Macy's added a feature to its own smartphone APP which provides in-store navigation. Using this APP, Macy's customers can find what they want conveniently and receive offers based on where they are standing.

Promotion Strategy Using Mobile GIS

A key application in mobile promotion strategies is location-based advertising (LBA). LBA is defined as "marketercontrolled information specially tailored for the place where users access an advertising medium (Dhar and Varshney 2011)." There are two types of LBA: pull and push (Bruner and Kumar 2007). In the pull type, consumers voluntarily provide location data and initiate requests for services. This type of advertising includes travel directions, taxi hailing, mobile yellow pages, buying services, and instant information. In the push type, service providers identify consumer's location information first then push the information to them without the consumer requesting it. Push-type advertising may therefore raise some privacy concerns. Typical push advertising includes notifications, friend finders, zone alerts, and traffic alerts (Koeppel 2000).

Future Directions

In summary, mobile marketing is an effective method to help make precise decisions in targeting and diversified marketing strategies. Compared to the high rate of mobile device market penetration, mobile marketing is still in its infancy. There is still broad research space available for mobile marketing commercial applications. Another possible future trend is integrated marketing. Mobile devices are often called "the third screen" (Martin 2011) after the television and the computer. This suggests mobile marketing is not the sole province of big firms. With future research integrating diverse approaches to mobile marketing, privacy and security have also been hotly debated issues. Companies need to make greater efforts at keeping the balance between consumer privacy and high-quality services.

References

Bouet M, Santos AL (2008) RFID tags: positioning principles and localization techniques. In: 1st IFIP wireless days, Dubai, vol 11, pp 1–5

Bruner GC, Kumar A (2007) Attitude toward location-based advertising. J Interact Advert 7(2):3–15

Burdet LA (2004) RFID multiple access methods. Smart Environments Seminar, Zurich, 11

Burns AC, Bush RF (2000) Marketing research, 3rd edn. Prentice-Hall, Upper Saddle River

Cheng J, Yang L et al (2014) Seamless outdoor/indoor navigation with WIFI/GPS aided low cost Inertial Navigation System. Phys Commun 13:31–43

Dhar S, Varshney U (2011) Challenges and business models for mobile location-based services and advertising. Commun ACM 54(5):121–128

Hsu E. Mobile marketing. Mobile marketing. Haettu 18 Apr 2014 osoitteesta http://blogs.uoregon.edu/emmahsu/files/2014/02/Mobile-Marketingt5qr6l.pdf

Kaplan ED, Hegarty CJ (ed) (2005) Understanding GPS: principles and applications, 2nd edn. Artech House, Boston

Kim C, Mirusmonov M et al. An empirical examination of factors influencing the intention to use mobile payment. Comput Hum Behav 26(3):310–322 (2010)

Koeppel I (2000) What are location services?-from a GIS perspective. ESRI white chapter

Kotler P (1997) Marketing management: analysis, planning, and control, 9th edn. Prentice-Hall, Upper Saddle River

Martin C (2011) The third screen: marketing to your customers in a world gone mobile. Nicholas Brealey Publishing, Boston

Papapostolou A, Chaouchi H (2011) RFID-assisted indoor localization and the impact of interference on its performance. J Netw Comput Appl 34(3):902–913

Schierz PG, Schilke O et al (2010) Understanding consumer acceptance of mobile payment services: an empirical analysis. Electron Commerce Res Appl 9(3):209–216

Serrao M, Rodrigues JM et al (2012) Indoor localization and navigation for blind persons using visual landmarks and a GIS. Proc Comput Sci 14:65–73

Shankar V, Balasubramanian S (2009) Mobile marketing: a synthesis and prognosis. J Interact Mark 23(2): 118–129

Sharaf R, Noureldin A (2007) Sensor integration for satellite-based vehicular navigation using neural networks. IEEE Trans Neural Netw 18(2):589–594

Shankar V, Venkatesh A et al (2010) Mobile marketing in the retailing environment: current insights and future research avenues. J Interact Mark 24(2):111–120

Mobile Object Indexing

George Kollios[1], Vassilis J. Tsotras[2], and Dimitrios Gunopulos[3]
[1]Computer Science Department, Boston University, Boston, MA, USA
[2]University of California-Riverside, Riverside, CA, USA
[3]Department of Computer Science and Engineering, Bourns College of Engineering, The University of California at Riverside, Riverside, CA, USA

Synonyms

Indexing moving objects; Spatio-Temporal Indexing

Definition

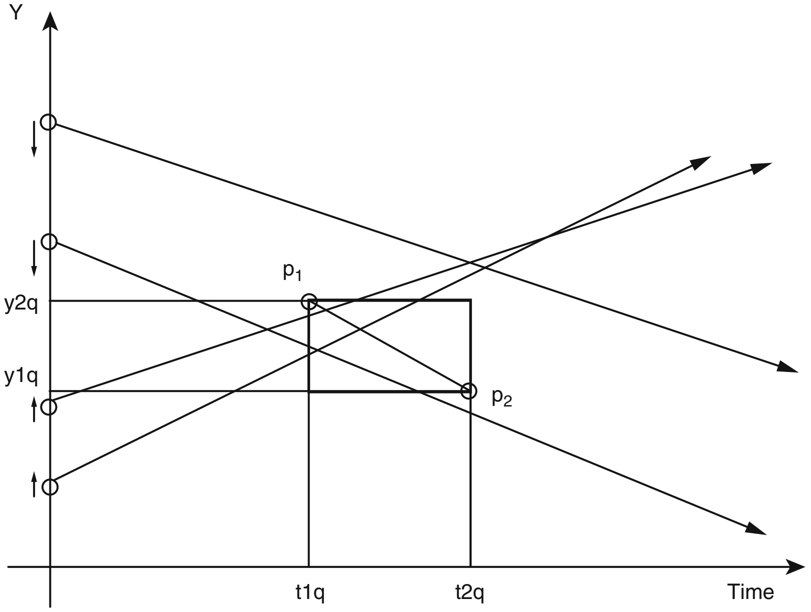
Consider a database that records the position of mobile objects in one and two dimensions, and following Kollios et al. (1999), Saltenis et al. (2000), and Wolfson et al. (1998), assume that an object's movement can be represented (or approximated) with a linear function of time. For each object, the system stores an initial location, a starting time instant and a velocity vector (speed and direction). Therefore, the future positions of the object can be calculated, provided that the characteristics of its motion remain the same. Objects update their motion information when their speed or direction changes. It is assumed that the objects can move inside a finite domain (a line segment in one dimension or a rectangle in two). Furthermore, the system is dynamic, i.e., objects may be deleted or new objects may be inserted.

Let $P(t_0) = [x_0, y_0]$ be the initial position of an object at time t_0. Then, the object starts moving and at time $t > t_0$ its position will be $P(t) = [x(t), y(t)] = [x_0 + v_x(t - t_0), y_0 + v_y(t - t_0)]$, where $V = [v_x, v_y]$ is its velocity vector. An example for the one-dimensional case is shown in Fig. 1.

Range predictive queries in this setting have the following form: "Report the objects located inside the rectangle $[x_{1q}, x_{2q}] \times [y_{1q}, y_{2q}]$ at the time instants between t_{1q} and t_{2q} (where $t_{\text{now}} \leq t_{1q} \leq t_{2q}$), given the current motion information of all objects" (i.e., the *two-dimensional Moving Objects Range (MOR) query* Kollios et al. 1999).

Historical Background

The straightforward approach of representing an object moving on an one-dimensional line is by plotting the trajectories as lines in the time-location (t, y) plane (same for (t, x) plane). The equation describing each line is $y(t) = vt + a$, where v is the slope (velocity in this case) and a is the intercept, which is computed using the motion information (Fig. 1). In this setting, the query is expressed as the two-dimensional interval $[(y_{1q}, y_{2q}), (t_{1q}, t_{2q})]$ and it reports the objects that correspond to the lines intersecting the query rectangle. The space-time approach provides an intuitive representation. Nevertheless, it is problematic since the trajectories correspond to very long lines (going to infinity). Using traditional indexing techniques in this setting tends to reveal many drawbacks. A method that is based



Mobile Object Indexing, Fig. 1 Trajectories and query in (t, y) plane

on this approach partitions the space into disjoint cells and stores those lines that intersect it in each cell (Chon et al. 2002; Tayeb et al. 1998). The shortcoming of these methods is that they introduce replication since each line is copied into the cells that intersect it. Given that lines are typically long, the situation becomes even worse. Moreover, using space partitioning would also result in high update overhead since when an object changes its motion information, it has to be removed from all cells that store its trajectory.

Saltenis et al. (2000) presented another technique to index moving objects. They proposed the time-parametrized R-tree (TPR-tree), which extends the R*-tree. The coordinates of the bounding rectangles in the TPR-tree are functions of time and, intuitively, are capable of following the objects as they move. The position of a moving object is represented by its location at a particular time instant (reference position) and its velocity vector. The bounding intervals employed by the TPR-tree are not always minimum since the storage cost would be excessive. Even though it would be the ideal case (if the bounding intervals were kept always minimum), doing so could deteriorate to enumerating all the enclosed moving points or rectangles. Instead, the TPR-tree uses "conservative" bounding rectangles, which are minimum at some time point, but not at later times. The bounding rectangles may be calculated at load-time (i.e., when the objects are first inserted into the index) or when an update is issued. The TPR-tree assumes a predefined time horizon H, from which all the time instances specified in the queries are drawn. This implies that the user has good knowledge of (or can efficiently estimate) H. The horizon is defined as $H = UI + W$, where UI is the average time interval between two updates and W is the querying window. The insertion algorithm of the R*-tree, which the TPR-tree extends to moving points, aims at minimizing objective functions such as the areas of the bounding rectangles, their margins (perimeters), and the overlap among the bounding rectangles. In the case of the TPR-tree, these functions are time dependent and their evolution in $[t_l, t_l + H]$ is considered where t_l is the time instance when the index is created. Thus, given an objective function $A(t)$, instead of minimizing the objective function, the integral $\int_{t_l}^{t_l+H} A(t)\,dt$ is minimized. An improved version of the TPR-tree, called TPR*-tree, was proposed by Tao et al. (2003). The authors provide a probabilistic model to estimate the number of disk accesses for answering predictive window range queries on moving objects and using this model they provide a hypothetical "optimal" structure for answering these queries. Then, they show that the TPR-tree insertion algorithm leads to structures that are much worse than the optimal one. Based on that, they propose a new insertion algorithm, which, unlike the TPR-tree, considers multiple paths and levels of the index in order to insert a new object. Thus, the TPR*-tree is closer to the optimal structure than the TPR-tree. The authors suggest that although the proposed insertion algorithm is more complex than the TPR-tree insertion algorithm, it creates better trees (MBRs with tighter parametrized extends), which leads to better update performance. In addition, the TPR*-tree employs improved deletion and node splitting algorithms that further improve the performance of the TPR-tree. The STAR-tree, introduced by Procopiuc et al. (2002), is also a time-parametrized structure. It is based upon R-trees, but it does not use the notion of the horizon. Instead, it employs kinetic events to update the index when the bounding boxes start overlapping a lot. If the bounding boxes of the children of a node v overlap considerably, it re-organizes the grand children of v among the children of v. Using geometric approximation techniques developed in Agarwal and Har-Peled (2001), it maintains a time-parametrized rectangle $A_v(t)$, which is a close approximation of $R_v(t)$, the actual minimum bounding rectangle of node v at any time instant t in to the future. It provides a trade-off between the quality of $A_v(t)$ and the complexity of the shape of $A_v(t)$. For linear motion, the trajectories of the vertices of $A_v(t)$ can be represented as polygonal chains. In order to guarantee that $A_v(t)$ is an ε-approximation of $R_v(t)$, trajectories of the corners of $A_v(t)$ need $O(1/\sqrt{\varepsilon})$ vertices. An ε-approximation means that the projection of the $A_v(t)$ on (x, t) or (y, t) planes contains the corresponding projections of $R_v(t)$, but it is not larger than $1 + \varepsilon$ than the extend on the $R_v(t)$ at any time instant.

Finally, another approach to index moving objects is based on the dual transformation that is discussed in detail in the following text. In particular, the *Hough-X representation* of a 2D moving point o is a fourdimensional vector (x, y, v_x, v_y) where x and y is the location of the moving object at the reference time t_{ref} and v_x and v_y is the velocity of the object projected on the x and y axes. Based on this approach, Agarwal et al. (2000) proposed the use of multi-level partition trees (Partition trees group a set of points into disjoint subsets denoted by triangles. A point may lie into many triangles, but it belongs to only one subset.) to index moving objects using the duality transform in order to answer range queries at a specific time instant (i.e., snapshot queries, where $t_{1q} = t_{2q}$). They decompose the motion of the objects on the plane by taking the projections on the (t, x) and (t, y) planes. They construct a primary partition tree T^x to keep the dual points corresponding to the motion projected on the (t, x) plane. Then, at every node v of T^x, they attach a secondary partition T_v^y for the points S_v^y with respect to the (t, y) projection, where S_v is the set of points stored in the primary subtree rooted at v. The total space used by the index is $O(n \log_B n)$, where N is the number of objects, B is the page capacity and n = N/B. The query is answered by decomposing it into two sub-queries, one on each of the two projections, and taking the dual of them, σ^x and σ^y, respectively. The search begins by searching the primary partition T^x for the dual points, with respect to the (t, x) projection, that satisfy the query σ^x. If it finds a triangle associated with a node v of the partition tree T^x that lies completely inside σ^x, then it continues searching in the secondary tree T_v^y and reports all dual points, with respect to (t, y) projection, that satisfy the query σ^y. The query is satisfied if and only if the query in both projections is satisfied. This is true for snapshot range queries. In Agarwal et al. (2000), it is shown that the query takes $O(n^{\frac{1}{2}+\varepsilon} + K/B)$ I/Os (here K is the size of the query result) and that the size of the index can be reduced to $O(n)$ without affecting the asymptotic query time. Furthermore, by using multiple multilevel partition trees, is also shown that the same bounds hold for the window range query. Elbassioni et al. (2003) proposed a technique (MB-index) that partitions the objects along each dimension in the dual space and uses B+-trees in order to index each partition. Assuming a set of N objects moving in d-dimensional space with uniformly distributed and independent velocities and initial positions, they proposed a scheme for selecting the boundaries of the partitions and answering the query, yielding a $O(n^{1-1/3d} * (\sigma \log_B n)^{1/3d} + k)$ average query time using $O(n)$ space ($n = N/B, k = K/B$). The total number of B-trees used is $\sigma 3^d s^{2d-1}$, where $\sigma = \prod_{i=1}^{d} \ln(v_{i,\max}/v_{i,\min})$ and $s = (n/\log_B n)^{1/d}$, where $v_{i,\max}$ and $v_{i,\min}$ are the maximum and minimum velocities in dimension i respectively. The dual transformation has been adapted in Patel et al. (2004), where the advantages over the TPR-trees methods have also been observed. Using the idea in Kollios et al. (1999), trajectories of d-dimensional moving objects are mapped into points in the dual 2d-dimensional space and a PR-quadtree is built to store the 2d-dimensional points. Similarly with Kollios et al. (1999), a different index is used for each of two reference times that change at periodic time intervals. At the end of each period, the old index is removed and a new index with a new reference point is built. Yiu et al. proposed the B^{dual}-tree that uses the dual transformation and maps the dual points to their Hilbert value, thereafter using a B+-tree to index the objects. That was an improvement of the method proposed by Jensen et al. (2004), which indexes the Hilbert value of the locations of the moving objects (without taking into account their velocities).

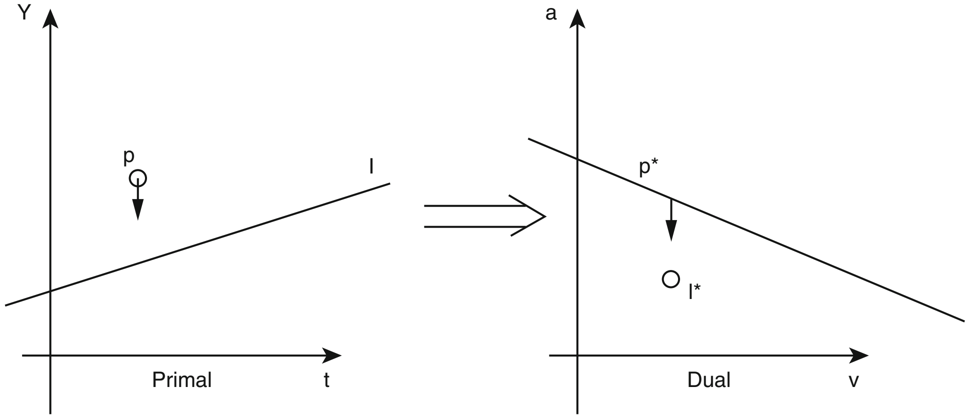
Scientific Fundamentals

The Dual Space-Time Representation

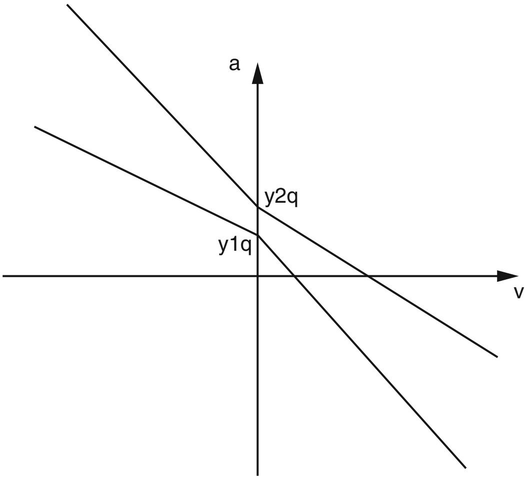
In general, the dual transformation is a method that maps a hyper-plane h from R^d to a point in R^d and vice-versa. In this section, it is briefly described how the problem at hand can be addressed in a more intuitive way by using the dual transform for the one-dimensional case.

Specifically, a line from the primal plane (t, y) is mapped to a point in the dual plane. A class of transforms with similar properties may be used for the mapping. The problem setting parameters determine which one is more useful.

One dual transformation for mapping the line with equation $y(t) = vt + a$ to a point in R^2 is to consider the dual plane where one axis represents the slope of an object's trajectory (i.e., velocity) and the other axis its intercept (Fig. 2). Thus, the dual point is (v, a) (this is called Hough-X transform). Similarly, a point $p = (t, y)$ in the primal space is mapped to line $a(v) = -tv + y$ in the dual space. An important property of the duality transform is that it preserves the above-below relationship. As it is shown in Fig. 2, the dual line of point p is above the dual point l^* of the line l. Based on the above property, it is easy



Mobile Object Indexing, Fig. 2 Hough-X dual transformation: primal plane (*left*), dual plane (*right*)



Mobile Object Indexing, Fig. 3 Query on the Hough-X dual plane

to show that the 1-d query $[(y_{1q}, y_{2q}), (t_{1q}, t_{2q})]$ becomes a polygon in the dual space. Consider a point moving with positive velocity. Then, the trajectory of this point intersects the query if and only if it intersects the segment defined by the points $p_1 = (t_{1q}, y_{2q})$ and $p_2 = (t_{2q}, y_{1q})$ (Fig. 1). Thus, the dual point of the trajectory must be above the dual line p_2^* and below p_1^*. The same idea is used for the negative velocities.

Therefore, using a linear constraint query (Goldstein et al. 1997), the query Q in the dual Hough-X plane (Fig. 3) is expressed in the following way:

$$\text{If } v > 0, \quad \text{then } Q = C_1 \wedge C_2,$$

$$\text{where: } C_1 = a + t_{2q}v \geq y_{1q} \text{ and}$$

$$C_2 = a + t_{1q}v < qy_{2q}$$

$$\text{If } v < 0, \quad \text{then } Q = D_1 \wedge D_2,$$

$$\text{where: } D_1 = a + t_{1q}v \geq y_{1q} \text{ and}$$

$$D_2 = a + t_{2q}v < qy_{2q}$$

By rewriting the equation $y = vt + a$ as $t = \frac{1}{v}y - \frac{a}{v}$, a different dual representation can be used. Now the point in the dual plane has coordinates (b, n), where $b = -\frac{a}{v}$ and $n = \frac{1}{v}$ (Hough-Y). Coordinate b is the point where the line intersects the line $y = 0$ in the primal space. By using this transform, horizontal lines cannot be represented. Similarly, the Hough-X transform cannot represent vertical lines. Therefore, for static objects, only the Hough-X transform can be used.

Indexing in One Dimension

In this section, techniques for the one-dimensional case are presented, i.e., for objects moving on a line segment. There are various reasons for examining the one-dimensional case. First, the problem is simpler and can give good intuition about the various solutions. It is also easier to prove lower bounds and approach optimal solutions for this case. Moreover, it can have practical uses as well. A large highway system can be approximated as a collection of smaller line segments (this is the 1.5 dimensional problem discussed in Kollios et al. 1999), on each of which the one-dimensional methods can be applied.

An (Almost) Optimal and Not Practical Solution

Matousek (1992) gave an almost optimal algorithm for simplex range searching given a static set of points. This main memory algorithm is based on the idea of simplicial partitions.

For a set S of N points, a simplicial partition of S is a set $\{(S_1,\Delta_1), \ldots (S_r, \Delta_r)\}$ where $\{S_1, \ldots, S_r\}$ is a partitioning of S and Δ_i is a triangle that contains all the points in S_i. If $\max_i |S_i| < 2 \min_i |S_i|$, where $|S_i|$ is the cardinality of the set S_i, the partition is balanced. Matousek (1992) shows that, given a set S of N points and a parameter s (where $0 < s < N/2$), it can be constructed in linear time a balanced simplicial partition for S of size $O(s)$ such that any line crosses at most $O(\sqrt{s})$ triangles in the partition.

This construction can be used recursively to construct a partition tree for S. The root of the tree contains the whole set S and a triangle that contains all the points. Then, a balanced simplicial partition of S of size $\sqrt{|S|}$ is found. Each of the children of the root are associated with a set S_i from the simplicial partition and the triangle Δ_i that contains the points in S_i. For each of the S_i 's, simplicial partitions of size $\sqrt{|S_i|}$ are computed and continue to until each leaf contains a constant number of points. The construction time is $O(N \log_2 N)$.

To answer a simplex range query, the procedure starts at the root. Each of the triangles in the simplicial partition at the root checks if (i) it is inside the query region, (ii) it is outside the query region or, (iii) it intersects one of the lines that define the query. In the first case, all points inside the triangle are reported and in the second case the triangle is discarded, while in the third case it continues the recursion on this triangle. The number of triangles that the query can cross is bounded since each line crosses at most $O(|S|^{1/4})$ triangles at the root. The query time is $O(N^{1/2+\varepsilon})$, with the constant factor depending on the choice of ε.

Agarwal et al. (2000) gave an external memory version of static partition trees that answers queries in $O(n^{1/2+\varepsilon} + k)$ I/Os. The structure can become dynamic using a standard technique by Overmars (1983).It can be shown that a point is inserted or deleted in a partition tree in $O(\log_2^2 N)$ I/Os and answer simplex queries in $O(n^{1/2+\varepsilon} + k)$ I/O's. A method that achieves $O(\log_B^2(\frac{N}{B}))$ amortized update overhead is presented in Agarwal et al. (2000).

Using Point Access Methods

Partition trees are not very useful in practice because the query time is $O(n^{1/2+\varepsilon} + k)$ and the hidden constant factor becomes large for small ε. In this section, two different and more practical methods are presented.

There are a large number of access methods that have been proposed to index point data. All of these structures were designed to address *orthogonal* queries, i.e., a query expressed as a multidimensional hyper-rectangle. However, most can be easily modified to address non-orthogonal queries like simplex queries.

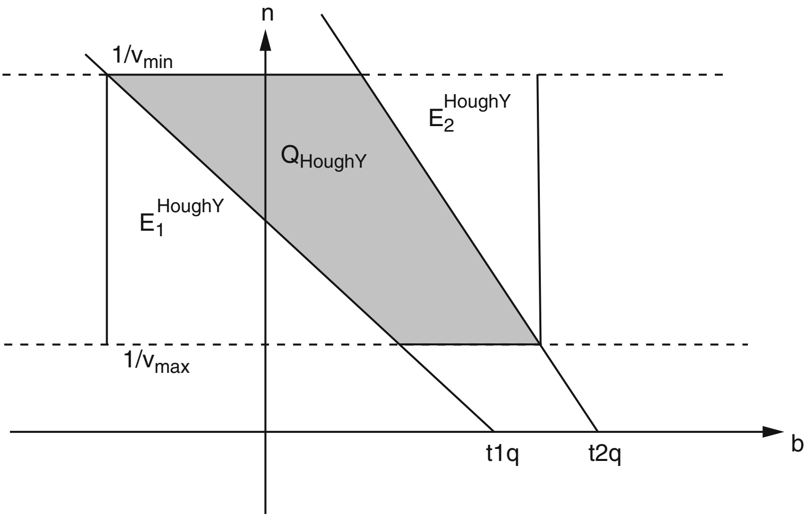
Goldstein et al. (1997) presented an algorithm to answer simplex range queries using R-trees. The idea is to change the search procedure of the tree. In particular, they gave efficient methods to test whether a linear constraint query region and a hyper-rectangle overlap. As mentioned in Goldstein et al. (1997), this method is not only applicable to the R-tree family, but to other access methods as well. This approach can be used to answer the one-dimensional MOR query in the dual Hough-X space.

This approach can be improved by using a characteristic of the Hough-Y dual transformation. In this case, objects have a minimum and maximum speed, $v_{\min}$ and $v_{\max}$, respectively. The $v_{\max}$ constraint is natural in moving object databases that track physical objects. On the other hand, the $v_{\min}$ constraint comes from the fact that the Hough-Y transformation cannot represent static objects. For these objects, the Hough-X transformation is used, as it is explained above. In general, the b coordinate can be computed at different horizontal ($y = y_r$) lines. The query region is described by the intersection of two half-plane queries (Fig. 4). The first line intersects the line $n = \frac{1}{v_{\max}}$ at the point $(t_{1q} - \frac{y_{2q}-y_r}{v_{\max}}, \frac{1}{v_{\max}})$ and the line $n = \frac{1}{v_{\min}}$ at the point $(t_{1q} - \frac{y_{2q}-y_r}{v_{\min}}, \frac{1}{v_{\min}})$. Similarly, the other line that defines the query intersects the horizontal lines at $(t_{2q} - \frac{y_{1q}-y_r}{v_{\max}}, \frac{1}{v_{\max}})$ and $(t_{2q} - \frac{y_{1q}-y_r}{v_{\min}}, \frac{1}{v_{\min}})$.

Since access methods are more efficient for rectangle queries, suppose that the simplex query is approximated with a rectangular one. In Fig. 4, the query approximation rectangle will be $[(t_{1q} - \frac{y_{2q}-y_r}{v_{\min}}, t_{2q} - \frac{y_{1q}-y_r}{v_{\max}}), (\frac{1}{v_{\max}}, \frac{1}{v_{\min}})]$. Note that the query area is enlarged by the area $E = E^{\text{HoughY}} = E_1^{\text{HoughY}} + E_2^{\text{HoughY}}$, which is computed as:

$$E^{\text{HoughY}} = \frac{1}{2}\left(\frac{v_{\max} - v_{\min}}{v_{\min} \cdot v_{\max}}\right)^2 (|\, y_{2q} - y_r \,| + |\, y_{1q} - y_r \,|) \quad (1)$$

The objective is to minimize E since it represents a measure of the extra I/O's that an access method will have to perform for solving a onedimensional MOR query. E is based on both y_r (i.e., where the b coordinate is computed) and the query interval (y_{1q}, y_{2q}), which is unknown. Hence, the method keeps c indices (where c is a small constant) at equidistant y_r's. All c indices contain the same information about the objects, but



Mobile Object Indexing, Fig. 4 Query on the dual Hough-Y plane

use different y_r's. The i-th index stores the b coordinates of the data points using $y_i = \frac{y_{\max}}{c} \cdot i, i = 0, \ldots, c-1$ (see Fig. 5). Conceptually, y_i serves as an "observation" element and its corresponding index stores the data as observed from position y_i. The area between subsequent "observation" elements is called a *sub-terrain*. A given onedimensional MOR query will be forwarded to, and answered exactly by, the index that minimizes E.

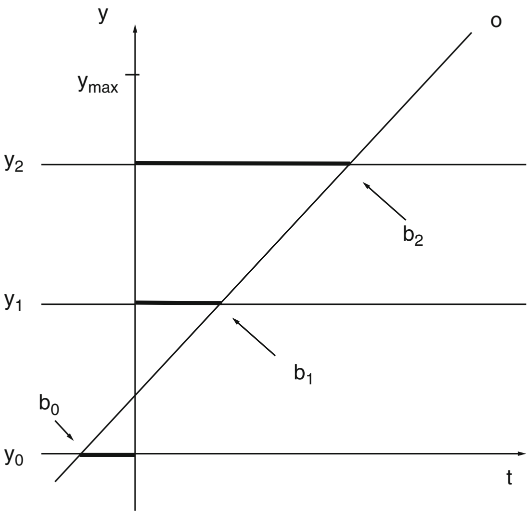
To process a general query interval $[y_{1q}, y_{2q}]$, two cases are considered, depending on whether the query interval covers a sub-terrain:

(i) $y_{2q} - y_{1q} < q\frac{y_{\max}}{c}$: then it can be easily shown that area E is bounded by:

$$E < q\frac{1}{2}\left(\frac{v_{\max} - v_{\min}}{v_{\min} \cdot v_{\max}}\right)^2 \left(\frac{y_{\max}}{c}\right) . \quad (2)$$

The query is processed at the index that minimizes $|y_{2q} - y_r| + |y_{1q} - y_r|$.

(ii) $y_{2q} - y_{1q} > \frac{y_{\max}}{c}$: the query interval contains one or more sub-terrains, which implies that if a query is executed at a single observation index, area E becomes large. To bound E, index each sub-terrain too. Each of the c sub-terrain indices records the time interval when a moving object was in the sub-terrain. Then, the query is decomposed into a collection of smaller sub-queries: one sub-query per sub-terrain fully contained by the original query interval, and one sub-query for each of the original query's endpoints. The sub-queries at the endpoints fall to the case (i) above, thus, they can be answered with bounded E using an appropriate "observation" index. To index the intervals in each sub-terrain, an



Mobile Object Indexing, Fig. 5 Coordinate b as seen from different 'observation' points external memory interval tree can be used which answers a sub-terrain query optimally (i.e., $E = 0$). As a result, the original query can be answered with bounded E. However, interval trees will increase the space consumption of the indexing method.

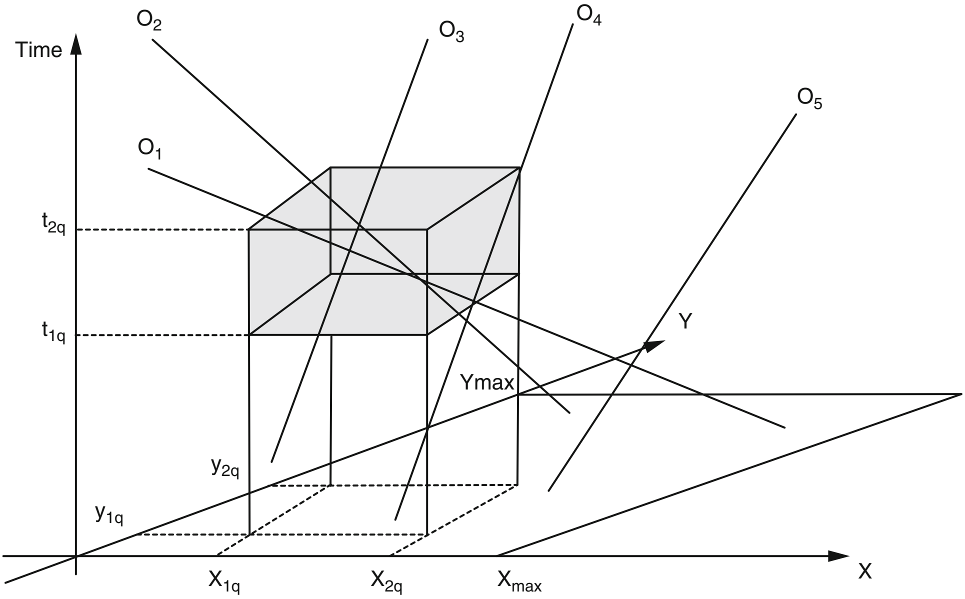
The same approach can be used for the Hough-X transformation, where, instead of different "observation" points, different "observation" times are used. That is, the intercept a can be computed using different vertical lines $t = t_i, i = 0, \ldots, c - 1$. For each different intercept, an index is created. Then, given a query, one of the indices is chosen to answer the query (the one that is constructed for the "observation" time closest to the query time.) However, note that if the query time(s) is far from the "observation" time of an index, then the index will not be very efficient since the query in the Hough-X will not be aligned with the rectangles representing the index and the data pages of this index. Therefore, one problem with this approach comes from the fact that the time in general and the query time in particular, are always increasing. Therefore, an index that is efficient now will become inefficient later. One simple solution to this problem is to create a new index with a newer observation time every T time instants, at the same time removing the index with the oldest observation time (Kollios et al. 1999; Patel et al. 2004). Note that this problem does not exist in the Hough-Y case since the terrain and the query domain do not change with time (or they change very slowly).

Indexing in Two Dimensions

For the two-dimensional problem, trajectories of the moving objects are lines in a three dimensional space (see Fig. 6). Thus, the dual transformation gives a 4-dimensional dual point. Another approach is to split the motion of an object into two independent motions, one in the (t, x) plane and one in the (t, y) plane. Each motion is indexed separately. Next, the procedure used to build the index is presented as well as the algorithm for answering the 2-d query.

Building the Index

The motion in (x, y, t) space is decomposed into two motions, one on the (t, x) and the other on the (t, y) plane. Furthermore, on each projection, the objects are partitioned according to their velocity. Objects with small velocity magnitudes are stored using the Hough-X dual transform, while the rest



Mobile Object Indexing, Fig. 6 Trajectories and query in (x,y,t) space are stored using the Hough-Y transform, i.e., into distinct index structures.

The reason for using different transforms is that motions with small velocities in the Hough-Y approach are mapped into dual points (b, n), having large n coordinates ($n = \frac{1}{v}$). Thus, since few objects have small velocities, by storing the Hough-Y dual points in an index structure such an R*-tree, MBRs with large extents are introduced and the index performance is severely affected. On the other hand, by using a Hough-X index for the small velocities' partition, this effect is eliminated since the Hough-X dual transform maps an object's motion to the (v, a) dual point. The objects are partitioned into slow and fast using a threshold *VT*.

When a dual point is stored in the index responsible for the object's motion in one of the planes, i.e., (t, x) or (t, y), information about the motion in the other plane is also included. Thus, the leaves in both indices for the Hough-Y partition store the record (n_x, b_x, n_y, b_y). Similarly, for the Hough-X partition in both projections, the record (v_x, a_x, v_y, a_y) is stored. In this way, the query can be answered by one of the indices; either the one responsible for the (t, x) or the (t, y) projection.

On a given projection, the dual points (i.e., (n, b) and (v, a)) are indexed using R*-trees (Beckmann et al. 1990). The R*-tree has been modified in order to store points at the leaf level and not degenerated rectangles. Therefore, extra information about the other projection can be stored. An outline of the procedure for building the index follows:

1. Decompose the 2-d motion into two 1-d motions on the (t, x) and (t, y) planes.
2. For each projection, build the corresponding index structure.
   2.1 Partition the objects according to their velocity: Objects with $|v| < VT$ are stored using the Hough-X dual transform, while objects with $|v| \geq VT$ are stored using the Hough-Y dual transform.
   2.2 Motion information about the other projection is also included in each point.

In order to choose one of the two projections and answer the simplex query, the following technique is used.

Answering the Query

The two-dimensional MOR query is mapped to a simplex query in the dual space. The simplex query is the intersection of four 3-d hyperplanes and the projections of the query on the (t, x) and (t, y) planes are wedges, as in the one-dimensional case.

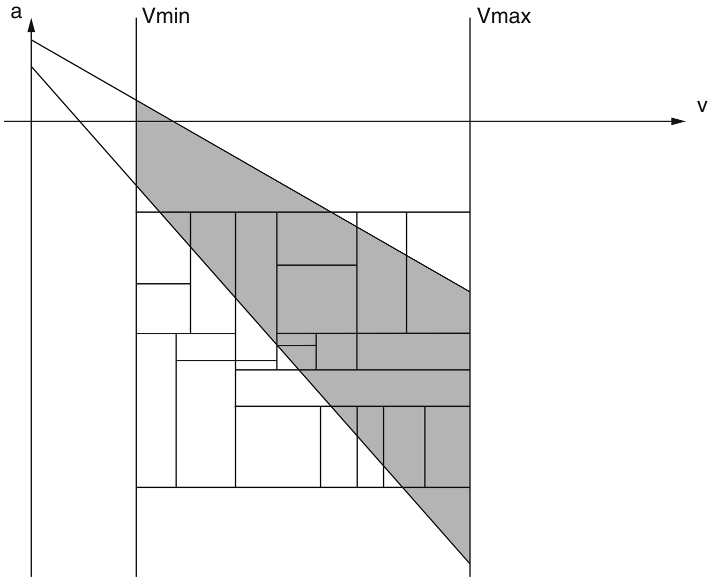
The 2-d query is decomposed into two 1-d queries, one for each projection, and it is answered exactly. Furthermore, on a given projection, the simplex query is processed in both partitions, i.e., Hough-Y and Hough-X.

On the Hough-Y plane the query region is given by the intersection of two half-plane queries, as shown in Fig. 4. Consider the parallel lines $n = \frac{1}{v_{\min}}$ and $n = \frac{1}{v_{\max}}$. Note that a minimum value for $v_{\min}$ is *VT*. As illustrated in Sect. 2, if the simplex query was answered approximately, the query area would be enlarged by $E^{\text{HoughY}} = E_1^{\text{HoughY}} + E_2^{\text{HoughY}}$ (the triangular areas in Fig. 4). Also, let the actual area of the simplex query be Q^{HoughY}. Similarly, on the dual Hough-X plane (Fig. 3), let Q^{HoughX} be the actual area of the query and E^{HoughX} be the enlargement. The algorithm chooses the projection which minimizes the following criterion κ:

$$\kappa = \frac{E^{\text{HoughY}}}{Q^{\text{HoughY}}} + \frac{E^{\text{HoughX}}}{Q^{\text{HoughX}}}. \tag{3}$$

The intuition for this heuristic is that simplex queries in the dual space are not aligned with the MBRs of the underlying index (see Fig. 7). Therefore, the projection where the query is as much aligned with the MBRs as possible is chosen. The empty space, as used in the aforementioned criterion definition, gives an indication of that.

Since the whole motion information is kept in the indices, it can be used to filter out objects that do not satisfy the query. An outline of the algorithm for answering the exact 2-d query is presented below:



Mobile Object Indexing, Fig. 7 Simplex query in dual space, not aligned with MBRs of underlying index

1. Decompose the query into two 1-d queries, for the (t, x) and (t, y) projection.
2. Get the dual query for each projection (i.e., the simplex query).
3. Calculate the criterion κ for each projection and choose the one (say p) that minimizes it.
4. Answer the query by searching the Hough-X and Hough-Y partition using projection p.
5. Put an object in the result set only if it satisfies the query. Use the whole motion information to do the filtering "on the fly".

Key Applications

Location-aware applications such as traffic monitoring, intelligent navigation, and mobile communications management require the storage and retrieval of the locations of continuously moving objects. For example, in a company that manages taxi services, a customer may want to find the taxis that will be in a specific area in the near future. This can be achieved by issuing a query: "Report the taxis that will be in the area around the customer in the next 5 minutes". Similarly, in an air-traffic control system, the locations of the airplanes that are flying close to an airport or a city must be continuously monitored. In that case, a predictive range query can be periodically issued every few seconds. The index will speed up the search and allow for multiple queries to be executed each minute.

Future Directions

There are a number of interesting future problems related to the discussed topic. The dual transformation has been used for objects moving in one and two dimensions. It is interesting to investigate how these methods will be extended to three or higher dimensions. Although the described methods can be applied on higher dimensions, it is not clear if they will have the same efficiency and practicality. Most of the work has been done for linearly moving objects. An interesting future direction is to consider storage and retrieval of non-linear movements. Another problem is to consider moving objects with extents that change over time in addition to their location.

References

Agarwal P, Har-Peled S (2001) Maintaining approximate exten measures of moving points. In: Proceedings of the 12th ACM-SIAM symposium on discrete algorithms, Washington, DC, 7–9 Jan 2001, pp 148–157

Agarwal PK, Arge L, Erickson J (2000) Indexing moving points. In: Proceedings of the 19th ACM symposium on principles of database systems, Dallas, 15–17 May 2000, pp 175–186

Beckmann N, Kriegel H, Schneider R, Seeger B (1998) The R*-tree: an efficient and robust access method for points and rectangles. In: Proceedings of the 1990 ACM SIGMOD, Atlantic City, May 1998, pp 322–331

Chon HD, Agrawal D, Abbadi AE (2002) Query processing for moving objects with space-time grid storage model. In: Proceedings of the 3rd international conference on mobile data management, Singapore, 8–11 Jan 2002, pp 121–126

Elbassioni K, Elmasry A, Kamel I (2003) An efficient indexing scheme for multidimensional moving objects. In: Proceedings of the 9th international conference on ICDT, Siena, 8–10 2003, pp 425–439

Goldstein J, Ramakrishnan R, Shaft U, Yu J (1997) Processing queries by linear constraints. In: Proceedings of the 16th ACM PODS symposium on principles of database systems, Tucson, 13–15 May 1997, pp 257–267

Jensen CS, Lin D, Ooi BC (2004) Query and update efficient B$^+$-tree based indexing of moving objects. In: VLDB, Toronto, 29 Aug–3 Sept 2004, pp 768–779

Kollios G, Gunopulos D, Tsotras V (1999) On indexing mobile objects. In: Proceedings of the 18th ACM symposium on principles of database systems, Philadelphia, 1–3 June 1999, pp 261–272

Matousek J (1992) Efficient partition trees. Discret Comput Geom 8:432–448

Overmars MH (1983) The design of dynamic data strucutures. Vol 156 of LNCS. Springer, Heidelberg

Patel J, Chen Y, Chakka V (2004) STRIPES: an efficient index for predicted trajectories. In: Proceedings of the 2004 ACM SIGMOD, Paris, 13–18 June 2004, pp 637–646

Procopiuc CM, Agarwal PK, Har-Peled S (2002) Star-tree: an efficient selfadjusting index for moving objects. In: Proceedings of the 4th workshop on algorithm engineering and experiments, San Francisco, 4–5 Jan 2002, pp 178–193

Saltenis S, Jensen C, Leutenegger S, Lopez MA (2000) Indexing the positions of continuously moving objects. In: Proceedings of the 2000 ACM SIGMOD, Dallas, 16–18 May 2000, pp 331–342

Tao Y, Papadias D, Sun J (2003) The TPR*-tree: an optimized spatiotemporal access method for predictive queries. In: Proceedings of the 29th international conference on very large data bases, Berlin, 9–12 Sept 2003, pp 790–801

Tayeb J, Olusoy O, Wolfson O (1998) A quadtree-based dynamic attribute indexing method. Comput J 41(3):185–200

Wolfson O, Xu B, Chamberlain S, Jiang L (1998) Moving objects databases: issues and solutions. In: Proceedings of the 10th international conference on scientific and statistical database management, Capri, 1–3 July 1998, pp 111–122

Yiu ML, Tao Y, Mamoulis N (To appear) The B^{dual}-tree: indexing moving objects by space-filling curves in the dual space. VLDB J

Mobile Objects Databases

Harvey J. Miller
Department of Geography, University of Utah,
Salt Lake City, UT, USA

Synonyms

Moving objects database

Definition

Mobile objects databases (MOD) store data about entities that can change their geometry frequently, include changes in location, sizes and shapes.

Main Text

Most database management systems (DBMS), even spatio-temporal DBMS, are not well equipped to handle data on objects that change their geometry frequently, in some cases continuously. In DBMS, data is assumed to be constant unless it is explicitly modified. Using standard database update techniques to update the geometry of a moving object is too expensive computationally. Traditional query languages such as structured query language (SQL) are not well-equipped to handle the spatio-temporal queries required by a MOD. These include: “Retrieve all objects that will intersect a region within the next 4 minutes.” “Retrieve all objects that will come within 3 kilometers of each other and the time when this will occur.” Finally, although the geometry of a moving object is changing continuously in some cases, digital technology for recording these geometries (such as location-aware technologies) as well as technologies for storing these data have finite resolution: it can only record and store position at discrete moments. Also, each one of these positions will have a degree of imprecision. Therefore, the position of an object at any given moment in time will have a degree of uncertainty.

MODs handle all three components of data that change their geometry frequently, namely: (i) rapid database updating; (ii) mobile object queries; (iii) uncertainty in locational tracking.

Although MODs theoretically handle entities that change their location, size and shapes, many applications focus on rigid objects that only change their positions such as vehicles.

Cross-References

- ▸ Geographic Knowledge Discovery
- ▸ Location-Aware Technologies

Recommended Reading

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

Wolfson O, Xu B, Chamberlain S, Jiang L (1998) Moving objects databases: issues and solutions. In: Proceedings of the 10th international conference on scientific and statistical database management (SSDBM 1998), pp 111–122

Mobile P2P Databases

Yan Luo[1] and Ouri Wolfson[2]
[1]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[2]Mobile Information Systems Center (MOBIS), The University of Illinois at Chicago, Chicago, IL, USA

Definition

A mobile peer-to-peer (P2P) database is a database that is stored in the peers of a mobile P2P network. The network is composed by a finite set of mobile peers that communicate with each other via short-range wireless protocols, such as IEEE 802.11, Bluetooth, Zigbee, or ultra wide band (UWB). These protocols provide broadband (typically tens of Mbps) but short-range (typically 10–100 m) wireless communication. On each mobile peer there is a local database that stores and manages a collection of data items or reports. A report is a set of values sensed or entered by the user at a particular time, or otherwise obtained by a mobile peer. Often a report describes a physical resource, such as an available parking slot. All the local databases maintained by the mobile peers form the mobile P2P database. The peers communicate reports and queries to neighbors directly, and the reports and queries propagate by transitive multi-hop transmissions. Figure 1 below illustrates the definition.

In contrast to the assumptions made in the literature on mobile ad hoc networks (MANETs) and mesh networks, a peer may not know the identities of other peers in the network and the data they store. Thus, routing in the traditional MANET sense is not a common operation in mobile P2P databases.

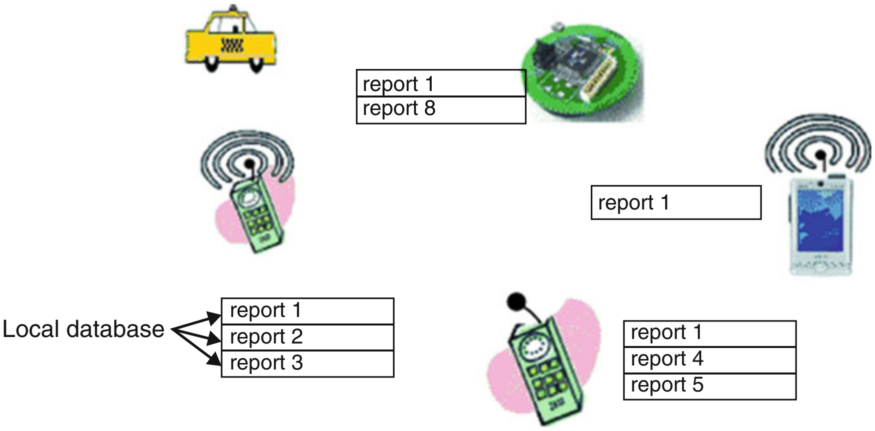
Mobile P2P databases enable matchmaking or resource discovery services in many application domains, including social networks, transportation, mobile electronic commerce, emergency response, and homeland security.

Communication is often restricted by bandwidth and power constraints on the mobile peers. Furthermore, often reports need to be stored and later forwarded, thus memory constraints on the mobile devices constitute a problem as well. Thus, careful and efficient utilization of scarce peer resources (specifically bandwidth, power, and memory) are an important challenge for mobile P2P databases.

Historical Background

Traditionally search databases have been implemented by a centralized architecture. Google is preeminent example of such architecture. However, mobile P2P databases have

Research supported by NASA Award NNA06AA25C, and NSF Grants OII-0611017, DGE-0549489, 0513736, and 0326284.



Mobile P$_2$P Databases, Fig. 1 A mobile P2P database

several advantages over centralized ones. First, because short-range wireless networks utilize the unlicensed spectrum, communication to the mobile P2P database is free; there is also no cost involved in setting up and maintaining the fixed infrastructure database. Second, mobile P2P databases can be used for search in emergency, disaster, and other situations where the infrastructure is destroyed or unavailable. Third, mobile P2P databases are harder to mine for private information, and fourth, they are more reliable in the sense that failure of the central site will not render the system unavailable. Fifth, mobile P2P databases can withstand the high update rates that will be generated when representing temporary physical resources (e.g. the available parking slots), or continuous phenomena, such as the location of moving objects. The disadvantage of mobile P2P databases is that they do not provide answer guarantees. In other words, although the answer to a query exists in the database, due to mobility and lack of global coordination, the mobile P2P database may not find it.

The concept of mobile P2P database is proposed for searching local information, particularly information of a temporary nature, i.e., valid for a short duration of time (Xu and Wolfson 2004).

Currently, there are quite a few experimental projects in mobile P2P databases. These can be roughly classified into pedestrians and vehicular projects. Vehicular projects deal with high mobility and high communication topology change-rates, whereas pedestrian projects have a strong concern with power issues. The following are several active experimental mobile P2P database projects for pedestrians and vehicles:

Pedestrians Projects

- **7DS** – Columbia University (Papadopouli and Schulzrinne 2001)
  - http://www.cs.unc.edu/~maria/7ds/
  - Focuses on accessing web pages in environments where only some peers have access to the fixed infrastructure.
- **iClouds** – Darmstadt University (Heinemann et al. 2003)
  - http://iclouds.tk.informatik.tu-darmstadt.de/
  - Focuses on the provision of incentives to brokers (intermediaries) to participate in the mobile P2P database.
- **MoGATU** – University of Maryland, Baltimore County (Perich 2004)
  - http://mogatu.umbc.edu/
  - Focuses on the processing of complex data management operations, such as joins, in a collaborative fashion.
- **PeopleNet** – National University of Singapore (Motani et al. 2005)
  - http://www.ece.nus.edu.sg/research/projects/abstract.asp?Prj=101
  - Proposes the concept of information bazaars, each of which specializes in a

particular type of information; reports and queries are propagated to the appropriate bazaar by the fixed infrastructure.

- **MoB** – University of Wisconsin and Cambridge University (Chakravorty et al. 2005)
  - ○ http://www.cs.wisc.edu/~suman/projects/agora/
  - ○ Focuses on incentives and the sharing among peers of virtual information resources such as bandwidth.
- **Mobi-Dik** – University of Illinois at Chicago (Xu and Wolfson 2004; Wolfson et al. 2006)
  - ○ http://www.cs.uic.edu/~wolfson/html/p2p.html
  - ○ Focuses on information representing physical resources, and proposes stateless algorithms for query processing, with particular concerns for power, bandwidth, and memory constraints.

Vehicular Projects

- **CarTALK 2000** – A European project
  - ○ http://www.cartalk2000.net/
  - ○ Develops a cooperative driver assistance system based upon inter-vehicle communication and mobile P2P databases via self-organizing vehicular ad hoc networks.
- **FleetNet** – Internet on the Road Project (Franz et al. 2001)
  - ○ http://www.ccrle.nec.de/Projects/fleetnet.htm
  - ○ Develops a wireless multi-hop ad hoc network for intervehicle communication to improve the driver's and passengers' safety and comfort. A data dissemination method called "contention-based forwarding" (CBF) is proposed in which the next hop in the forwarding process is selected through a distributed contention mechanism based on the current positions of neighbors.
- **VII** – Vehicle Infrastructure Integration, a US DOT project
  - ○ http://www.its.dot.gov/vii/
  - ○ The objective of the project is to deploy advanced vehicle-to-vehicle (using the mobile P2P paradigm) and vehicle-toinfrastructure communications that could keep vehicles from leaving the road and enhance their safe movement through intersections.
- **Grassroots** – Rutgers University (Goel et al. 2003)
  - ○ http://paul.rutgers.edu/~gsamir/dataspace/grassroots.html
  - ○ Develops an environment in which each vehicle contributes a small piece of traffic information to the network based on the P2P paradigm, and each vehicle aggregates pieces of the information into a useful picture of the local traffic information.

Scientific Fundamentals

There are two main paradigms for answering queries in mobile P2P databases, one is report pulling and the other one is report pushing.

Report pulling means that a mobile peer makes an explicit request for the report it is interested in receiving, and the whole network is flooded with queries, the interested report will be pulled from the mobile peers that have them. Report pulling is widely used in resource discovery, such as route discovery in mobile ad hoc networks and file discovery by query flooding in wired P2P networks like Gnutella. Flooding in a wireless network is in fact relatively efficient as compared to wired networks because of wireless multicast advantage.

Another possible approach for data dissemination is report pushing. Report pushing is the dual problem of report pulling; reports are flooded, and consumed by peers whose query is answered by received reports. So far there exist mechanisms to broadcast information in the complete network, or in a specific geographic area (geocast), apart from to any one specific mobile node (unicast/mobile ad hoc routing) or any one arbitrary node (anycast). Report pushing paradigm can be further divided into stateful methods and stateless methods. Most stateful methods are topology-based, i.e., they impose a structure of links in the network, and maintain states of data dissemination. PStree, which organizes the peers as a tree, is an example of topology-based methods.

Another group of stateful methods is the cluster- or hierarchy-based method, such as Visvanathan et al. (2005), in which moving peers are grouped into some clusters or hierarchies and the cluster heads are randomly selected. Reports are disseminated through the network in a cluster or hierarchy manner, which means that reports are first disseminated to every cluster head and each cluster head then broadcasts the reports to the member peers in its group. Although cluster- or hierarchy-based methods can minimize the energy dissipation in moving peers, these methods will fail or cost more energy in highly mobile environments since they have to maintain a hierarchy structure and frequently reselect cluster heads.

Another stateful paradigm consists of location-based methods (see Mauve et al. 2001). In location-based methods, each moving peer knows the location of itself and its neighbors through some localization techniques, such as GPS or atomic multilateration (see Mauve et al. 2001).

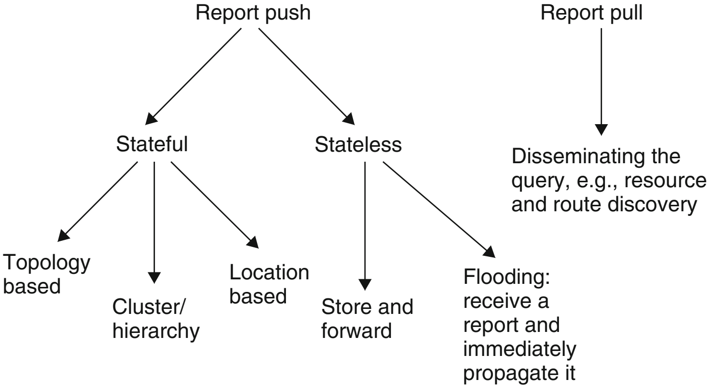
The simplest location-based data dissemination is greedy forwarding, in which each moving peer transmits a report to a neighbor that is closer to the destination than it is. However, greedy forwarding can fail in some cases, such as when a report is stuck in local minima, which means that the report stays in a mobile peer whose neighbors are all further from the destination. Therefore, some recovery strategies are proposed, such as greedy perimeter stateless routing (GPSR) (Karp and Kung 2000). Other location-based methods, such as geographic adaptive fidelity (GAF) (Xu et al. 2001) and geographical and energy aware routing (GEAR) (Yu et al. 2001), take advantage of knowledge about both location and energy to disseminate information and resources more efficiently.

In stateless methods, the most basic and simplest one is the flooding-based method, such as Oliveira et al. (2005). In flooding-based methods, mobile peers simply propagate received reports to all neighboring mobile peers until the destination or maximum a hop is reached. Each report is propagated as soon as is received. Flooding-based methods have many advantages, such as no state maintenance, no route discovery, and easy deployment. However, they inherently cannot overcome several problems, such as implosion, overlap, and resource blindness. Therefore, other stateless methods are proposed, such as gossiping-based methods and negotiation-based methods.

Gossiping-based methods, such as Datta et al. (2004), improve flooding-based methods by transmitting received reports to a randomly selected neighbor or to the neighbors that are interested in the particular content. The advantages of gossiping-based methods include reducing the implosion and lowering the system overhead. However, the cost of determining the particular interests of each moving peer can be huge and transmitting reports to a randomly selected neighbor can still cause the implosion problem and waste peers' memory, bandwidth and energy. Furthermore, dissemination and, thus, performance are reduced compared to pure flooding.

Negotiation-based methods solve the implosion and overlap problem by transmitting first the IDs of reports; the reports themselves are transmitted only when requested (see Kulik et al. 2002). Thus, some extra data transmission is involved, which costs more memory, bandwidth, and energy. In addition, in negotiation-based methods, moving peers have to generate metadata or a signature for every report so that negotiation can be carried out, which will increase the system overhead and decrease the efficiency.

Another important stateless paradigm for data dissemination in mobile P2P networks is store-and-forward, such as Wolfson et al. (2006), which to ranks all the reports in a peer's database in terms of their relevance or expected utility, and then the reports are communicated and saved in the order of their relevance. Alternatively, the reports requested and communicated are the ones with the relevance above a certain threshold. The notion of relevance quantifies the importance or the expected utility of a report to a peer at a particular time and at a particular location. Other store-and-forward methods include PeopleNet (Motani et al. 2005) and 7DS (Papadopouli and Schulzrinne 2001).


Report push
Report pull
Stateful
Stateless
Disseminating the query, e.g., resource and route discovery
Topology based
Cluster/ hierarchy
Location based
Store and forward
Flooding: receive a report and immediately propagate it

Mobile P$_2$P Databases, Fig. 2 Query answering methods in mobile P2P databases

In summary, the paradigms for data dissemination in mobile P2P databases are summarized in Fig. 2 below.

Key Applications

Mobile P2P databases provide mobile users a search engine for transient and highly dynamic information in a local geospatial environment. Mobile P2P databases employ a unified model for both the cellular infrastructure and the mobile ad hoc environments. When the infrastructure is available, it can be augmented by the mobile P2P database approach.

Consider a mobile P2P database platform, i.e., a set of software services for data management in a mobile P2P environment; it is similar to a regular database management system, but geared to mobile P2P interactions. Such a platform will enable quick building of matchmaking or resource discovery services in many application domains, including social networks, emergency response, homeland security, military, airport applications, mobile e-commerce, and transportation.

Social Networks

In a large professional, political, or social gathering, mobile P2P databases are useful to automatically facilitate a face-to-face meeting based on matching profiles. For example, in a professional gathering, mobile P2P databases enable attendees to specify queries (interest profiles) and resource descriptions (expertise) to facilitate conversations, when mutual interest is detected. This opportunistic matchmaking can greatly enhance the value of networking events allowing users to connect with targeted, interested parties without a priori knowledge of their name, title, phone number, or other personal information. A face-to-face meeting can be setup by including in the resource description the identification information of the resource (person), such as cell-phone number, name, screen name, picture, physical description, etc. This information may be used together with the (possibly imprecise) location to help set up the face-to-face meeting. Thus, the individual's profile that is stored in mobile P2P databases will serve as a "wearable web-site". Similarly, mobile P2P databases can facilitate face-to-face meetings in singles matchmaking.

Emergency Response, Homeland Security, and the Military

Mobile P2P databases offer the capability to extend decision-making and coordination capability. This finds applications in emergency environments, an area of particular concern to the government trying to find technologies that can be exploited to support the more than eight million first responders in US homeland security. Consider workers in disaster areas, soldiers and military personnel operating in environments where the wireless fixed infrastructure is significantly degraded or non-existent. They would welcome a capability that lets them automatically propagate messages, pictures, or resource information to other workers, based on matching profiles, security, and attribute values rather than node-id. As mobile users involved in an emergency response naturally cluster around the location of interest, a self-forming, high-bandwidth network that allows secure point-to-point or point-to-multipoint communication without the need of potentially compromised infrastructure could be of great benefit. For instance, a picture of a wanted person could be propagated to all those involved in a targeted search at the scene.

Consider a related emergency response application. Scientists are developing cockroach-sized robots or sensors that are carried by real cockroaches, which are able to search victims in exploded or earthquake-damaged buildings. These robots or sensors are equipped with radio transmitters. When a robot discovers a victim by sensing carbon dioxide, it may not have the transmission power to reach the outside rescuers; it can use local data dissemination to propagate the information to human rescuers outside the rubble. Sensors can also be installed on wild animals for endangered species assistance. A sensor monitors its carrier's health condition, and it disseminates a report when an emergency symptom is detected.

Airport Applications

Airports provide several different opportunities for the use of mobile P2P databases. From the point of view of commerce, airports have stores and kiosks where merchandise is sold similarly to a mall. Imagine arriving at a large airport and realizing you do not have the computer power cord you need for your presentation. Mobile P2P databases will enable a user to search for the needed product – just like in a mall. Merchants can similarly provide their location information and offer promotional incentives to passengers.

Mobile P2P databases can also be used by airport personnel to coordinate their activities. This is especially important when there is a communication failure due an emergency that degrades the infrastructure. Like the case of early responders, airport personnel can continue to coordinate their activities through the use of the mobile P2P network that is available even though the infrastructure is not functioning. Another potential opportunity that will benefit both the consumer and the airport operations is the dissemination of real-time information regarding flight changes, delays, queue length, parking information, special security alerts and procedures, and baggage information. This can augment the present audio announcements that often cannot be heard in nearby restaurants, stores, or restrooms, and the limited, expensive displays.

Mobile E-commerce

Consider short-range wireless broadcast and mobile P2P dissemination of a merchant's sale and inventory information. It will enable a customer (whose cell phone is mobile P2P databases enabled) that enters a mall to locate a desired product at the best price. When a significant percentage of people have mobile devices that can query retail data, merchants will be motivated to provide inventory/sale/coupons information electronically to nearby potential customers. The information will be provided and disseminated in a P2P fashion (in, say, a mall or airport) by the mobile P2P databases software.

Transportation Safety and Efficiency

Mobile P2P databases software can improve safety and mobility by enabling travelers to cooperate intelligently and automatically. A vehicle will be able to automatically and transitively communicate to trailing vehicles its "slow speed" message when it encounters an accident, congestion, or dangerous road surface conditions. This will allow other drivers to make decisions, such as finding alternative roads. Also, early warning messages may allow a following vehicle to anticipate sudden braking or a malfunctioning brake light, thus preventing pile-ups in some situations. Similarly, other resource information, such as ridesharing opportunities, transfer protection (transfer bus requested to wait for passengers), will be propagated transitively, improving the efficiency of the transportation system.

Inefficiencies in the transportation system result in excessive environmental pollution, fuel consumption, risk to public safety, and congestion. Ridesharing (i.e., vehicles carrying more than one person, either publicly provided such as transit, a taxi, or a vanpool, or prearranged rides in a privately owned vehicle) and car sharing (i.e., a program that allows registered users to borrow a car on an hourly basis from fixed locations) have the potential to alleviate these problems. Currently the matchmaking required in ridesharing is performed offline. However, the success of ridesharing will depend largely on the efficient identification and matching of riders/drivers to vehicles in real time in a local environment, which is where the benefit of our technology lies, providing information that is simultaneously relevant in time, location, and interest. Mobile P2P databases incorporated in navigational devices and PDA's can be used to disseminate to other devices and PDA's information about relevant resources, such as ridesharing partners, free parking slots, and available taxicabs or taxicab customers.

Future Directions

There are many challenges and directions for the future research in mobile P2P databases in mobile P2P networks:

1. **Prolong network lifetime** How to maximize the network life is a common but difficult problem in mobile P2P databases. Currently, some approaches as discussed above, e.g., ranking and cluster-based-methods, are proposed to address this problem and prolong the lifetime of sensor networks, mobile ad hoc networks, and mobile P2P databases. The future research question is how to employ the redundancy of networks and the density of peers in order to maximally extend the network lifetime.
2. **Sparse networks** Currently, the performance of many algorithms and systems heavily depends on the density of peers in mobile P2P networks. They do not perform very well if the network is sparse. Therefore, understanding how to design and develop mobile P2P databases for sparse networks is an important and difficult challenge. Recent work that heads in this direction includes delay tolerant networks, store and forward flooding, and mobile peers whose sole function is to provide connectivity.
3. **Rapid topology changes** Another challenge for designing and developing mobile P2P databases is high mobility of peers. This poses problems to mobile P2P databases, e.g., how to efficiently disseminate queries and answers, and how to reconfigure rapidly when the topology of networks changes frequently. Stateless approaches seem most suitable to address these problems.
4. **Emergent global behavior from local knowledge** Mobile P2P databases can be treated as a special type of distributed system. Each peer maintains a local database and all the local databases form the virtual mobile P2P database. Therefore, peers can only use the local knowledge to predict or affect the global behavior of the whole mobile P2P database. The future research direction will be how to employ the local knowledge and propose the adaptive local algorithms to direct or affect the global behavior of mobile P2P databases.
5. **(Self-) localization techniques** Location-based approaches are more and more popular and necessary, and location information of peers is useful for efficiently storing and managing information. However, selflocalization techniques are still not efficient and effective enough due to the limitation of peers or localization techniques. For example, GPS is not available indoors and the accuracy of GPS is not enough for some mobile P2P databases. Therefore, creating efficient and effective selflocalization technique for mobile P2P databases is an important research direction.
6. **Integration of mobile P2P databases and infrastructure** As discussed above, mobile P2P databases do not guarantee answer completeness. In this sense, the integration with an available infrastructure, such as the Internet or a cellular network may improve performance significantly. This integration has two aspects. First, using the communication infrastructure in order to process queries more efficiently in the mobile P2P database; and second, using data on the fixed network in order to provide

better and more answers to a query. The seamless integration of mobile P2P databases and infrastructure databases introduces important research challenges. Recent work on data integration in the database community can provide a starting point for is research.

7. **Specialized queries** Existing mobile P2P query processing methods deal with simple queries, e.g. selections; each query is satisfied by one or more reports. However, in many application classes one may be interested in more sophisticated queries. For example, in mobile electronic commerce a user may be interested in the minimum gas price within the next 30 miles on the highway. Processing of such queries may present interesting optimization opportunities.
8. **Mathematical modeling of data dissemination** Many query processing and data dissemination algorithms may benefit from a mathematical model of data propagation. For example, a formula giving the number *n* of mobile peers having a report that was generated at time *t* at location *l* would be very useful in ranking of such a report. The number *n* is a function of the density of mobile peers, motion speed, bandwidth and memory availability at the peers, memory management, etc. Related work done in epidemiology about the spread of infectious diseases would be a good starting point for this research. Results in random graphs may also be applicable.

Other important research directions include incentives for broker participation in query processing, and transactions/atomicity/recovery issues in databases distributed over mobile peers.

Recommended Reading

Chakravorty R, Agarwal S, Banerjee S, Pratt I (2005) MoB: a mobile bazaar for wide-area wireless services. In: International conference mobile computing and networking (MobiCom'05), Cologne

Datta A, Quarteroni S, Aberer K (2004) Autonomous gossiping: a self organizing epidemic algorithm for selective information dissemination in wireless mobile ad hoc networks. In: The International conference on semantics of a networked world

Goel S, Imielinski T, Ozbay K, Nath B (2003) Grassroots: a scalable and robust information architecture. Technical report DCS-TR-523, Department of Computer Science, Rutgers University

Franz W, Eberhardt R, Luckenbach T (2001) FleetNet – Internet on the road. In: The 8th world congress on intelligent transportation systems, Sydney

Heinemann A, Kangasharju J, Lyardet F, Mühlhäuser M (2003) iClouds – P2P information sharing in mobile environments. In: International conference on parallel and distributed computing (Euro-Par 2003), Klagenfurt

Karp B, Kung HT (2000) GPSR: greedy perimeter stateless routing for wireless sensor networks. In: The 6th annual ACM/IEEE international conference on mobile computing and networking (MobiCom'00), Cologne, pp 243–254

Kulik J, Heinzelman W, Balakrishnan H (2002) Negotiation-based protocols for disseminating information in wireless sensor networks. Wirel Netw 8:169–185

Mauve M, Widmer A, Hartenstein H (2001) A survey on position-based routing in mobile ad hoc networks. IEEE Netw 15(6):30–39

Motani M, Srinivasan V, Nuggehalli P (2005) PeopleNet: engineering a wireless virtual social network. In: International conference mobile computing and networking (MobiCom'05), Cologne

Oliveira R, Bernardo L, Pinto P (2005) Flooding techniques for resource discovery on high mobility MANETs. In: Workshop on wireless ad hoc networks

Papadopouli M, Schulzrinne H (2001) Design and implementation of a P2P data dissemination and prefetching tool for mobile users. In: First NY Metro Area Networking Workshop, IBM TJ Watson Research Center, Hawthorne

Perich F (2004) On P2P data management in pervasive computing environments. PhD. thesis, UMBC

Visvanathan A, Youn JH, Deogun J (2005) Hierarchical data dissemination scheme for large scale sensor networks. In: IEEE international conference on communications (ICC'05), pp 3030–3036

Wolfson O, Xu B, Yin HB, Cao H (2006) Search-and-discover in mobile P2P network databases. In Proceedings of the 26th IEEE international conference on distributed computing systems (ICDCS'06), Lisbon

Xu B, Wolfson O (2004) Data management in mobile P2P networks. In: Proceedings of the 2nd international workshop on databases, information systems, and P2P computing (DBISP2P'04), Toronto. Lecture notes in computer science. Springer

Xu Y, Heidemann J, Estrin D (2001) Geography informed energy conservation for ad hoc routing. In: The ACM international conference on mobile computing and networking, Rome, pp 70–84

Yu Y, Govindan R, Estrin D (2001) Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report UCLA/CSD-TR-01-0023, UCLA

Mobile Population

► Population Distribution During the Day

Mobile Recommendation

► Location-Based Recommendation Systems

Mobile Recommender Systems in Tourism

► Mobile Travel Tour Recommendation

Mobile Robotics

► Indoor Positioning, Bayesian Methods

Mobile Sequential Recommendation

Yong Ge
The University of Arizona, Tucson, AZ, USA

Synonyms

Mobile sequential recommendation (MRS); Potential travel distance (PTD)

Definition

Recommender systems (Adomavicius and Tuzhilin 2005) address the information-overloaded problem by identifying user interests and providing personalized suggestions. In general, there are three ways to develop recommender systems. The first one is content based (Mooney and Roy 1999). It suggests items which are similar to those a given user has liked in the past. The second way is based on collaborative filtering. In other words, recommendations are made according to the tastes of other users that are similar to the target user. Finally, a third way is to combine the above and have a hybrid solution (Pazzani 1999). However, the development of personalized recommender systems in mobile and pervasive environments is much more challenging than developing recommender systems from traditional domains due to the complexity of spatial data and intrinsic spatiotemporal relationships, the unclear roles of context-aware information, and the increasing availability of environment sensing capabilities. We exploit the knowledge extracted from location traces and develop a mobile recommender system based on business-success metrics instead of predictive performance measures based on user ratings. Indeed, the key idea is to leverage the business knowledge from the historical data of successful taxi drivers for helping other taxi drivers improve their business performance. Along this line, we provide a pilot feasibility study of extracting business-success knowledge from location traces by taxi drivers and exploiting this business information for guiding taxis' driving routes. Specifically, we first extract a group of successful taxi drivers based on their past performances in terms of revenue per energy use. Then, we can cluster the pickup points of these taxi drivers for a certain time period. The centroids of these clusters can be used as the recommended pickup points with a certain probability of success for new taxi drivers in these areas. This problem can be formally defined as a mobile sequential recommendation problem, which recommends sequential pickup points for a taxi driver to maximize his/her business success.

Historical Background

Recommender systems in the mobile environments have been studied before (Abowd et al. 1997; Averjanova et al. 2008; Cena et al. 2006; Cheverst et al. 2000; Miller et al. 2003; Tveit 2001; van der Heijden et al. 2005). For instance, the work in Abowd et al. (1997) and Cena et al. (2006) targets the development of mobile tourist

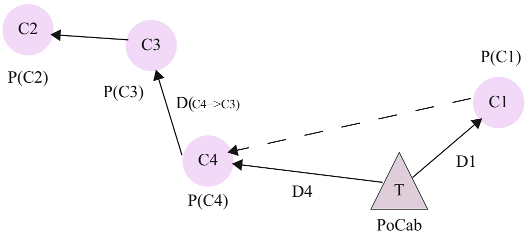
The MSR Problem

Given: A set of potential pickup points $\mathcal{C}$ with $|\mathcal{C}| = N$, a probability set $\mathcal{P} = \{P(C_1), P(C_2), \cdots, P(C_N)\}$, a directed sequence set $\overrightarrow{\mathcal{R}}$ with $|\overrightarrow{\mathcal{R}}| = M$, and the current position ($PoCab$) of a cab driver, who needs the service.

Objective: Recommending an optimal driving route $\overrightarrow{\mathcal{R}}$ ($\overrightarrow{\mathcal{R}} \in \overrightarrow{\mathcal{R}}$). The goal is to minimize the PTD:

$$\min_{\overrightarrow{R_i} \in \overrightarrow{\mathcal{R}}} \mathcal{F}(PoCab, \overrightarrow{R_i}, \mathcal{P}_{\overrightarrow{R_i}}) \quad (1)$$

guides. Also, Heijden et al. have discussed some technological opportunities associated with mobile recommender systems (van der Heijden et al. 2005). In addition, Averjanova et al. have developed a map-based mobile recommender system that can provide users with some personalized recommendations (Averjanova et al. 2008). However, this prior work is mostly based on user ratings and is only exploratory in nature, and the problem of leveraging unique features distinguishing mobile recommender systems remains pretty much open.



Mobile Sequential Recommendation, Fig. 1 An illustration example

Scientific Fundamentals

Consider a scenario that a large number of GPS traces of taxi drivers have been collected for a period of time. In this collection of location traces, we also have the information when a cab is empty or occupied. In this data set, it is possible to first identify a group of taxi drivers who are very successful in business. Then, we can cluster the pickup points of these taxi drivers for a certain time period. The centroids of these clusters can be used as the recommended pickup points with a certain probability of success for new taxi drivers in these areas. Then, a mobile sequential recommendation problem can be formulated as follows.

Assume that a set of N potential pickup points, $\mathcal{C}$=$\{C_1, C_2, \cdots, C_N\}$, is available. Also, the estimated probability that a pickup event could happen at each pickup point is known as $P(C_i)$, where $P(C_i)(i = 1, \cdots, N)$ is assumed to be independently distributed. Let $\mathcal{P} = \{P(C_1), P(C_2), \cdots, P(C_N)\}$ denote the probability set. In addition, let $\overrightarrow{\mathcal{R}} = \{\overrightarrow{R_1}, \overrightarrow{R_2}, \cdots, \overrightarrow{R_M}\}$ be the set of all the directed sequences (potential driving routes) generated from $\mathcal{C}$ and $|\overrightarrow{\mathcal{R}}| = M$ is the size of $\overrightarrow{\mathcal{R}}$ – the number of all possible driving routes. Note that the pickup points in each directed sequence are assumed to be different from each other. Next, let $L_{\overrightarrow{R_i}}$ be the length of route $\overrightarrow{R_i}(1 \leq i \leq M)$, where $1 \leq L_{\overrightarrow{R_i}} \leq N$. Finally, for a directed sequence $\overrightarrow{R_i}$, Let $\mathcal{P}_{\overrightarrow{R_i}}$ be the route probability set which are the probabilities of all pickup points containing in $\overrightarrow{R_i}$, where $\mathcal{P}_{\overrightarrow{R_i}}$ is a subset of $\mathcal{P}$.

The objective of this MSR problem is to recommend a travel route for a cab driver in a way such that the potential travel distance before having customer is minimized. Let $\mathcal{F}$ be the function for computing the potential travel distance (PTD) before having a customer. The PTD can be denoted as $\mathcal{F}(PoCab, \overrightarrow{\mathcal{R}}, \mathcal{P})$. In other words, the computation of PTD depends on the current position of a cab (PoCab), a suggested sequential pickup point ($\overrightarrow{\mathcal{R}_\rangle}$), and the corresponding probabilities associated with all recommended pickup points.

Based on the above definitions and notations, we can formally define the problem as (Fig. 1):

The MSR problem involves the recommendation of a sequence of pickup points and has combinatorial complexity in nature. However, this problem is practically important and interesting, since it helps to improve the business performances of taxi companies, the efficient use of energy, the productivity of taxi drivers, and the user experiences. For the MSR problem, there are two major challenges. First, how to find reliable pickup points from the historical data and how to estimate the successful probability at each pickup point. Second, there is a computational challenge to search an optimal route.

It is computationally prohibited to search for the optimal solution of the general MSR problem. Therefore, from a practical perspective, we consider a simplified version of the MSR problem. Specifically, we put a constraint on the length of a recommended route $L_{\overrightarrow{R_i}}$. In other words, the length of a recommended route is set to be a constant, that is, $L_{\overrightarrow{R_i}} = \mathcal{L}$. To simplify the discussion, let $\overrightarrow{R_i^{\mathcal{L}}}$ denote the recommended route with a length of $\mathcal{L}$. Based on this constraint, we can simplify the original objective function of the MSR problem as follows.

The MSR Problem with a Length Constraint

Objective: Recommending an optimal sequence $\overrightarrow{\mathcal{R}^{\mathcal{L}}}(\overrightarrow{\mathcal{R}^{\mathcal{L}}} \in \overrightarrow{\mathcal{R}})$. The goal is to minimize the PTD:

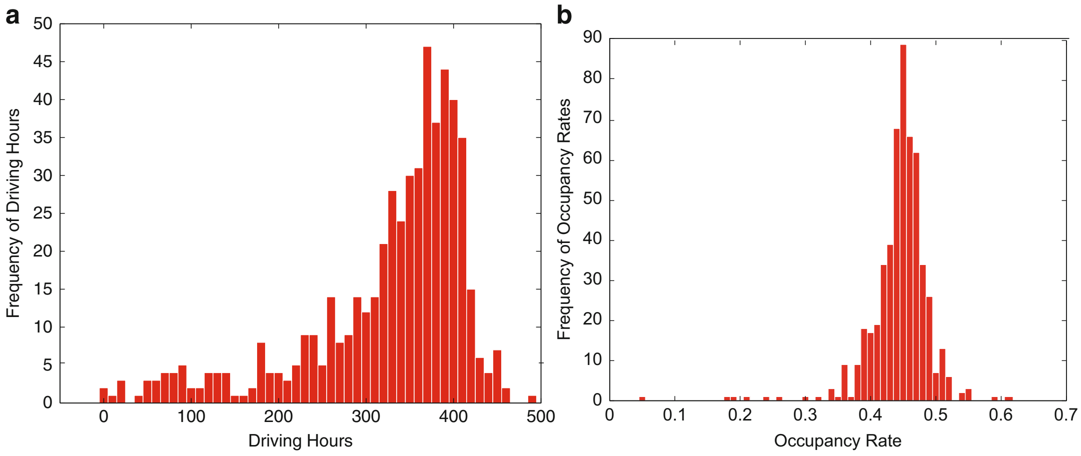
$$\min_{\overrightarrow{R_i^{\mathcal{L}}} \in \overrightarrow{\mathcal{R}}} \mathcal{F}(PoCab, \overrightarrow{R_i^{\mathcal{L}}}, \mathcal{P}_{\overrightarrow{R_i^{\mathcal{L}}}})$$

In real world, there are always high-performance experienced cab drivers, who typically have sufficient driving hours and higher customer occupancy rates – the percentage of driving time with customers. For example, Fig. 2a, b show the distributions of driving hours and occupancy rates of more than 500 drivers in San Francisco over a period of about 30 days. In the figure, we can clearly see that the drivers have different performances in terms of occupancy rates. Based on this observation, we will first extract a group of high-performance drivers with sufficient driving hours and high occupancy rates. The past pickup records of these selected drivers will be used for the generation of potential pickup points for recommendation.

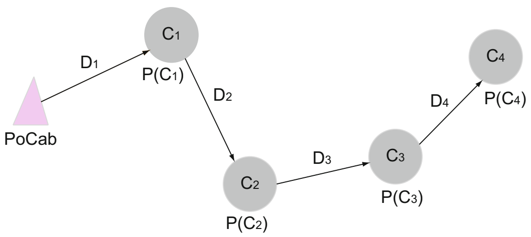
After carefully observing historical pickup points of high-performance drivers, we notice that there are relative more pickup events in some places than others. In other words, there are cluster effects of historical pickup points. Therefore, we propose to cluster historical pickup points of high-performance drivers into N clusters. The centroids of these clusters will be used for recommending pickup points. For this clustering algorithm, we use driving distance rather than Euclidean distance as the distance measure. In this study, we perform clustering based on driving distance during different time periods in order to have recommending pickup points for different time periods. Another benefit of clustering historical pickup points is to dramatically reduce the computational cost of the MRS problem.

For each recommended pickup point (the centroid of historical pickup cluster), the probability of a pickup event can be computed based on historical pickup data. The idea is to measure how frequent pickup events can happen when cabs travel across each pickup cluster. Specifically, we first obtain the spatial coverage of each cluster. Then, let $\#_T$ denote the number of cabs which have no customer before passing a cluster. For these $\#_T$ empty cabs, the number of pickup events $\#_P$ is counted in this cluster. Finally, the probability of pickup event for each cluster (each recommended pickup point) can be estimated as $P(C_i)_{1 \leq i \leq N} = \frac{\#_P}{\#_T}$, where $\#_P$ and $\#_T$ are recorded for each historical pickup cluster at different time periods.

Next we introduce the potential travel distance (PTD) function, which will be exploited for algorithm design. To simplify the discussion, we illustrate the PTD function via an example. Specifically, Fig. 3 shows a recommended driving route $PoCab \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4$ for



Mobile Sequential Recommendation, Fig. 2 Some statistics of the cab data. (**a**) Driving hours. (**b**) Occupancy rates



Mobile Sequential Recommendation, Fig. 3 A recommended driving route

the cab $PoCab$, where the length of suggested driving route $\mathcal{L} = 4$.

When a cab driver follows this route $\overrightarrow{R^{\mathcal{L}}}$, he/she may pick up customers at each pickup point with a probability $P(C_i)$. For example, a pickup event may happen at C_1 with the probability $P(C_1)$, or at C_2 with the probability $\overline{P(C_1)}P(C_2)$, where $\overline{P(C_i)} = 1 - P(C_i)$ is the probability that a pickup event does not happen at C_i. Therefore, the travel distance before a pickup event is discretely distributed. In addition, it is possible that there is no pickup event happening after going through the suggested route. This probability is $\overline{P(C_1)} \cdot \overline{P(C_2)} \cdot \overline{P(C_3)} \cdot \overline{P(C_4)}$.

In this paper, since we only consider the driving routes with a fixed length, the travel distance beyond the last pickup point is set to be D_∞ equally for all suggested driving routes. Formally, we represent the distribution of the travel distance before the next pickup event with two vectors: $\mathcal{D}_{\overrightarrow{R^{\mathcal{L}}}} = \langle D_1, (D_1+D_2), (D_1+D_2+D_3), (D_1+D_2+D_3+D_4), D_\infty\rangle$ and $\mathcal{P}_{\overrightarrow{R^{\mathcal{L}}}} = \langle P_1, \overline{P(C_1)} \cdot P(C_2), \overline{P(C_1)} \cdot \overline{P(C_2)} \cdot P(C_3), \overline{P(C_1)} \cdot \overline{P(C_2)} \cdot \overline{P(C_3)} \cdot P(C_4), \overline{P(C_1)} \cdot \overline{P(C_2)} \cdot \overline{P(C_3)} \cdot \overline{P(C_4)}\rangle$. Finally, the Potential Travel Distance (PTD) function $\mathcal{F}$ is defined as the mean of this distribution as follows.

$$\mathcal{F} = \mathcal{D}_{\overrightarrow{R^{\mathcal{L}}}} \cdot \mathcal{P}_{\overrightarrow{R^{\mathcal{L}}}},$$

where $\cdot$ is the dot product of two vectors.

From the definition of the PTD function, we know that the evaluation of a suggested driving route is only determined by the probability of each pickup point and the travel distance along the suggested route, except the common D_∞. These two types of information associated with each driving route $\overrightarrow{R_i^{\mathcal{L}}}$ can be represented with one $2\mathcal{L}$-dimensional vector $\mathcal{DP} = \langle DP_1, \cdots, DP_l, \cdots DP_{2\mathcal{L}}\rangle$. Let us consider the example in Fig. 3, where $\mathcal{L} = 4$. The 8-dimensional vector $\mathcal{DP}$ for this specific driving route is $\mathcal{DP} = \langle D_1, \overline{P(C_1)}, D_2, \overline{P(C_2)}, D_3, \overline{P(C_3)}, D_4, \overline{P(C_4)}\rangle$.

However, to find the optimal suggested route, if we use a brute-force method, we need to compute the PTD for all directed sequences with a length $\mathcal{L}$. This involves a lot of computation. Indeed, many suggested routes can be removed without computing the PTD function, because all pickup points along these routes are far away from the target cab. Along this line, we identify a monotone property of the function $\mathcal{F}$, which is that the PTD function $\mathcal{F}(\mathcal{DP})$ is strictly monotonically increasing with each attribute of vector $\mathcal{DP}$, which is a $2\mathcal{L}$-dimensional vector.

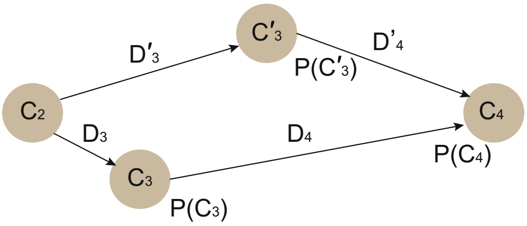
Now we introduce the $\mathcal{LCP}$ algorithm for finding an optimal driving route. In $\mathcal{LCP}$, we exploit the monotone property of the PTD function and two other pruning strategies, *route dominance* and *constrained sub − route dominance*, for pruning the search space.

Definition 1 (Route Dominance) A recommended driving route $\overrightarrow{R^{\mathcal{L}}}$, associated with the vector $\mathcal{DP}$, dominates another route $\overrightarrow{\widetilde{R^{\mathcal{L}}}}$, associated with the vector $\widetilde{\mathcal{DP}}$, iff $\exists 1 \leq l \leq 2\mathcal{L}$, $DP_l < \widetilde{DP}_l$ and $\forall 1 \leq l \leq 2\mathcal{L}$, $DP_l \leq \widetilde{DP}_l$. This can be denoted as $\overrightarrow{R^{\mathcal{L}}} \Vdash \overrightarrow{\widetilde{R^{\mathcal{L}}}}$.

By this definition, if a candidate route A is dominated by a candidate route B, A cannot be an optimal route. Next, we provide a definition of constraint sub-route dominance.

Definition 2 (Constrained Sub-route Dominance) Consider that two sub-routes $\overrightarrow{R}_{sub}$ and $\overrightarrow{R'}_{sub}$ with an equal length (the number of pickup points) and the same source and destination points. If the associated vector of $\overrightarrow{R}_{sub}$ dominates the associated vector of $\overrightarrow{R'}_{sub}$, then $\overrightarrow{R}_{sub}$ dominates $\overrightarrow{R'}_{sub}$, i.e., $\overrightarrow{R}_{sub} \Vdash \overrightarrow{R'}_{sub}$.

For example, as shown in Fig. 4, $\overrightarrow{R}_{sub}$ is $C_2 \rightarrow C_3 \rightarrow C_4$ and $\overrightarrow{R'}_{sub}$ is $C_2 \rightarrow C'_3 \rightarrow C_4$. The associated vectors of $\overrightarrow{R}_{sub}$ and $\overrightarrow{R'}_{sub}$ are $\mathcal{DP}_{sub} = \langle D_3, \overline{P(C_3)}, D_4, \overline{P(C_4)} \rangle$ and $\mathcal{DP}'_{sub} = \langle D'_3, \overline{P(C'_3)}, D'_4, \overline{P(C_4)} \rangle$, respectively. Then the dominance of $\overrightarrow{R}_{sub}$ over $\overrightarrow{R'}_{sub}$ is determined by the dominance of these two vectors. Here, we have the constraints that two routes have the same length as well as the same source and destination. The constrained sub-route dominance enables us to prune the search space in advance. This is shown in the following.



Mobile Sequential Recommendation, Fig. 4 Illustration: the sub-route dominance

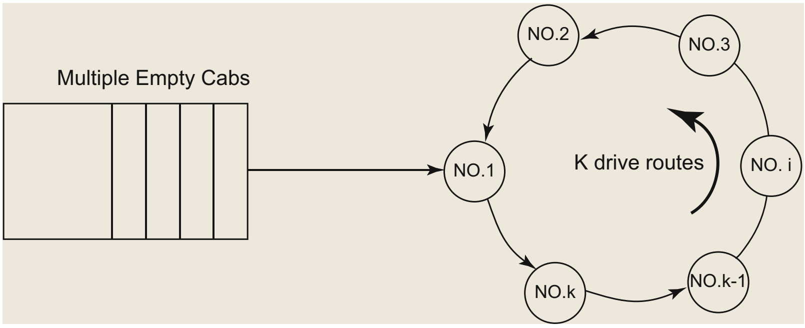
$\mathcal{LCP}$ **Pruning.** For two sub-routes A and B with a length $\mathcal{L}$, which include only pickup points, if sub-route A is dominated by sub-route B under Definition 2, the candidate routes with a length $\mathcal{L}$ which contain sub-route A will be dominated and can be pruned in advance.

Specifically, the $\mathcal{LCP}$ algorithm will enumerate all the $\mathcal{L}$-length sub-routes, which include only pickup points and prune the dominated sub-routes by Definition 2 offline. This pruning process could be done offline before the position of a taxi driver is known. As a result, $\mathcal{LCP}$ pruning will save a lot of computational cost since it reduces the search space effectively.

After all the pruning process, we will have a set of final candidate routes for a given taxi driver. To obtain the optimal driving route, we can simply compute the PTD function $\mathcal{F}$ for all the remaining candidate routes with a length $\mathcal{L}$. Then, the route with the minimal PTD value is the optimal driving route for this given taxi driver.

Even though we can find the optimal driving route for a given cab with its current position, it is still a challenging problem about how to make the recommendation for many cabs in the same area. In this section, we address this problem and introduce a strategy for the recommendation process in the real world.

A simple way is to suggest all these empty cabs to follow the same optimal driving route; however, there is naturally an overload problem, which will degrade the performance of the recommender system. To this end, we employ load balancing techniques (Grosu and Chronopoulos 2004) to distribute the empty cabs to follow multiple optimal driving routes. The problem of load balancing has been widely used in distributed



Mobile Sequential Recommendation, Fig. 5 Illustration of the *circulating mechanism*

systems for the purpose of optimizing a given objective through finding allocations of multiple jobs to different computers. For example, the load balancing mechanism distributes requests among web servers in order to minimize the execution time. For the proposed mobile recommender system, we can treat multiple empty cabs as jobs and multiple optimal driving routes as computers. Then, we can deal with this overload problem by exploiting existing load balancing algorithms. Specifically, in this study, we apply the *circulating mechanism* for the recommender systems by exploiting a round robin algorithm (Xu and Huang 2008), which is a static load balancing method.

Under the *circulating mechanism*, to make recommendation for multiple empty cabs, a round robin scheduler alternates the recommendation among multiple optimal driving routes in a circular manner. As shown in Fig. 5, we could search k optimal driving routes and recommend the NO.1 route to the first coming empty cab. Then, for the second empty cab, the NO. 2 driving route will be recommended. Assume there are more than k empty cabs, recommendations are repeated from NO. 1 route again after the kth empty cab. In practice, to achieve this, one central dispatch (processor) is needed to maintain the empty cabs and assignments among the top-k driving routes. Note that the load balancing techniques are not the focus of this entry.

Recent Development. Powell et al. (2011) proposed a grid-based method to suggest the profit locations for taxi drivers by constructing a spatiotemporal profitability map. In addition, Yuan et al. (2011, 2013) have carried out a series of studies on mobile intelligence by leveraging taxi trajectories, such as pickup point detection based on probabilistic models, and location recommendation for both the taxi drivers and customers. Different from the above studies, in this paper, we propose to develop a novel recommender system that is capable of providing an entire driving route instead of discrete pickup points, and the drivers are able to find a customer for the largest potential profit by following the recommendations. Qu et al. (2014) proposed a cost-effective mobile recommender system for taxi driver. The design goal is to maximize taxi drivers' profits when following the recommended routes for finding passengers. Specifically, they first design a net profit objective function for evaluating the potential profits of the driving routes. Then, they develop a graph representation of road networks by mining the historical taxi GPS traces and provide a brute-force strategy to generate optimal driving route for recommendation. Furthermore, they develop a novel recursion strategy based on the special form of the net profit function for searching optimal candidate routes efficiently.

Key Applications

Among the diverse class of transportation applications, the ones that will stand to benefit the most from our study include taxi and truck driving, flexible city bus tours and package delivery applications, such as the ones run by UPS and FedEx. In addition, our method could be applied for recommending driving routes for searching street parking lots in cities.

Future Directions

There are a few directions that could be explored in the future. First, there may be multiple requests of route recommendations from many users; how to optimize recommended routes in a global way is a challenging problem. Also, different users may have different preferences for multiple hidden aspects of routes, such as traffic and potential benefit; more advanced methods could be developed to learn and infer these aspects and make more personalized recommendations.

References

Abowd G, Atkeson C et al (1997) Cyber-guide: a mobile context-aware tour guide. Wirel Netw 3(5):421–433

Adomavicius G, Tuzhilin A (2005) Towards the next generation of recommender systems: a survey of the state-of-the art and possible extensions. In: TKDE

Averjanova O, Ricci F, Nguyen QN (2008) Map-based interaction with a conversational mobile recommender system. In: The 2nd international conference on mobile ubiquitous computing, systems, services and technologies

Cena F, Console L et al (2006) Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. AI Commun 19(4):369–384

Cheverst K, Davies N et al (2000) Developing a context-aware electronic tourist guide: some issues and experiences. In: The SIGCHI conference on human factors in computing systems, pp 17–24

Grosu D, Chronopoulos AT (2004) Algorithmic mechanism design for load balancing in distributed systems. IEEE TSMC-B 34(1):77–84

Miller BN, Albert I et al (2003) Movielens unplugged: experiences with a recommender system on four mobile devices. In: International conference on intelligent user interfaces

Mooney RJ, Roy L (1999) Content-based book recommendation using learning for text categorization. In: Workshop on recommender systems: algorithms and evaluation

Pazzani M (1999) A framework for collaborative, content-based, and demographic filtering. Artif Intell Rev 13:393

Powell J, Huang Y et al (2011) Towards reducing taxicab cruising time using spatio-temporal profitability maps. In: SSTD

Qu M, Zhu H et al (2014) A cost-effective recommender system for taxi drivers. In: ACM SIGKDD

Tveit A (2001) Peer-to-peer based recommendations for mobile commerce. In: The 1st international workshop on mobile commerce

van der Heijden H, Kotsis G, Kronsteiner R (2005) Mobile recommendation systems for decision making 'on the go'. In: ICMB

Xu Z, Huang R (2008) Performance study of load balancing algorithms in distributed web server systems. In: TR, CS213 University of California, Riverside

Yuan J, Zheng Y et al (2011) Where to find my next passenger. In: Ubicomp

Yuan J, Zheng Y et al (2013) T-drive: enhancing driving directions with taxi drivers intelligence. In: TKDE

Recommended Reading

Applegate DL, Bixby RE et al (2006) The traveling salesman problem: a computational study. Princeton University Press, Princeton

Borzsonyi S, Stocker K, Kossmann D (2001) The skyline operator. In: ICDE, pp 421–430

Chomicki J, Godfrey JP, Liang D (2003) Skyline with presorting. In: ICDE, pp 717–719

Dell'Amico M, Fischetti M, Toth P (1993) Heuristic algorithms for the multiple depot vehicle scheduling problem. Manag Sci 39(1):115–125

http://cabspotting.org/.

Karypis G Cluto: http://glaros.dtc.umn.edu/gkhome/views/cluto

Kian-Lee T, Pin-Kwang E, Ooi BC (2001) Efficient progressive skyline computation. In: VLDB

Papadias D, Tao GY, Seeger B (2005) Progressive skyline computation in database systems. ACM TODS 30(1):43–82

Portugal R, Lourenço HR, Paixao JP (2009) Driver scheduling problem modelling. Public Transp 1(2):103–120

Tian Y, Lee KCK, Lee W-C (2009) Finding skyline paths in road networks. In: GIS, pp 444–447

Mobile Sequential Recommendation (MRS)

► Mobile Sequential Recommendation

Mobile Travel Tour Recommendation

Qi Liu
West Lab Building of Science and Technology,
University of Science and Technology of China,
Hefei, Anhui, China

Synonyms

Mobile Recommender Systems in Tourism; Travel Recommendation for Tourists

Definition

Generally speaking, the "mobility" in users' (i.e., tourists') travel tour comes from both the users (i.e., they intend/plan to move from place to place) and their devices (i.e., some of the contexts of the user mobility may be recorded by their mobile devices). By mining these mobilities, either explicitly (interacting with the users) or implicitly (learning from user profiles and historical records), mobile travel tour recommendation aims to precisely and efficiently recommend users the destinations (e.g., Place of Interests (POI)) to visit, the routes to take, the attractive packages to choose, some other context-aware information (e.g., the events/activities nearby), etc. As a privileged type of recommender system, mobile travel tour recommendation has become a valuable tool to deal with the information overload problem in tourism.

Historical Background

Mobile travel tour recommendation has its roots in recommender systems (Gavalas et al. 2014). As is well known, recommender systems attempt to suggest items to users that they may be interested in, where items are used for denoting all the things that the systems recommend (e.g., products or services), users can be individuals (e.g., customers) or groups of individuals (e.g., groups of tourists), etc. Liu et al. (2013b). In the last decade, recommender systems have been successfully applied for improving the quality of services in a number of fields (Ricci et al. 2011). Thus, it is a natural direction to develop mobile travel tour recommender systems for recommending the right service or information to the right tourist anytime and anywhere, and similar types of recommendation solutions (e.g., content-based, collaborative filtering, and hybrid methods) could also be adopted.

However, there are still some unique features which distinguish the mobile travel tour recommendation from the traditional ones. First, in terms of the amount of information (data sources) exploited for input, traditional methods only capture the nonpersonalized or personalized information (e.g., user ID) of the users. While in mobile travel tour scenarios, the context (e.g., physical locations) of mobile tourist at a particular time is also recorded, and these contextual information can be exploited as an important source for enhancing recommendations. Thus, "context awareness" is one of the most important characteristics of mobile travel tour recommender systems. Second, in terms of the goal of the algorithm (i.e., the evaluation type), the effectiveness of the traditional recommendation algorithms is usually evaluated either by how close the predicted ratings are to the true user ratings or by the click/consumption ratio of the recommendations. Actually, these offline performances may not be the major concern of the mobile travel tour recommender systems, as the tourists usually care more about whether the feedbacks could be perfectly displayed on their portable devices (e.g., smartphones, which usually have very small interfaces) in real time. Thus, both the design of the user interface and the efficiency of recommendation algorithms should be considered. At last, it should be noted that the focus of this article is not to cover each aspect of an entire recommender system (i.e., from a systematic perspective), while we will have a focus on the recommendation techniques and key applications that make the research of mobile travel tour recommendation unique and significant.

Scientific Fundamentals

Given the collected travel tour data from a massive number of users, we can apply the methods and algorithms developed from several fields such as data mining and machine learning for mobile travel tour recommendation. However, it is worth noting that our focus is not to describe the details of all the algorithms and techniques. Instead, we would like to summarize several typical kinds of activities/tasks that have been applied, i.e., clustering, classification, pattern mining, and context-aware recommendation. In the following, we will briefly introduce the major ideas and technical solutions of these tasks one by one, and meanwhile, we will explain why they are generally recognized in mobile travel tour recommendation. For easy understanding, Fig. 1 shows one usage for each of them.

Clustering

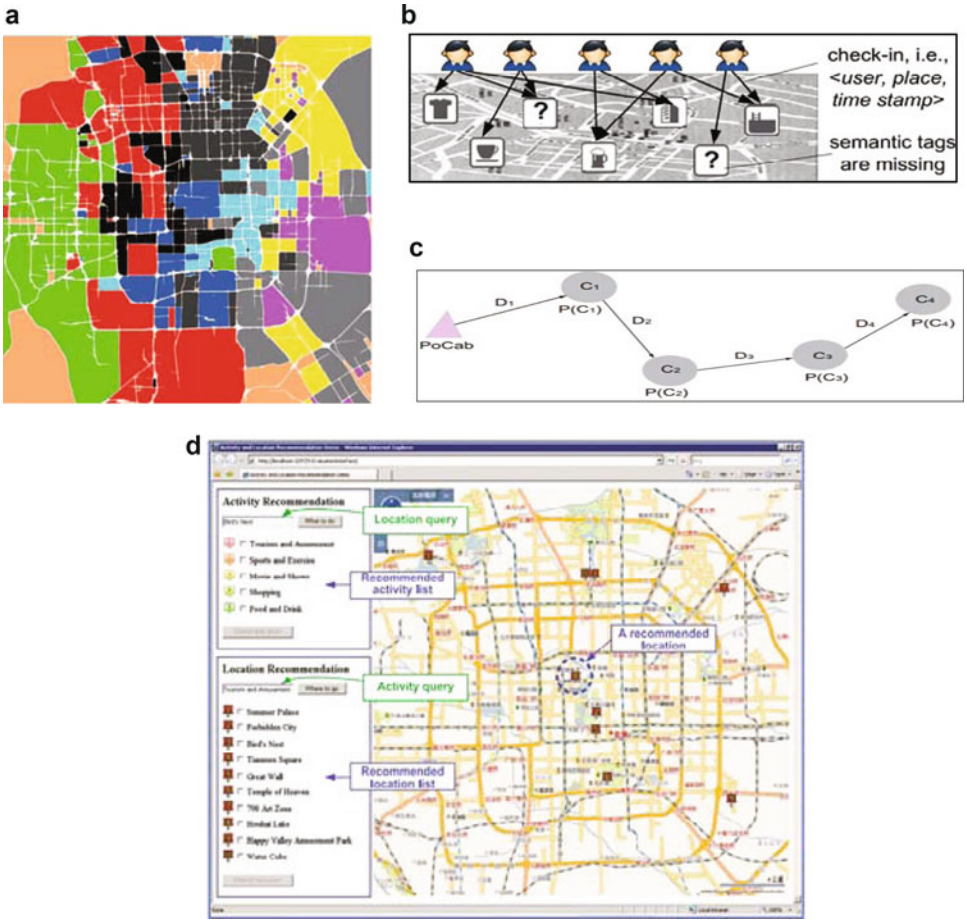
In general, clustering tries to group a set of objects (e.g., mobile users, items) and find whether there is some relationship between the objects (Liu et al. 2013b). The task of clustering can be achieved by various algorithms, and these algorithms are different in terms of their notion of what is the most appropriate similarity measure for two objects, what constitutes a cluster, and how to efficiently find these clusters. Traditional clustering solutions include partitional clustering, K-means-related clustering, hierarchical clustering, density-based clustering (e.g., DBSCAN), graph-based clustering, etc. Pang-Ning et al. (2006). For mobile travel tour recommendation, there are generally three kinds of interesting clusters to be discovered: user clusters, item clusters, and context clusters. Obviously, clustering of users helps find out the users with similar travel behaviors, and clustering of items will discover the sets of items belonging to the similar category (or with similar content). On the other hand, clustering of context is a process of recognizing and reasoning about similar contexts and situations in a mobile environment (Bao et al. 2012). All these clusters open a venue for mining the personal preferences of tourists under varying contexts and thus enable the development of personalized context-aware recommender systems (Zhu et al. 2013).

Classification

Different from clustering, we have a set of predefined class labels for classification, and we want to know which class a new object (e.g., a mobile user, a Place of Interest (POI)) belongs to Liu et al. (2013b). Classification can be done by using supervised inductive learning algorithms such as decision tree classifiers, naive Bayesian classifiers, K-nearest neighbor classifiers, Support Vector Machine (SVM) (Srivastava et al. 2000). With these labeled data and the given algorithm, the task of classification could lead to a better understanding of users, items, and contexts in the mobile travel tour domain. For instance, Zheng et al. provided an approach based on supervised learning to automatically infer transportation mode (e.g., walking, driving) from mobile users' raw GPS data (Zheng et al. 2008).

Pattern Mining

Frequent pattern mining (Agrawal et al. 1993) discovers frequent subsequences or item sets as patterns in a sequential or transactional database. It is an important data mining problem in many mobile travel tour recommendation-related tasks, such as association-rule-based classification and clustering. Specifically, existing work in this field usually adopts "support" (frequency) to measure a candidate pattern's popularity, i.e., the fraction of sequences or transactions that include the pattern in a database. Thus, the frequent patterns are the subsequences or item sets that appear with frequency no less than a given threshold. There is a great amount of work that studies efficient mining of frequent patterns, and these algorithms can be generally classified into three categories: mining frequent patterns (Han et al. 2000), frequent closed patterns (Pasquier et al. 1999), and frequent maximal patterns, respectively (Burdick et al. 2001). Since sequence is a typical data type for mining (modelling) mobile user behaviors (e.g., the time-ordered travel and trajectory sequences), we could mainly pay attention to



Mobile Travel Tour Recommendation, Fig. 1 An illustration of the usage of clustering, classification, pattern mining, and context-aware recommendation, i.e., (**a**) clustering regions in a city based on their functions (e.g., entertainment or living areas) (Yuan et al. 2012), (**b**) semantic classification (Ye et al. 2011) of significant places, (**c**) energy-efficient transportation pattern mining for drivers (Ge et al. 2010), and (**d**) location-aware mobile recommender system (Zheng et al. 2012)

sequence mining. For instance, Giannotti et al. introduced trajectory patterns as concise descriptions of frequent behaviors in terms of both space and time (Giannotti et al. 2007). In this way, service providers can predict trends and change points of mobile users, which will be helpful in placing precautions and targeted services (Liu et al. 2013b).

Context-Aware Recommendation

Since most scenarios of mobile travel tour recommendation are context aware, it is significant to incorporate contextual information into these recommender systems. Along this line, there are two major issues: First, which contextual data should really influence the recommendation procedure; second, how to incorporate selected contextual information into the mobile travel tour recommendation process (Liu et al. 2013b). For the first issue, current researches either exploit some heuristic context selection methods (e.g., use common sense and domain knowledge or take advantage of the statistical results) (Yuan et al. 2010) or choose as many contextual data as possible. For instance, different types of contexts could be treated as additional dimensions in the representation of the data as a tensor, and then tensor factorizations can project the user, the item, and each contextual dimension into a lower-dimensional feature vector (Baltrunas et al. 2011). Similar solutions could also be found in the topic-model-based methods. For instance, Bao et al. proposed two topic models (MUC and LDAC) to learn personalized contexts of mobile users, in the form of probabilistic distributions of raw context data from the context sessions (Bao et al. 2012). For the second issue, according to Adomavicius and Tuzhilin's classifications (Ricci et al. 2011), different approaches using contextual information in the recommendation process can be broadly categorized into two groups: (1) recommendation via context-driven querying and search (i.e., use contextual information to query or search a certain repository of resources, e.g., restaurants) and (2) recommendation via contextual preference elicitation and estimation. There are three different algorithmic paradigms of recommendation via contextual preference elicitation and estimation–contextual pre-filtering, post-filtering, and collaborative modeling–for incorporating contextual information into the recommendation process. For instance, for travel package recommendation, Liu et al. exploited the seasonal collaborative filtering as contextual pre-filtering (contextual information drives data selection or data construction), i.e., they first found the seasonal (contextual) nearest neighbors for given tourist and then used collaborative filtering for ranking the candidate packages (Liu et al. 2011).

Key Applications

It is a common phenomenon that individuals on the move, e.g., tourists on a sight-seeing trip in an unfamiliar area (e.g., city), often find themselves overwhelmed by the challenges of coping with the unfamiliar environments. Actually, new developments in mobile computing, wireless networking, web technologies, and social networking could leverage massive opportunities to provide highly accurate and effective tourist recommendations in a number of applications. These recommendations include but are not limited to the destinations (e.g., Place of Interests) to visit, the routes to take, the attractive packages to choose, and some other context-aware information (e.g., the events/activities nearby). For instance, Fig. 2 shows a toy example of the items in key applications of mobile travel tour recommendation.

POI Recommendation

The widespread use of location-based social network services (LBSNs, e.g., Foursquare, Facebook Places, and Google Latitude) has enabled the opportunities for better location-based services through Point-of-Interest (POI) recommendation (Liu and Xiong 2013), that is, providing personalized recommendation of Places of Interests (or check-ins (Lian et al. 2014)), such as landscapes (e.g., museum), restaurants, stores, and cinema theaters, for tourists. In this way, the owners of POIs could have more targeted customers, and for mobile users, they could identify the most relevant POIs and have better user experiences. Due to its complexity and its connection to location-based social networks, the decision process in which a user choosing a POI is very complex and can be influenced by various factors, such as user preferences, geographical influences, and user mobility behaviors (Liu et al. 2013a). Thus, there are some unique features and challenges in POI recommendation.

Route Planning

The most prominent outcome of recent research efforts in mobile tourism has been the substantial number of mobile electronic guide systems (e.g., guiding the travel/driving route), which have been in the spotlight over the past few years (Gavalas et al. 2014; Kenteris et al. 2011). Different from the recommendation of a single POI, travel route planning of these guide systems will recommend an entire route (composed by several POIs with visiting orders) for users. As the tourists are usually unfamiliar with the travel areas (i.e., city), and meanwhile, they may have limited travel time and budget, it is necessary and urgent to provide energy-efficient travel route recommendation (Ge et al. 2010; Yuan et al. 2010).

a
b
New Route
Third Ward
Fourth Ward
First Ward
Charlotte
Downtown Charlotte
MapAnything
c
Discover
Croatia & Montenegro
14 NIGHTS
£899
ALL INCLUSIVE
TRAVELANDTOURTODAY.CO.UK
Holiday includes...
Call free on
0800 015 6512
d
LET'S PARTY!

1 First Ward Law Offices
714 S. Mcdowell St. Charlotte, NC 28205
9:00am - 0 hr, 30min
0.25 miles / 3 minutes
8:45am - 9:00am

2 The Marshall Group
1300 10th St. Unit B Charlotte, NC 28206
10:00am - 0 hr, 30min
0.58 miles / 5 minutes
9:35am - 10:00am

3 Loyal Training Co.
700 Parkway Dr E 9th St. Charlotte, NC 28206
11:30am - 1 hr, 00min
0.88 miles / 7 minutes
10:37am - 11:30am

4 Datacorp Financial
2499 South Blvd. Charlotte, NC 28207
1:00 pm - 0hr, 30min
0.68 miles / 6 minutes
12:36pm - 1:00pm

5 CKE Limited

Total Trip: 5 stops, 2.8 miles, 25 minutes

Crystal-clear water, beautiful islands and historic cities – these are just a few of the reasons why Croatia is known as one of Europe's finest gems. But just to the south is the lesser-explored but equally precious Montenegro, with its own fair share of natural and cultural treasures. Why not visit both on our fabulous-value all-inclusive two-centre holiday? You'll spend seven nights on the Dubrovnik Riviera with included excursions to Dubrovnik and the Konvale Valley before heading to Becici in Montenegro.

Here, two further included excursions take you to the charming resorts of Budva and Sveti Stefan but there's also plenty of time to relax and enjoy the facilities of your hotel or to explore this fascinating and beautiful part of the world further.

- Travel insurance and additional cancellation cover
- Return scheduled flights and transfers with British Airways from Gatwick to Dubrovnik
- All breakfasts, lunches and dinners
- Free bar open 12 hours a day
- FOUR EXCURSIONS
- Plus much more...

SINGLE ROOMS AT NO EXTRA COST

quoting TELE2 or visit
or visit saga.co.uk/adriatic

Mobile Travel Tour Recommendation, Fig. 2 A toy example of the items in key applications of mobile travel tour recommendation, i.e., (**a**) POI: statue of liberty, (**b**) a toy route from MapAnything (http://cloudbilt.com/mapanything/), (**c**) a travel package named "Discover Croatia and Montenegro" from http://www.travelandtourtoday.co.uk/, and (**d**) a simulated social event

Travel Package Recommendation

As an emerging trend, more and more travel companies provide online services. However, the rapid growth of online travel information imposes an increasing challenge for tourists, who have to choose from a large number of available travel packages for satisfying their personal needs, when he/she is planning a trip (Liu et al. 2014). Specifically, a travel package is a general service package consisting of the POIs (landscapes) and some related information, such as the travel route, the price, and the travel period. For convenience, most of the current tourists tend to travel with a given package, e.g., during the vacation period. On the other side, to get more business and profit, the travel companies have to understand these preferences from different tourists and serve more attractive packages. Therefore, the demand for intelligent travel package recommendation, from both tourists and travel companies, is expected to increase dramatically (Liu et al. 2011).

Advanced Context-Aware Tourist Services

When traveling, the curious tourists usually want to get more exploratory, even unexpected, attractions or services, e.g., attend the social events nearby. Indeed, according to Meetup.com, (http://www.meetup.com/), there are more than 10,000 events which are organized every day, and the number of responses to invitations may even exceed 100 times per minute, and some of them are coming from the tourists. In other words, the tourists need to get more context-aware services beyond the "traveling." For instance, Lee and Park presented a news (e.g., local news) recommender for the mobile user (Lee and Park 2007); Quercia et al. tested a variety of algorithms for recommending social events (Quercia et al. 2010); Zheng and Xie developed a system to visualize the positions of nearby visitors (e.g., for personalized social/travel friend recommendation) (Zheng and Xie 2011).

Future Directions

Compared with traditional recommendations, mobile travel tour recommender systems can provide better recommendation results when coupled with personalized contextual information, since they can understand mobile users more accurately. However, there are also some limitations and key issues that should be taken care of when conducting context-aware mobile travel tour recommendation, and these limitations and issues lead to the future research directions (Liu et al. 2013b; Gavalas et al. 2014):

Privacy Issue

With the benefits brought by context-aware mobile applications, more attentions have been paid to privacy issues due to the capability of these applications to collect, store, use, and disclose the contextual information of those who use them. Hence, the success of mobile travel tour recommendation is also conditioned on the availability of effective privacy protection mechanisms. Recently, many privacy protection techniques have been specifically developed for location-based services, which are mainly in the form of online compared to the traditional offline privacy protection in microdata release due to the dynamic nature of spatiotemporal information. Unfortunately, these techniques are often insufficient and inadequate when applied to generic context-aware travel tour services. Therefore, studying on the privacy protection mechanisms is still a hot and important direction.

User Interface

Due to the small interfaces of mobile devices, such as mobile phones and PDAs, recommendation sessions can be difficult and frustrating for end users. The limitation of small interfaces for browsing and item recommendation are in three aspects: (1) on a small screen the user may be forced to carry out extensive scrolling while browsing a web page, and the more a user has to scroll down, the smaller the chances of an item being clicked; (2) a user of a small screen is less effective in completing an assigned task when compared with users with large screens; (3) mobile devices offer limited input and interaction capabilities. Though there are so many established techniques for addressing this user interface issue (Ricci 2010), to develop new techniques is still an important topic with the increasing popularity of smart mobile devices.

Multisource Data Incorporation

Indeed, both structured and unstructured context-aware data are now quickly being gathered by ubiquitous information-sensing sources. These data are so large and complex that they become difficult to process them using traditional data analysis tools. It is necessary to develop new solutions for dealing with (e.g., storing, searching, sharing, analyzing, incorporating, and visualizing this type of big data) and thus mine more information for personalized mobile travel tour recommendation.

Energy Efficiency Issues

There are generally two aspects of the energy efficiency issue in mobile travel tour recommendation, i.e., from the device perspective and from the tourist perspective, respectively. First, with the development of those intelligent context-aware mobile applications, the energy cost of the devices' context sensing becomes the bottleneck for the success of these applications limited by the battery capacity, and many studies for energy-efficient context sensing have been reported. Second, the tourists usually have to interact (e.g., inputting their requirements) with the mobile system for getting travel recommendation, and this process could easily make the tourists lose their patience. Even though many works have been conducted to improve the energy efficiency, they usually have to make a trade-off between efficiency and recommendation accuracy. Hence, the energy issue is still a tricky and important problem in the field of context-aware mobile travel tour recommendation.

Recommended Reading

Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. In: ACM SIGMOD record, Washington, DC, vol 22. ACM, pp 207–216

Baltrunas L, Ludwig B, Ricci F (2011) Matrix factorization techniques for context aware recommendation. In: Proceedings of the fifth ACM conference on recommender systems. ACM, New York, pp 301–304

Bao T, Cao H, Chen E, Tian J, Xiong H (2012) An unsupervised approach to modeling personalized contexts of mobile users. Knowl Inf Syst 31(2):345–370

Burdick D, Calimlim M, Gehrke J (2001) Mafia: a maximal frequent itemset algorithm for transactional databases. In: Proceedings of 17th international conference on data engineering, Heidelberg. IEEE, Los Alamitos, pp 443–452

Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) Mobile recommender systems in tourism. J Netw Comput Appl 39:319–333

Ge Y, Xiong H, Tuzhilin A, Xiao K, Gruteser M, Pazzani M (2010) An energy-efficient mobile recommender system. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose. ACM, New York, pp 899–908

Giannotti F, Nanni M, Pinelli F, Pedreschi D (2007) Trajectory pattern mining. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 330–339

Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: ACM SIGMOD record, Dallas, vol 29. ACM, New York, pp 1–12

Kenteris M, Gavalas D, Economou D (2011) Electronic mobile guides: a survey. Pers Ubiquitous Comput 15(1):97–111

Lee HJ, Park SJ (2007) Moners: a news recommender for the mobile web. Expert Syst Appl 32(1):143–150

Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y (2014) Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 831–840

Liu B, Xiong H (2013) Point-of-interest recommendation in location based social networks with topic and location awareness. In: SDM, Austin. SIAM, pp 396–404

Liu Q, Ge Y, Li Z, Chen E, Xiong H (2011) Personalized travel package recommendation. In: 2011 IEEE 11th international conference on data mining (ICDM), Vancouver. IEEE, Los Alamitos, pp 407–416

Liu B, Fu Y, Yao Z, Xiong H (2013a) Learning geographical preferences for point-of-interest recommendation. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, Chicago. ACM, New York, pp 1043–1051

Liu Q, Ma H, Chen E, Xiong H (2013b) A survey of context-aware mobile recommendations. Int J Inf Technol Decis Mak 12(01):139–172

Liu Q, Chen E, Xiong H, Ge Y, Li Z, Wu X (2014) A cocktail approach for travel package recommendation. IEEE Trans Knowl Data Eng 26(2):278–293

Pang-Ning T, Steinbach M, Kumar V et al (2006) Introduction to data mining. In: Library of congress. Addison Wesley

Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Database Theory ICDT 99, Jerusalem. Springer, Berlin/New York, pp 398–416

Quercia D, Lathia N, Calabrese F, Di Lorenzo G, Crowcroft J (2010) Recommending social events from mobile phone location data. In: 2010 IEEE 10th international conference on data mining (ICDM), Sydney. IEEE, Los Alamitos, pp 971–976

Ricci F (2010) Mobile recommender systems. Inf Technol Tour 12(3):205–231

Ricci F, Rokach L, Shapira B (2011) Introduction to recommender systems handbook. Springer, New York

Srivastava J, Cooley R, Deshpande M, Tan P-N (2000) Web usage mining: discovery and applications of usage patterns from web data. ACM SIGKDD Explor Newsl 1(2):12–23

Ye M, Shou D, Lee W-C, Yin P, Janowicz K (2011) On the semantic annotation of places in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 520–528

Yuan J, Zheng Y, Zhang C, Xie W, Xie X, Sun G, Huang Y (2010) T-drive: driving directions based on taxi trajectories. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, San Jose. ACM, New York, pp 99–108

Yuan J, Zheng Y, Xie X (2012) Discovering regions of different functions in a city using human mobility and pois. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, Beijing. ACM, New York, pp 186–194

Zheng Y, Xie X (2011) Learning travel recommendations from user-generated GPS traces. ACM Trans Intell Syst Technol 2(1):2

Zheng Y, Liu L, Wang L, Xie X (2008) Learning transportation mode from raw gps data for geographic applications on the web. In: Proceedings of the 17th international conference on World Wide Web, Beijing. ACM, New York, pp 247–256

Zheng VW, Zheng Y, Xie X, Yang Q (2012) Towards mobile intelligence: learning from GPS history data for collaborative recommendation. Artif Intell 184: 17–37

Zhu H, Chen E, Xiong H, Yu K, Cao H, Tian J (2013, to appear) Mining mobile user preferences for personalized context-aware recommendation. ACM Trans Intell Syst Technol 5(4):1–27

Mobile Usage and Adaptive Visualization

Tumasch Reichenbacher
Department of Geography, University of Zurich,
Zurich, Switzerland

Synonyms

Adaption, complete; Adaptive, context-aware; Customization; Personalized maps; Personalized visualization

Definition

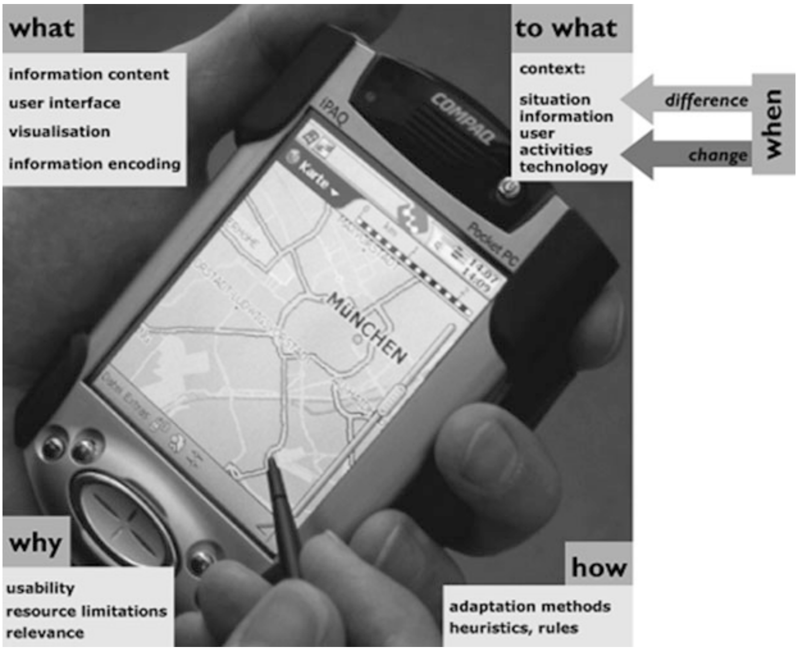
Generally, adaptive visualization is the adjustment of the visualization of geographic information and associated parts in the visualization process such as the interface, the information content, and the information encoding by a visualization application or a geospatial web service to a specific usage context.

The concept of adaptation has been applied mainly to the mobile usage context, where maps and other visualization forms of geographic information are stored and/or displayed on portable devices that are in most cases owned by the user and are constantly carried around. In such a context, the mobile user generally demands geospatial information that is retrieved either locally or remotely over an internet connection and visualized through the mediation of a user interface (Reichenbacher 2004). Figure 1 outlines the main components of the adaptation of visualization. First of all, the reason for the adaptation is given through the need for improving the usability of the mobile geospatial information access, limitations of resources in the mobile usage context, and the desire to enhance the relevance of the presented geospatial information. The adaptation concept further distinguishes what is adapted, the so-called adapted, from what it is adapted to, the adaptation target. The former are the objects of a visualization of geographic information that are adaptable and the latter is the referential source of information to which the adaptation is directed to, i.e., the mobile usage context. Dependent on changes or differences in the usage context, adaptation methods are triggered that adapt the adaptation objects to the adaptation target.

The adaptation target, the mobile usage context, and its components will be analyzed in more detail below.

Historical Background

The first attempts at adaptivity were made in the field of human-computer interaction in the late 1980s. Numerous prototypes of adaptive user interfaces were developed documented in



Mobile Usage and Adaptive Visualization, Fig. 1 The basic components of adaptive visualization (Based on Brusilovsky 2001)

the literature of the early 1990s (Browne et al. 1990; Oppermann 1994; Schneider-Hufschmidt et al. 1993). The rise of multimedia provided the next field of adaptivity. In the latter half of the 1990s, researchers focused more on adaptive hypermedia (Brusilovsky 2001). The first investigation of adapting maps and the visualization of geographic information for mobile usage dates roughly to the new millennium when mobile network technology had matured and mobile telecommunications had become a mass market. This new technology, based on cell nets and mobile phones, allowed users to access geographic information almost everywhere and revealed at the same time the need for adaptation. Around the same time the first location-based services (LBS) were developed. This concept is related to adaptive visualization of geographic information, although it is much narrower and more technology-oriented as elaborated below.

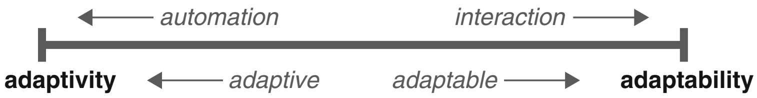
Scientific Fundamentals

The prerequisite of any change to visualization is that the visualization is in itself flexible and could principally be changed at all. This quality refers to the term adaptability. In the case of geospatial visualization this has only become possible with digital representations of geographic information. Digital representations of geographic information separate the storage and the display of the information offering the flexibility for any possible changes or adjustments that were not possible with analogue paper maps or atlases. The latter are not adaptable and hence not susceptible to adaptation.

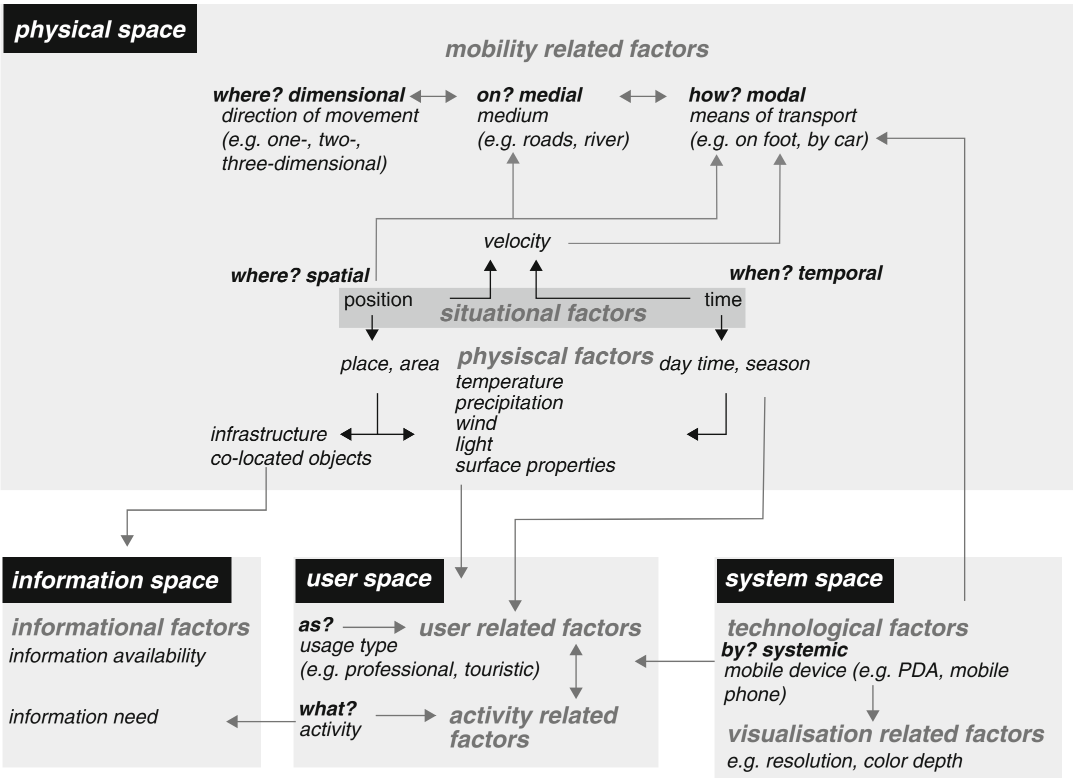
Regarding the adaptation of geospatial visualizations, two types can be distinguished. Adaptable geospatial visualizations offer the user tools to change and modify properties of the visualization. This corresponds basically to the concept of customization.

Adaptive geospatial visualizations, on the other hand, can change their characteristics automatically based on the usage context. Often this adaptivity is called "self-adapting". This dichotomy of adaptable and adaptive systems is depicted in Fig. 2 (Oppermann 1994).

Adaptivity can take many forms depending on whether and how much the user and the system are involved in initiating, deciding about and conducting adaptation steps. For a detailed analysis of possible combinations of adaptivity, see



Mobile Usage and Adaptive Visualization, Fig. 2 The adaptivity-adaptability dichotomy



Mobile Usage and Adaptive Visualization, Fig. 3 The dimensions of the mobile usage context

Schneider-Hufschmidt et al. (1993). If the focus of the adaptation is on the user, often the terms individualization or personalization are used. A good overview on the general topic of adaptation is given in Browne et al. (1990) and Oppermann (1994).

The prerequisite for the mobile usage of geographic information is remote access to geospatial information. This is provided by information transmission through mobile telecommunication networks and the availability of small, portable computing devices. These two major technological fields, mobile telecommunication and mobile computing, offer the user the mobility of geospatial information usage. This mobility of the user and hence the usage of the geographic information lead to different and changing usage contexts.

In the definition of adaptive visualization, it was stated that the target of the adaptation is the mobile usage context. This context is naturally composed of several dimensions. Figure 3 illustrates these mobile usage context dimensions, some of their interrelationships and a few exemplary parameters for these dimensions. First of all it is possible to distinguish the scope of the context: the information space, the physical space, the visualization space, and the user space. These spaces are characterized by different contextual factors.

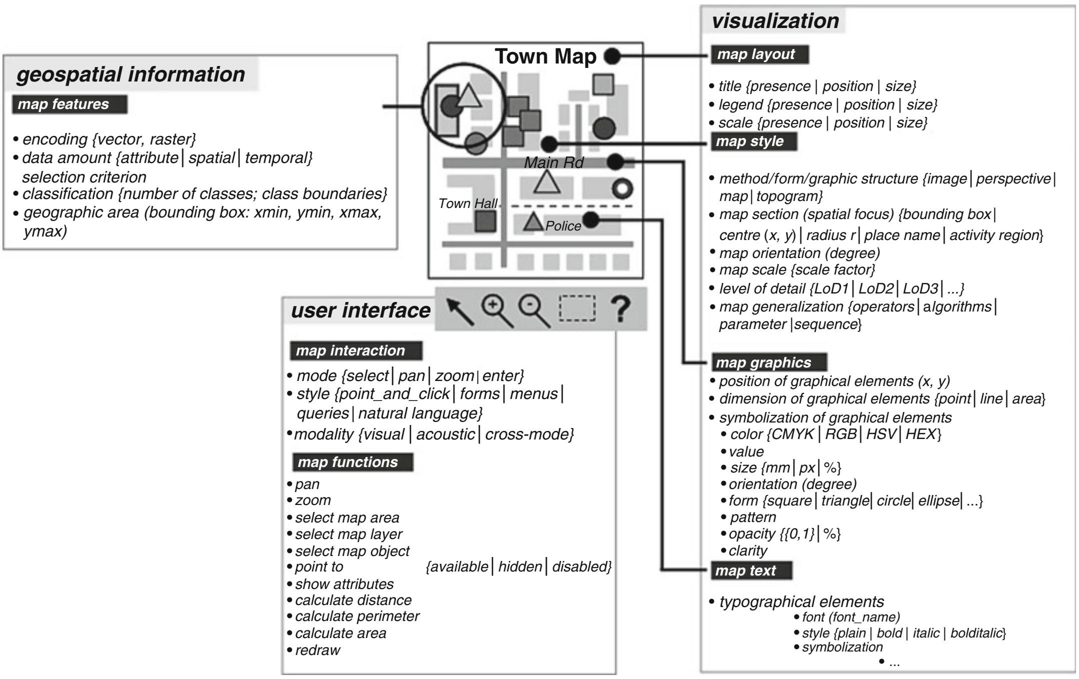
The physical space is primarily defined by the position of the user in space and time. These two parameters define the user's situation. Any situation is characterized by further physical conditions such as light conditions, temperature, precipitation, surface etc. Some of these parameters also influence the user's mobility which is typified by mobility factors such as the direction, the medium and mode of the movement. The mobility factors, as well as the situation, have themselves a strong influence on the user's activity. Space, time, and mobility constrain the user activities. The user space incorporates factors related to the user characteristics, the kind of geographic information usage, and user activities. Some adaptation approaches regard the user as a separate source of information for the adaptation process. This is for example the case in adaptive user interfaces or personalization of maps, where the focus is more on modeling the user and his or her characteristics. However, it is argued that the user is a central part of the mobile usage context and therefore better to be modeled together with the other context dimensions. The activities of a mobile user have an influence on the informational factors, e.g., the information needed for the successful accomplishment of current actions to reach the user's goal. The information space is also important for inferring the information available for a specific user situation, for instance by determining collocated objects in the spatial context of the user's position. And finally, the information space is also connected to the system space. The system space covers technological factors, such as the telecommunication technology in use, network bandwidths and mobile device capabilities. These factors determine, in parts, the possibility, amount, and speed of information transmission to the mobile user. The mobile device in use and its properties constrain visualization-related factors, e.g., the number of displayable colors or data formats that can be rendered by the device. For further information on context modeling, see Dey (2001) and Sarjakoski and Nivala (2005).

In mobile computing applications or services that have knowledge about the context they are used in or run are called context-aware. One way of capturing this knowledge about the context is the application of sensors. Sensors help a machine or software agent to sense parts of its environment and hence get information about this environment. In the case of the context of geographic information usage the most important and prominent sensor is a global positioning system (GPS) receiver. By determining the current position of the mobile device and thus of the mobile user it provides a special kind of context awareness, spatial awareness. Spatial awareness offers the opportunity for the simplest and most evident adaptivity of geographic information visualization for mobile usage. This kind of adaptivity is implemented in LBS where the information content is adapted to a specific location, in most cases a mobile user's current position. The simplest way of applying spatial awareness to the visualization of geographic information for a mobile user is a self-orienting map that moves the map section automatically based on the position received from a GPS receiver. This is implemented in car navigation systems where the map section is recentered when you are reaching the edge of the map.

Although the visualization of geographic information on mobile devices could be implemented in heavy, adaptable applications, it will rather be based on a service-oriented architecture (SOA). Such a web service concept for interoperable geospatial information services is specified by the Open Geospatial Consortium (OGC). The advantage of geospatial web services is their adaptability potential through the adjustment of service parameters.

The adaptation objects, i.e., the objects that can be changed for visualization either by the user or the system/service, to the adaptation target, i.e., the mobile usage context, are summarized in Fig. 4 using the example of a mobile map. Although visualization and its subcomponents are the central object of adaptation, the user interface and the geospatial information can be treated as separate adaptation objects of adaptive visualization.

The final building blocks of adaptive visualization are the adaptation methods that


geospatial information
map features
• encoding {vector, raster}
• data amount {attribute| spatial| temporal}
selection criterion
• classification {number of classes; class boundaries}
• geographic area (bounding box: xmin, ymin, xmax, ymax)
Town Map
Main Rd
Town Hall
Police
user interface
?
map interaction
• mode {select| pan| zoom| enter}
• style {point_and_click| forms| menus| queries| natural language}
• modality {visual| acoustic| cross-mode}
map functions
• pan
• zoom
• select map area
• select map layer
• select map object
• point to
• show attributes
• calculate distance
• calculate perimeter
• calculate area
• redraw
{available| hidden| disabled}
visualization
map layout
• title {presence | position | size}
• legend {presence | position | size}
• scale {presence | position | size}
map style
• method/form/graphic structure {image| perspective| map| topogram}
• map section (spatial focus) {bounding box| centre (x, y)| radius r| place name| activity region}
• map orientation (degree)
• map scale {scale factor}
• level of detail {LoD1| LoD2| LoD3| ...}
• map generalization {operators| algorithms| parameter |sequence}
map graphics
• position of graphical elements (x, y)
• dimension of graphical elements {point| line| area}
• symbolization of graphical elements
• color {CMYK | RGB| HSV| HEX}
• value
• size {mm| px| %}
• orientation (degree)
• form {square| triangle| circle| ellipse| ...}
• pattern
• opacity {{0,1}| %}
• clarity
map text
• typographical elements
• font (font_name)
• style {plain | bold | italic | bolditalic}
• symbolization
• ...

Mobile Usage and Adaptive Visualization, Fig. 4 Adaptable objects in adaptive visualization

are responsible for adjusting one or several adaptation objects to the adaptation target.

The input to the adaptation method is one or more adaptation objects. The context parameter values are used to control the adaptation method. The output of the adaptation method is a set of adapted objects. For the adaptation of the visualization of geographic information to the mobile usage context, the following adaptation methods are applicable:

- Selection method: this method selects map features depending on the usage context in order to reduce the map content and the information density. This method operates as a kind of filter. The filter criterion can be a spatial relation as in the case of LBS, or it could be a user preference stored in a user profile.
- Prioritization method: this method classifies the priority of selected information items with regard to the current usage context. The priority classes are based on the relevance of the information for different context factors, e.g., relevance for the current location or relevance for the current activity.
- Substitution method: this method substitutes one visualization form with an equivalent presentation form. A map could, for instance, be substituted by an image or abstract symbols might be replaced by pictorial symbols.
- Symbolization method: this method changes the symbolization of the visual elements by applying a different symbol style or by switching to a predefined design alternative.
- Configuration method: this method configures or reconfigures visual components in order to adjust the visualization to the usage context. For instance a different base map or a different scale might be selected depending on mobilityrelated factors such as means of transport or movement speed. The configuration method could also configure the user interface through hiding or aggregating functions or changing interaction modes.
- Encoding method: this method changes the encoding of the information (e.g., vector to raster) to be transmitted to a mobile device in dependence on technological factors such as the bandwidth available or the capabilities of the device.

In the simplest case, a fixed rule base exists that triggers the adaptation methods and states what changes will be applied if specific context conditions are given. More advanced adaptive systems incorporate a learning component that dynamically changes the knowledge base in dependence on the system usage and performed user interactions.

Key Applications

Adaptive visualization, so far, is used mainly in the domain of mobile systems and services. The domains discussed here are mobile guide systems, LBS, and mobile geospatial web services.

Mobile Guide Systems

The adaptivity principle has been applied in many mobile guide systems (Oppermann and Specht 1999; Patalaviciute et al. 2005). The visualization in such guide systems is either adapted to the user characteristics or preferences (personalization) or to the usage context factors in general.

Location-Based Services

LBS are inherently adaptive by spatially filtering information for a specified location (Krug et al. 2003). Different ways of adapting LBS are described in Steiniger et al. (2006).

Mobile Geospatial Web Services

Most mobile geospatial web services include maps or other forms of visualization of geographic information to assist the user in way-finding, orientation or similar spatial tasks. Adaptive visualizations are employed in such services to improve their usability (Sarjakoski and Nivala 2005).

Future Directions

In principle, any visualization of geographic information can be adapted to the context of use. With the growing amount of geospatial data available the demand for the adaptation to specific needs and contexts will probably get stronger in the near future.

The adaptivity principle is especially important for the evolving mobile geospatial web services and their interfaces where usability is a crucial issue. A thoughtful implementation of adaptive behavior helps to reduce complexity and take some of the cognitive load inherent in mobile usage contexts. Adaptive visualization will play an important role in personalized geospatial web services and egocentric maps (Reichenbacher 2005).

Cross-References

- ▶ Constraint Data, Visualizing
- ▶ Web Services, Geospatial

References

Browne D, Totterdell P, Norman M (eds) (1990) Adaptive user interfaces. Academic, London

Brusilovsky P (2001) Adaptive hypermedia. User Model User-Adapt Interact 11:87–110

Dey AK (2001) Understanding and using context. Personal Ubiquitous Comput J 5:4–7

Krug K, Mountain D, Phan D (2003) Location-based services for mobile users in protected areas. Geoinformatics 6(2):26–29

Oppermann R (ed) (1994) Adaptive user support: ergonomic design of manually and automatically adaptable software. Lawrence Erlbaum, Hillsdale

Oppermann R, Specht M (1999) Adaptive mobile museum guide for information and learning on demand, in human-computer interaction. In: Bullinger HJ, Ziegler J (eds) Proceedings of HCI International'99. Communication, cooperation, and application design, vol 2. Erlbaum, Mahwah, pp 642–646

Patalaviciute V et al (2005) Using SVG-based maps for mobile guide systems-a case study for the design, adaptation, and usage of SVG-based maps for mobile nature guide applications. Available via http://www.svgopen.org/2005/papers/MapsForMobileNatureGuideApplications/index.html

Reichenbacher T (2004) Mobile cartography adaptive visualisation of geographic information on mobile devices. Verlag Dr. Hut, Munich

Reichenbacher T (2005) Adaptive egocentric maps for mobile users. In: Meng L, Zipf A, Reichenbacher T (eds) Map-based mobile services. Theories, methods and implementations. Springer, Heidelberg, pp 141–158

Sarjakoski T, Nivala A-M (2005) Adaptation to context – a way to improve the usability of mobile maps. In: Meng L, Zipf A, Reichenbacher T (eds) Map-based mobile services. Theories, methods and implementations. Springer, Heidelberg, pp 7–123

Schneider-Hufschmidt M, Kühme T, Malinowski U (eds) (1993) Adaptive user interfaces: principles and practice. North-Holland, Amsterdam

Steiniger S, Neun M, Edwardes A (2006) Foundations of location based services. Available via http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf

MOBR

► Minimum Bounding Rectangle

Model Driven Architecture

► Modeling with Enriched Model-Driven Architecture

Model Driven Development (MDD)

► Modeling with Enriched Model-Driven Architecture

Model Driven Engineering

► Modeling with Enriched Model-Driven Architecture

Model Generalization

► Abstraction of Geodatabases

Modeling and Multiple Perceptions

Christine Parent[1], Stefano Spaccapietra[2], and Esteban Zimányi[3]
[1]University of Lausanne, Lausanne, Switzerland
[2]Swiss Federal Institute of Technology, Lausanne, Switzerland
[3]Free University of Brussels, Brussels, Belgium

Synonyms

Data modeling; Multirepresentation; Multiscale databases

Definition

Multirepresentation generalizes known concepts such as database views and geographic multiscale databases. This chapter describes the handling of multi-representation in the MADS (Modeling Application Data with Spatio-temporal features) data modeling approach. MADS builds on the concept of orthogonality to support multiple modeling dimensions. The structural basis of the MADS model is based on extended entity-relationship (ER) constructs. This is complemented with three other modeling dimensions: space, time, and representation. The latter allows the specification of multiple perceptions of the real world and modeling of the multiple representations of real-world elements that are needed to materialize these perceptions.

Historical Background

Traditional database design organizes the data of interest into a database schema, which describes objects and their relationships, as well as their attributes. At the conceptual level, the design task relies on well-known modeling approaches such as the ER model (Chen 1976) and Unified Modeling Language (UML) (Rumbaugh et al. 2005). These approaches only deal with classical alphanumeric data. The idea of using conceptual spatial and spatiotemporal data models emerged in the 1990s. Most proposals were extensions of either the ER (e.g., Bédard et al. 1992; Tryfona and Jensen 1999; Parent et al. 2006a), UML (e.g., Price et al. 1999; Bédard et al. 2004), or object-oriented data models (e.g., David et al. 1993). Spatiotemporal data models are the current focus of research and development, both in academia and in industry: They allow the representation of the past and future evolution of geographic objects as well as moving and deforming objects. Both features are essential for complex development issues such as environmental management and city planning.

Most geographic applications confronted with a variety of user categories showing different requirements of the same data (e.g., different administrations involved in city management) also need another modeling dimension: multiple representations. Multiple representations allow, for example, the spatial feature of a city being described as a point and as an area, the former for use in statewide maps, the latter in local maps. Interest in supporting this functionality has emerged recently (Parent et al. 2006a; Bédard and Bernier 2002) and the concept of multirepresentation is now popular in the research community and with user groups. The MADS proposal is playing an important role in establishing and advancing this trend.

Scientific Fundamentals

Definition 1 (Conceptual data model). A data model is conceptual if it enables a direct mapping between the perceived real world and its representation with the concepts of the model. In particular, a conceptual model does not have implementationrelated concerns.

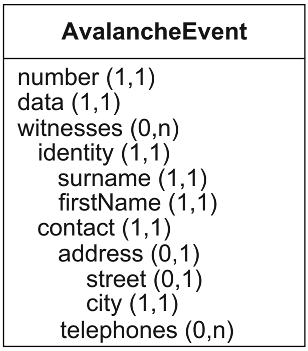
Definition 2 (Data modeling dimension). A data modeling dimension is a domain of representation of the real world that focuses on a specific class of phenomena. Examples of modeling dimensions include: data structure, space, time, and multirepresentation.

Definition 3 (Orthogonality). Modeling dimensions are said to be orthogonal if, when designing a database schema, choices in a given dimension do not depend on the choices in other dimensions. For example, it should be possible to record the location of a reservoir on a river (a spatial feature in the space dimension) irrespective of whether the reservoir, in the data structure dimension, has been modeled as an independent object or as an attribute of the river object. Orthogonality greatly simplifies the data model and its use, while enhancing its expressive power, i.e., its ability to represent all phenomena of interest. The orthogonality of multiple modeling dimensions is an essential characteristic of MADS. A detailed presentation of the model can be found in Parent et al. (2006a, b).

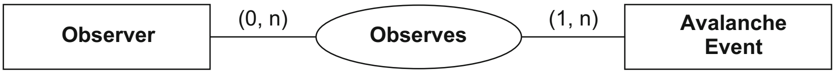
Thematic Data Structure Modeling

Definition 4, 5, and 6 (Object Types, Attributes, and Methods). Database *objects* represent real-world entities of interest to applications. An *object type* defines the properties of interest for a set of objects that, from the application viewpoint, are considered as similar. Properties are either attributes or methods. An *attribute* is a property that is represented by a value in each object of the type. A *method* is a behavioral property common to all objects of the type.

Figure 1 illustrates an object type *Avalanche Event* and its attributes. These may be *simple*,



Modeling and Multiple Perceptions, Fig. 1 A diagram of an object type with its attributes



Modeling and Multiple Perceptions, Fig. 2 A diagram showing a relationship type linking two object types

e.g., *number* for *AvalancheEvent*, or *complex* (i.e., composed of other attributes), e.g., *witnesses*.

Definition 7 (Cardinality). *Attribute cardinality*, defined by two numbers (min, max), denotes the minimum and maximum number of values that an attribute may hold within an object. The maximum cardinality determines whether an attribute is *monovalued* or *multivalued*, i.e., whether it holds at most one or several values. The minimum cardinality determines whether an attribute is *optional* (min = 0) or *mandatory* (min > 0), i.e., whether an object may hold no value or must hold at least one value.

Definition 8 and 9 (Relationship Types and Roles). Relationships represent real-world links between objects that are of interest to the application. Relationships that, from the application perspective, have the same characteristics are grouped in relationship types. Roles represent the involvement of an object type into a relationship type. A relationship type defines two or more roles: They are n-ary, n being the number of roles. Figure 2 shows an example of a binary relationship of the type Observes.

Cardinality constraints define the minimum and maximum number of relationships that may link an object in a role. For example, in Fig. 2 the (0, n) cardinalities on the role associated to the *Observer* object type express that an observer may have never observed an avalanche, and that an observer may have observed any number of avalanches.

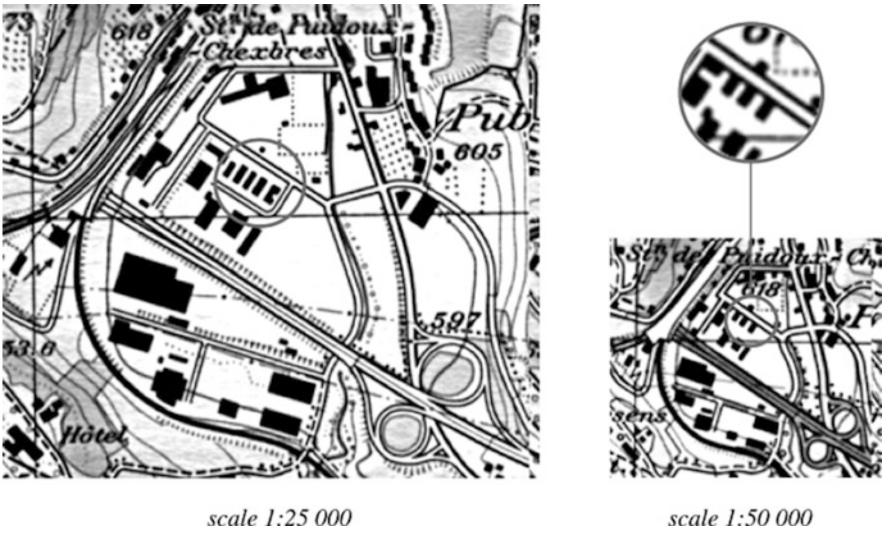
As object types, relationship types may be described by properties (attributes and methods). The generic term *instance* denotes either an object or a relationship: an object (relationship) is an instance of the object (relationship) type it belongs to. MADS identifies two basic kinds of relationship types, association and multiassociation, described next.

Definition 10 (Association Types). An *association type* is a relationship type such that each role links exactly one instance of the linked object type.

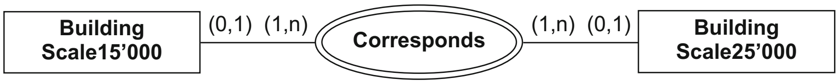
Associations are the usual kind of relationships. However, in some situations the association relationship does not allow accurate representation of real-world links existing between objects. Figure 3 shows two maps of the same area at different scales. Focusing on the area within the superimposed circles, the left-hand, more detailed, map shows five aligned buildings whereas at the same location the right-hand map shows only three. Suppose that the application stores the five buildings in the left-hand map as instances of the *BuildingScale 15'000* object type, and the three buildings in the right-hand map as instances of the *BuildingScale 25'000* object type. If the application requires the correlation of cartographic representations, at different scales, of the same real-world entities, some link must relate the five instances of *BuildingScale 15'000* to the three instances of *BuildingScale 25'000*. The multiassociation construct allows directly representation of such a link.

Definition 11 (Multiassociation Types). A *multiassociation type* is a relationship type such that each role links a nonempty collection of instances of the linked object type.

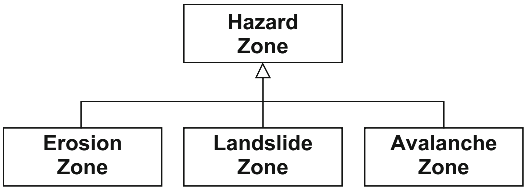
Consequently, each role in a multiassociation type bears two pairs of (min, max) cardinalities. A first pair is the conventional one, defining for each object instance how many relationship instances it can be linked to via the role. The second pair defines for each relationship instance how many object instances it can link with this role. Its value for minimum is at least 1. Using a multiassociation type, the above correspondence between cartographic buildings can be modeled as shown in Fig. 4.



Modeling and Multiple Perceptions, Fig. 3 An example situation calling for a multiassociation relationship



Modeling and Multiple Perceptions, Fig. 4 An example of a multiassociation type



Modeling and Multiple Perceptions, Fig. 5 Object types connected by is-a links

Apart from the additional cardinality constraints, multiassociation types share the same features as association types.

Definition 12 (is-a Links). The *is-a* (or generalization/specialization) link relates two object or two relationship types, a generic one (the *supertype*) and a specific one (the *subtype*). It states that the subtype describes a subset of the real-world instances described by the supertype, and this description is a more precise one.

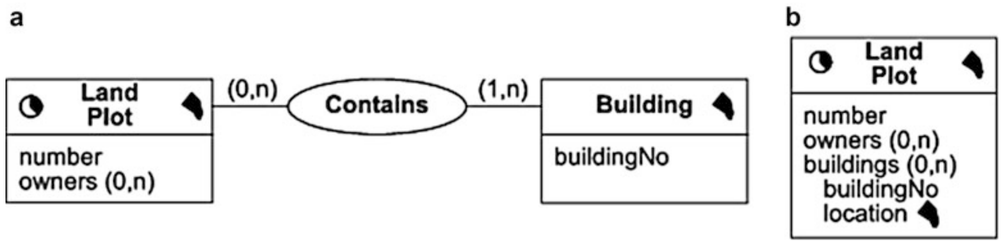
Figure 5 shows a generalization hierarchy of object types with three is-a links. The generic object type representing hazard zones is specialized in subtypes representing landslide, erosion, and avalanche zones.

A well-known characteristic of is-a links is *property inheritance*: All properties and links defined for the supertype also hold for the subtype. An immediate benefit of inheritance is *type substitutability*, i.e., enforcing the fact that wherever an instance of a type can be used in some data manipulation, an instance of any of its subtypes can be used instead.

Describing Space and Time Using the Discrete View

In MADS, space and time description is orthogonal to data structure description, which means that the description of a phenomenon may be enhanced by spatial and temporal features whatever data structure (i.e., object, relationship, attribute) has been chosen to represent it.

Definition 13 (Discrete View). The *discrete* (or *object*) *view* of space and time defines the spatial and temporal extents of the phenomena of interest. The *spatial extent* is the set of points



Modeling and Multiple Perceptions, Fig. 6 (**a**) and (**b**) Alternative schemas for land plots and buildings

that the phenomenon occupies in space, while the *temporal extent* is the set of instants that it occupies in time.

Specific data types support the manipulation of spatial and temporal values, like a point, a surface, or a time instant. MADS supports a hierarchy of spatial data types and another hierarchy of temporal data types. Generic data types allow the description of object types whose instances may have different types of spatial or temporal extents. For example, a *River* object type may contain large rivers with an extent of type *Surface* and small rivers with an extent of type *Line*. Examples of spatial data types are: *Geo* (🌍), the most generic spatial data type, *Surface* (◆), and *SurfaceBag* (◆). The latter is useful for describing objects with a nonconnected surface, like an archipelago. Examples of temporal data types are: *Instant* (◷), *TimeInterval* (◑), and *IntervalBag* (⊗). The latter is useful for describing the periods of activity of noncontinuous phenomena.

In MADS, temporality associated to object/relationship types or to attributes corresponds to *valid time*, which conveys information on when a given fact of the database is considered valid from the application viewpoint.

Definition 14 (Spatial, Temporal, Spatiotemporal Objects and Relationships). A *spatial* (and/or *temporal*) *object type* is an object type that holds spatial (and/or temporal) information pertaining to the object as a whole.

For example, in Fig. 6a both object types are spatial as shown by the *Surface* (◆) icon, while only *LandPlot* is temporal as shown by the *TimeInterval* (◑) icon. Following common practice, an object type is called *spatiotemporal* if either it has both a spatial and a temporal extent, separately, or has a time-varying spatial extent (i.e., its spatial extent changes over time and the history of extent values is recorded).

Similarly, *spatial*, *temporal*, and *spatiotemporal relationship types* hold spatial and/or temporal information pertaining to the relationship as a whole, as for an object type. For example, in Fig. 2, the *Observes* relationship type can be defined as temporal, of the type *Instant*, to record when observations are made.

Spatial and temporal information at the object- or relationship-type level is kept in dedicated system-defined attributes: *geometry* for space and *lifecycle* for time. Geometry is a spatial attribute (see below) with any spatial data type as domain. When representing a moving object, geometry is a time-varying spatial attribute. On the other hand, the lifecycle allows users to record when an object (or link) was (or is planned to be) created and deleted. It may also support recording that an object is temporarily suspended, like an employee who is on temporary leave. Therefore the lifecycle of an instance says for each instant what is the status of the corresponding real-world object (or link) at this instant: scheduled (its creation is planned later), active, suspended (it is temporarily inactive), disabled (definitively inactive).

Definition 15 (Spatial, Temporal, Spatiotemporal Attributes). A *spatial* (*temporal*) *attribute* is a simple attribute whose domain of values is one of the spatial (temporal) data types. A *spatiotemporal attribute* is a time-varying spatial attribute, i.e., a spatial attribute whose value changes over time and the history of its values is recorded (see Definition 17).

Each object and relationship type, whether spatiotemporal or not, may have spatial, temporal, and spatiotemporal attributes. For example, in Fig. 6b the *LandPlot* object type includes, in addition to its spatial extent, a complex and multivalued attribute *buildings* whose second component attribute, *location*, is a spatial attribute describing, for each building, its spatial extent.

Constraining Relationships with Space and Time Predicates

The links among spatial (temporal) object types often describe a spatial (temporal) constraint on the spatial (temporal) extents of the linked objects. For example, designers may want to enforce each *Contains* relationship of Fig. 6a to link a pair of objects provided that the spatial extent of the land plot contains the spatial extent of the building. In MADS, this can be done by defining *Contains* as a constraining relationship of the type topological inclusion, as shown in Fig. 7 by the ⬬ icon.

Definition 16 (Constraining Relationships). *Constraining relationships* are binary relationships enforcing the geometries or lifecycles of the linked objects types to comply with a topological or synchronization constraint.

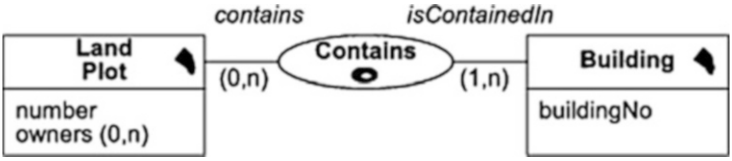
Figure 8 shows a temporal synchronization relationship of the type *During*, stating that observers can observe avalanche events only while they are on duty. Relationship types may simultaneously bear multiple constraining semantics. For example, the *Intersects* relationship type shown in Fig. 9 enforces both a topological overlapping and a synchronization overlapping constraint.

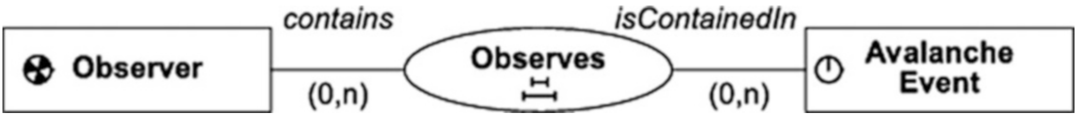
Describing Space and Time Using the Continuous View

Beyond the discrete view, there is a need to support another perception of space and time, the *continuous* (or *field*) *view*.

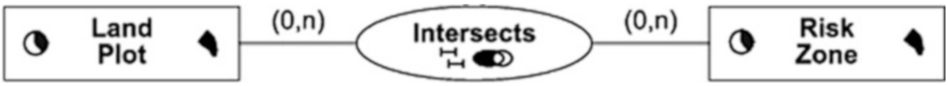
Definition 17 (Continuous View, Varying Attribute). In the *continuous view*, a phenomenon is perceived as a function associating to each point (instant) of a spatial (temporal) extent a value. In the MADS model the continuous view is supported by space (and/or time) *varying attributes*, which are attributes whose value is a function. The domain of the function is a spatial (and/or temporal) extent. Its range can be a set of simple values (e.g., *Real* for temperature, *Integer* for rainfall, *Point* for a moving car), or a set of composite values if the attribute is complex as, in Fig. 10, *weather*. If the attribute is multivalued, the range is a powerset of values.

Modeling and Multiple Perceptions, Fig. 7 A topological relationship of the type *Inclusion* (⬬)

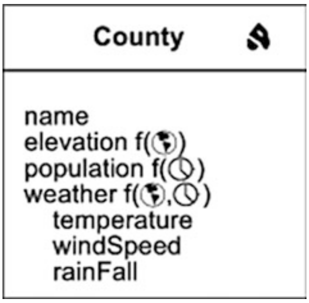




Modeling and Multiple Perceptions, Fig. 8 A synchronization relationship type of the *During* kind (⊢⊣)



Modeling and Multiple Perceptions, Fig. 9 A topological *Overlap* (⬬) and synchronization *Overlap* (⊢⊣) relationship type



Modeling and Multiple Perceptions, Fig. 10 An object type with varying attributes

Figure 10 shows examples of varying attributes and their visual notation in MADS. For instance, *elevation* is a space-varying attribute defined over the geometry of the county. It provides for each geographic point of the county its elevation. An example of a time-varying attribute is *population*, which is defined over a time interval, e.g., [1900, 2008]. Then, *weather* is a space and time-varying attribute which gives for each point of the spatial extent of the county and each instant of a time interval a composite value describing the weather at this location and this instant. Attributes that are space and time-varying are also called *spatiotemporal attributes*.

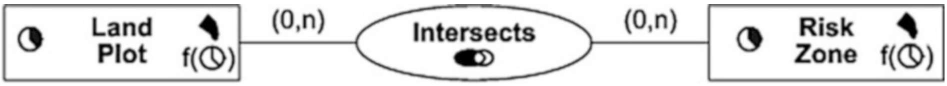
A constraining topological relationship may link moving or deforming objects, i.e., spatial objects whose geometries are time-varying. Figure 11 shows an example. In this case two possible interpretations can be given to the topological predicate, depending on whether it must be satisfied either in at least one instant or in every instant belonging to both time extents of the varying geometries (Erwig and Schneider 2002). Applied to the example of Fig. 11, the two interpretations result in accepting in the relationship *Intersects* only instances that link a land plot and a risk zone such that their geometries intersect for at least one instant or for every instant belonging to both life spans. When defining the relationship type, the designer has to specify which interpretation holds.

Supporting Multiple Perceptions and Multiple Representations

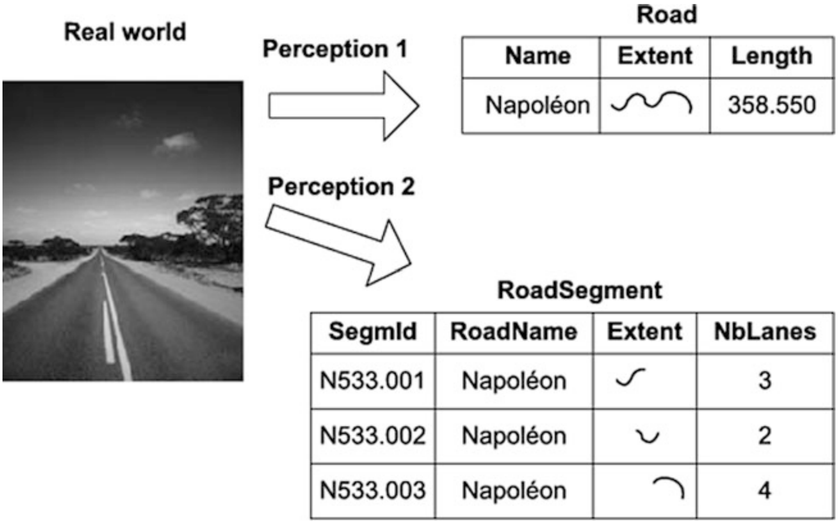
Databases store representations of real-world phenomena that are of interest to a given set of applications. However, while the real world is supposed to be unique, its representation depends on the intended purpose.

Definition 18 (Perceptions and Representations). Each application has a peculiar *perception* of the real world of interest. These perceptions may vary both in terms of what information is to be kept and in terms of how the information is to be represented. Fully coping with such diversity entails that any database element may have several descriptions, or *representations*, each one associated to the perceptions it belongs to. Both metadata (descriptions of objects, relationships, attributes, is-a links) and data (instances and attribute values) may have multiple representations. There is a bidirectional mapping between the set of perceptions and the set of representations. This mapping links each perception to the representations perceived through this perception.

Classic databases do not support the mapping between perceptions and representations. They usually store for each real-world entity or link a unique, generic representation, hosting whatever is needed to globally comply with all application perceptions. An exception exists for databases supporting generalization hierarchies, which allow the storage of several representations of the same entity in increasing levels of specificity. These classic databases have no knowledge of perceptions. Hence the system cannot provide any service related to perception-dependent data



Modeling and Multiple Perceptions, Fig. 11 An example of a topological relationship that links spatial object types with deforming geometries



Road

Name	Extent	Length
Napoléon		358.550

RoadSegment

SegmId	RoadName	Extent	NbLanes
N533.001	Napoléon		3
N533.002	Napoléon		2
N533.003	Napoléon		4

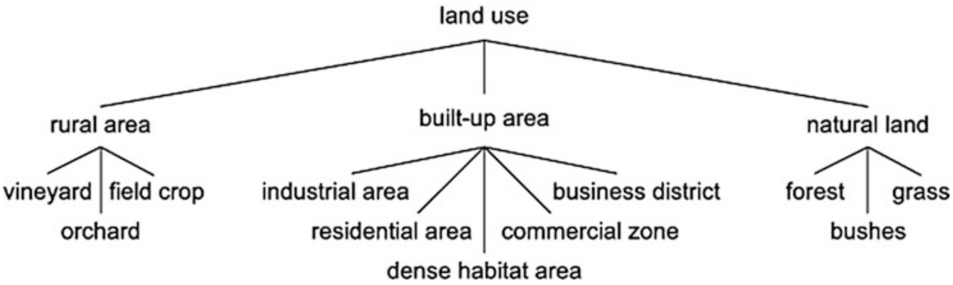
Modeling and Multiple Perceptions, Fig. 12 Two different perceptions of the same reality, leading to different representations

management. Applications have to resort to the view mechanism to define and extract data sets that correspond to their own perception of the database. But still, each view is a relational table or object class. They cannot make up a whole perception, which is a subset of the database containing several object types related by relationship types and is-a links. Instead, MADS explicitly supports multiple perceptions for the same database.

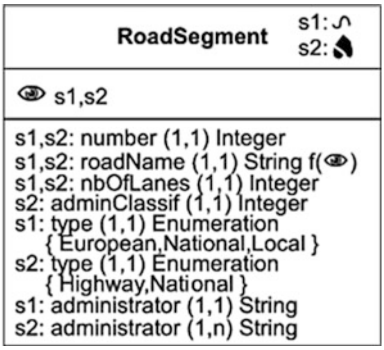
Definition 19 (Multiperception Databases). A *multiperception database* is a database where designers and users have the ability to define and manipulate several perceptions of the database. A multiperception database stores one or several representations for each database element, and records for each perception the representations it is made up.

Geographical applications have strong requirements in terms of multiple representations. For example, cartographic applications need to keep multiple geometries for each object, each geometry corresponding to a representation of the extent of the object at a given *spatial resolution*. The resolution of a spatial database is the minimum size of any spatial extent stored in the database. Resolution is closely related to the scale of the maps that are produced from the database. The scale of a printed map is the amount of reduction between the real world and its graphic representation in the map. Multiscale representations are needed as there is still no complete set of algorithms for *cartographic generalization* (Weibel and Dutton 1999), i.e., the process to automatically derive a representation at a less-detailed resolution from a representation at a more precise resolution.

Geographical databases are also subject to classical semantic resolution differences. For example, an application may see a road as a single object, while another one may see it in more detail as a sequence of road sections, each one represented as an object, as in Fig. 12. As another



Modeling and Multiple Perceptions, Fig. 13 An example of a hierarchical domain


RoadSegment
s1:
s2:
👁 s1,s2
s1,s2: number (1,1) Integer
s1,s2: roadName (1,1) String f(👁)
s1,s2: nbOfLanes (1,1) Integer
s2: adminClassif (1,1) Integer
s1: type (1,1) Enumeration
{ European,National,Local }
s2: type (1,1) Enumeration
{ Highway,National }
s1: administrator (1,1) String
s2: administrator (1,n) String

Modeling and Multiple Perceptions, Fig. 14 An illustration of a birepresentation type, defined for perceptions s1 and s2

example, geographical databases need to support *hierarchical value domains* for attributes, where values are chosen depending on the level of detail. In the hierarchical domain for *land use*, Fig. 13, *orchard* and *rural area* are two representations of the same value at different resolutions.

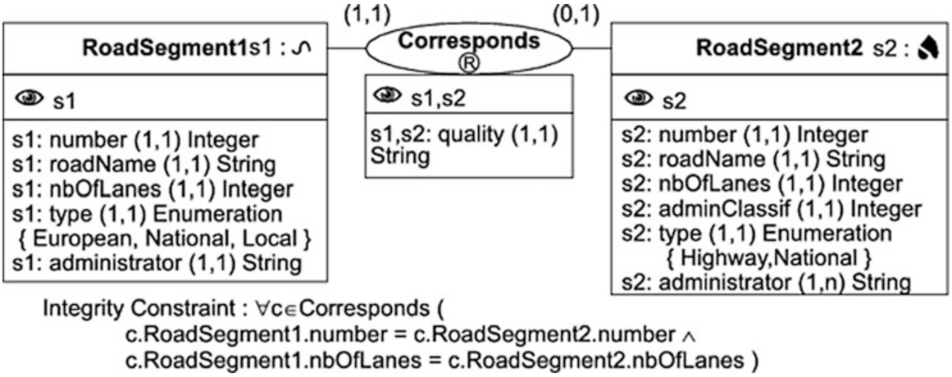
In MADS, each perception has a user-defined identifier, called its *perception stamp*, or just *stamp*. In the sequel, perception stamps are denoted as s1, s2, ..., sn. From data definitions (metadata) to data values, anything in a database (object type, relationship type, attribute, role, instance, value) belongs to one or several perceptions. Stamping an element of the schema or of the database defines for which perceptions the element is relevant. In the diagrams, e.g., Fig. 14, the box identified by the 👁 icon defines the set of perceptions for which this type is valid. Similarly, the specification of the relevant stamps is attached to each attribute and method definition.

There are two complementary techniques to organize multiple representations. One solution is to build a single object type containing several representations, the knowledge of "which representation belongs to which perception" being provided by the stamps of the properties of the type. Following this approach, in Fig. 14 the designer has defined a single object type *RoadSegment*, grouping two representations, one for perception s1 and one for perception s2.

Definition 20 (Multirepresentation Object/Relationship Types). An object or relationship type is *multirepresentation* if at least one of its characteristics has at least two different representations. The characteristic may be at the schema level (e.g., an attribute with different definitions) or at the instance level (i.e., different sets of instances or an instance with two different values).

The alternative solution to organize multiple representations is to define two separate object types, each one bearing the corresponding stamp(s) (cf. Fig. 15). The knowledge that the two representations describe the same entities is then conveyed by linking the object types with a relationship type that holds an *interrepresentation* semantics (indicated by the Ⓡ icon). In the example of Fig. 15, the same real-world road segment is materialized in the database as two object instances, one in *RoadSegment1* and one in *RoadSegment2*. Instances of the relationship type *Corresponds* tell which object instances represent the same road segment.

The actual representation of instances of multirepresentation object types changes from one


(1,1)
(0,1)
RoadSegment1s1 :
Corresponds
®
RoadSegment2 s2 :
s1
s1,s2
s2
s1: number (1,1) Integer
s1: roadName (1,1) String
s1: nbOfLanes (1,1) Integer
s1: type (1,1) Enumeration
{ European, National, Local }
s1: administrator (1,1) String
s1,s2: quality (1,1) String
s2: number (1,1) Integer
s2: roadName (1,1) String
s2: nbOfLanes (1,1) Integer
s2: adminClassif (1,1) Integer
s2: type (1,1) Enumeration
{ Highway,National }
s2: administrator (1,n) String
Integrity Constraint : ∀c∈Corresponds (
c.RoadSegment1.number = c.RoadSegment2.number ∧
c.RoadSegment1.nbOfLanes = c.RoadSegment2.nbOfLanes)

Modeling and Multiple Perceptions, Fig. 15 The *RoadSegment* type (from Fig. 14) split into two monorepresentation object types and an interrepresentation relationship type

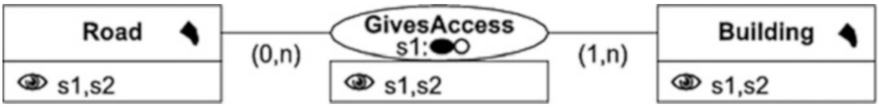
perception to another. Consider the object type *RoadSegment* of Fig. 14. The spatial extent of the type is represented either as a surface (the more precise description, perception s2) or as a line (the less precise description, perception s1) depending on resolution. Furthermore, perception s1 needs the attributes *number*, *roadName*, *numberOfLanes*, *type*, and *administrator* (denoting the maintenance firm in charge). Perception s2 needs the attributes *number*, *roadName*, *numberOfLanes*, *adminClassification*, *type*, and *administrator*. While the road segment number and the number of lanes are the same for s1 and s2, the name of the road is different, although a string in both cases. For instance, the same road may have name "RN85" in perception s1 and name "Route Napoléon" in s2. We call this a *perception-varying attribute* (see below), identified as such by the *f* (👁) notation. The *type* attribute takes its values from predefined sets of values, the sets being different for s1 and s2. Several administrators for a road segment may be recorded for s2, while s1 records only one.

Definition 21 (Perception-Varying Attribute). An attribute is perception-varying if its value in an instance may change from one perception to another. A perception-varying attribute is a function whose domain is the set of perceptions of the object (or relationship) type and whose range is the value domain defined for this attribute. These attributes are the counterpart of space-varying and time-varying attributes in the space and time modeling dimensions.

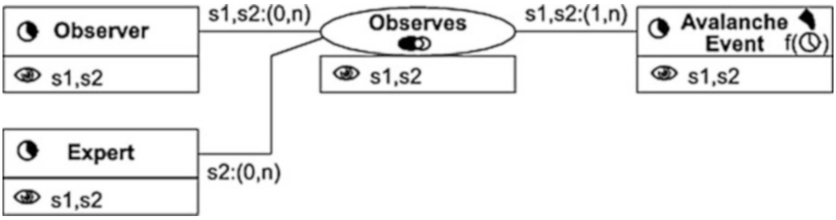
Stamps may also be specified at the instance level. This allows the defining of different subsets of instances that are visible for different perceptions. For example, as the object type *RoadSegment* in Fig. 14 has two stamps, it is possible to define instances that are only visible to s1, instances that are only visible to s2, and instances that are visible to both s1 and s2.

Relationship types can be in multirepresentation, like object types. Their structure (roles and association/multiassociation kind) and semantics (e.g., topology, synchronization) may also have different definitions depending on the perception. Figure 16 shows an example of different semantics, where the designer defined the relationship *GivesAccess* as (1) a topological adjacent relationship type for perception s1, and (2) a plain relationship without any peculiar semantics or constraint for perception s2.

A relationship type may have different roles for different perceptions. For example, Fig. 17 shows that an observation is perceived in s1 as a binary relationship between an observer and an avalanche event, while perception s2 sees the same observation as a ternary relationship, also involving the expert who has validated the observation.



Modeling and Multiple Perceptions, Fig. 16 A relationship type with two different semantics: topological adjacency for s1 and plain relationship for s2



Modeling and Multiple Perceptions, Fig. 17 A relationship type with a role specific to perception s2

Key Applications

Multirepresentation databases are an essential feature for all applications that need to provide different categories of users with different sets of data, organized in different ways, without imposing a single centralized model of the real world for the whole enterprise.

Multirepresentation databases play a key role in guaranteeing and maintaining the consistency of a database with decentralized control and autonomy of updates from different user categories. Multiscale map production applications are one example of a domain where this feature is highly desirable. Management and analysis of municipal, regional, and statewide databases are other examples where the same geographic area is input to a variety of uses. Domain-oriented (e.g., environmental) interorganizational as well as international applications call for integrated databases that provide autonomous usage by the contributing organizations and countries.

Future Directions

Considering the large number of perceptions that some huge databases may have to support, an investigation into how perceptions may be organized and structured (as objects of interest per se) is planned.

Another direction for future research is adding complementary modeling dimensions to the MADS model, such as the multimedia dimension, the precision dimension (supporting, for example, fuzzy, imprecise, and approximate spatial and temporal features), and the trust or data quality dimension.

Finally, spatiotemporal and trajectory data warehousing represent wide-open research domains to develop efficient support for geographically based decision making.

Cross-References

▸ Modeling with ISO 191xx Standards

References

Bédard Y, Bernier É (2002) Supporting multiple representations with spatial databases view management and the concept of VUEL. In: The joint workshop on multiscale representation of spatial data, ISPRS WG IV/3, ICA Commission. On Map Generalization, Ottawa, 7–8 July 2002

Bédard Y, Pageau J, Caron C (1992) Spatial data modeling: the Modul-R formalism and CASE technology. In: Proceedings of the ISPRS symposium, Washington DC, 1–14 Aug 1992

Bédard Y, Larrivée S, Proulx MJ, Nadeau M (2004) Modeling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on perceptory. In: Wang S et al (eds) ER workshops, Shanghai. Lecture notes in computer science, vol 3289. Springer, Berlin/Heidelberg, pp 17–30

Chen PP (1976) The entityrelationship model: towards a unified view of data. ACM Trans Database Syst 1: 9–36

David B, Raynal L, Schorter G (1993) GeO2: why objects in a geographical DBMS? In: Abel DJ, Ooi BC (eds) Proceedings of the 3rd international symposium on advances in spatial databases, SSD'93, Singapore, 23–25 June 1993. Lecture notes in computer science, vol 692, pp 264–276. Springer, Berlin/Heidelberg

Erwig M, Schneider M (2002) Spatiotemporal predicates. IEEE Trans Knowl Data Eng 14:881–901

Parent C, Spaccapietra S, Zimányi E (2006a) Conceptual modeling for traditional and spatiotemporal applications: the MADS approach. Springer, Berlin/Heidelberg

Parent C, Spaccapietra S, Zimányi E (2006b) The MurMur project: modeling and querying multirepresented spatiotemporal databases. Inf Syst 31:733–769

Price R, Ramamohanarao K, Srinivasan B (1999) Spatiotemporal extensions to unified modeling language. In: Proceedings of the workshop on spatiotemporal data models and languages, IEEE DEXA'99 workshop, Florence, 1–3 Sept 1999

Rumbaugh J, Jacobson I, Booch G (2005) The unified modeling language, reference manual, 2nd edn. AddisonWesley, Boston

Tryfona N, Jensen CS (1999) Conceptual data modeling for spatiotemporal application. GeoInformatica 3(3):245–268

Weibel R, Dutton G (1999) Generalizing spatial data and dealing with multiple representations. In: Longley PA, Goodchild MF, Maguire DJ, Rhind DW (eds) Geographical information systems: principles, techniques, management and applications, 2nd edn., vol. 1. Wiley, New York, pp 125–155

Recommended Reading

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

Koubarakis M et al (eds) (2003) Spatiotemporal databases: the chorochronos approach. Lecture notes in computer science, vol 2520. Springer, Berlin/Heidelberg

Malinowski E, Zimányi E (2007, in press) Advanced data warehouse design: from conventional to spatial and temporal applications. Springer, Berlin/Heidelberg

Modeling Cycles in Geospatial Domains

Jorge Campos
Department of Exact Sciences, Salvador
University – UNIFACS, Salvador, Brazil

Synonyms

Event, cyclic; Event, Periodic; Event, Recurrent

Definition

In a geospatial context, cycles can be defined as regularly repeated phenomena or events. According to the distribution of this kind of happening over time, cycles can be classified as *strong periodic*, *near periodic*, or *intermittent*. Strong periodicity refers to cycles in which the duration of every occurrence of the event is the same as is the temporal interval between two consecutive occurrences. Train schedules and tidal movements are examples of events with a strong periodic pattern of repetition. The second class of periodicity deals with events that occur regularly, but the occurrences do not necessarily have the same duration nor are they equally spaced in time. For this kind of cycle, the duration of events and the temporal gap between each of them may vary in a stochastic or deterministic manner. The spill of a geyser is an example of a near periodic event. The third class of periodicity deals with events that occur in a more irregular fashion or not regularly enough to be predicted with any degree of certainty. The occurrences of category 5 hurricanes over the Gulf of Mexico are an example of an intermittent phenomenon.

Main Text

Currently, there are many kinds of specialists dealing with or studying phenomena that repeat regularly. Engineers and social scientists, for example, cooperate to synchronize the schedules of the public transportation system with people's daily routine in urban environments. Environmental, social and financial experts struggle to predict the effect of global warming on seasonal precipitations and, therefore, on the occurrences of floods. Biologists and life scientists are working to identify relations between birds' migration cycles and the recurrent occurrences of some human and animals' diseases. Thus, there are many research questions driving the efforts of a wide range of specialists who deal with cyclic phenomena. In the database area, for example, there are needs for conceptual models and formalisms for expressing periodicity and for constructing queries about periodic data. In knowledge representation and temporal reasoning, there are concerns about periodic-based temporal constraints and modeling cyclic temporal relations (Hornsby et al. 1999), which are important subjects for scheduling and for constraint satisfaction problems involving cyclic events (Isli 2001). In the knowledge discovery and data mining field, there are requests for the support of mining temporal data to discover trends, patterns, and relationships between cyclic recurrent events (Roddick and Spiliopoulou 1999).

The geographic information science research community has had a strong interest in capturing dynamic or time-varying phenomena in geographic space and representing such phenomena in spatio-temporal data models. Most successful models have adopted either spatio-temporal extensions of the entity-relationship or object-oriented models, or an event- and process-based approach. Whichever model is used to represent the temporal dimension of the phenomena, few studies have focused on happenings that repeat themselves in a cyclic manner.

Cross-References

▶ Geographic Dynamics, Visualization and Modeling
▶ Processes and Events
▶ Time Geography

References

Hornsby K, Egenhofer M, Hayes P (1999) Modeling cyclic change. In: Proceedings of the ER'99 workshops on advances in conceptual modeling, Paris. Lecture notes in computer science, vol 1727. Springer, Berlin, pp 98–109

Isli A (2001) On deciding consistency for CSPs of cyclic time intervals. In: Proceedings of the fourteenth international Florida artificial intelligence research society conference, Key West, 21–23 May 2001

Roddick JF, Spiliopoulou M (1999) A bibliography of temporal, spatial and spatiotemporal data mining research. SIGKDD Explor Newsl 1:34–38

Modeling Geospatial Application Database

▶ Modeling with ISO 191xx Standards

Modeling Geospatial Databases

▶ Modeling with ISO 191xx Standards

Modeling the Spread of Infectious Diseases in Global Transport Systems

Lauren M. Gardner
School of Civil and Environmental Engineering,
University of New South Wales, Sydney, NSW,
Australia

Historical Background

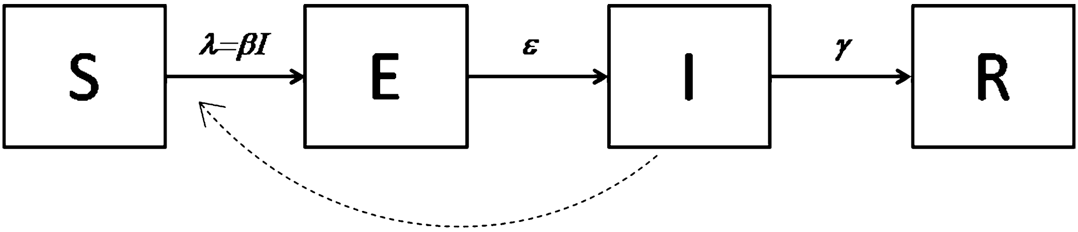
Infectious diseases currently present serious public-health threats worldwide, concerning health systems, governments, industry, and society in general. Prior to the existence of modern transportation networks, natural barriers limited certain diseases to specific geographic regions. However, contemporary global transport systems have connected previously isolated regions, providing a means for pathogens to move around the globe faster and further than ever before. Additionally, a rise in the volume of international air travel has resulted in an increased likelihood of imported infections among travelers into new regions. For these reasons, it is imperative to develop models which quantify the risk of importing infected passengers and vectors into a new region, as well as predict the expected impact an infectious disease may have on a given region once introduced. Furthermore, the extensive range of GIS tools now available offers a means for researchers and public-health authorities to understand and visualize spatial databases including outbreak locations, human mobility networks, and ecological and environmental conditions. Given the spatial and temporal component of infectious disease outbreaks, GIS tools can and should be exploited to develop new and improved prediction models. Such models should be robust, meaning they can be applied to a range of newly emerging outbreaks, and parameterizable, thus adaptable for real-time implementation. If designed properly, these models can be used to aid decision makers in designing optimal public-health policies, such as prioritizing specific travel routes and locations (origin cities, destination airports, etc.) on which to implement passenger surveillance and control strategies.

To accurately model risk of disease spread through global transport systems, there are substantial data requirements and multiple critical components which must be considered. As will become evident from this chapter, a wide range of approaches have been implemented. The best choice of model is highly dependent on the disease type (contact based or vector borne), the objective sought, and the level of available data. The remainder of this chapter will outline some of the current modeling approaches as well as their strengths and weaknesses.

The remainder of this chapter is broken into the following sections: "Scientific Fundamentals," "Key Applications," and "Future Directions." "Scientific Fundamentals" contains an overview of the traditional mathematical models (i.e., compartmental models) used to represent the spread of an infectious disease within a human population. Compartmental models are presented for two different types of infectious diseases, contact based and vector borne, which vary based on their spreading dynamics. Subsequently, species distribution models (SDMs), which are relevant for modeling vector-borne diseases, are briefly discussed. The "Key Applications" section focuses on disease-spreading models at the global scale. Various papers from the literature are reviewed, and a selection of risk models (varying by their methodology) are presented. The "Future Directions" section highlights existing gaps in the literature where further research would be of value.

Scientific Fundamentals

To model the risk of disease spread at a global scale, network analysis and optimization tools can be utilized. In a mathematical modeling context, a network structure is defined by a set of nodes and links. For example, the air traffic network can be represented by a set of nodes which correspond to airports and links which correspond to air travel routes between airports. The maritime freight network can analogously be defined by the set of port and shipping movements between them. For a given mobility network structure, link weights can be defined to represent the risk posed by a given travel route, as either numerical values or a function of network attributes. Link weights can have very simple definitions, such as the volume of passengers using a given air traffic link or volume of sea cargo movements, or they can take a more complex functional form which accounts for both link-specific attributes such as travel volume and distance, as well as node-specific attributes such as regional population, local environmental conditions, the presence of pathogens, regional economic indicators, outbreak size, etc. However, defining route-level risk functions in real time to accurately model emerging epidemics is not a simple task. For a model to be useful to planners, it must be validated, which is in itself a challenging task and highly dependent on available data.



Modeling the Spread of Infectious Diseases in Global Transport Systems, Fig. 1 An SEIR compartmental model illustrating the force of infection, $\boldsymbol{\lambda}$, which is a function of the transmission rate; $\boldsymbol{\beta}$ and proportion of the population infected, *I;* the incubation rate, $\boldsymbol{\varepsilon}$; and recovery rate, $\boldsymbol{\gamma}$

Published methods which identify the risk of infection associated with travel vary from simple patient-based surveys to complex mathematical models. The surveys generally rely on evaluating patient travel histories after they are diagnosed with a specific illness, whereas the mathematical models seek to predict the future epidemic dynamics based on assumptions about the human population, mobility patterns, and characteristics of the disease itself.

This chapter will introduce a subset of the methodologies used to model the spread of disease, which will be further classified as either contact-based or vector-borne disease models. The two classes of diseases display distinctly different dynamics due to the presence of the third-party spreading agent in the vector-borne diseases, and for this reason, modeling them requires different methodological approaches.

Compartmental Models

The stochastic nature of infectious disease transmission poses a significant challenge to predicting the impact that a new disease might have on a population. Over the last 100 years, significant research efforts have focused on predicting the expected spreading behavior of infectious diseases, which exploit characteristics of both population dynamics and the disease itself.

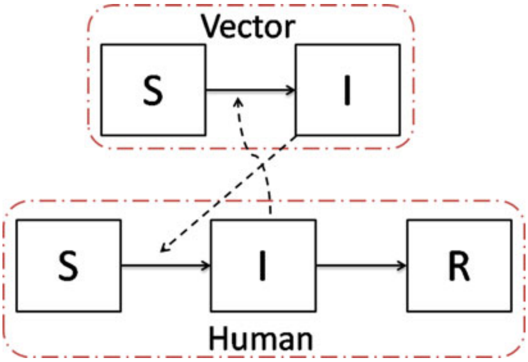
While the focus of this chapter is modeling risk at a global scale, the epidemic dynamics of infectious diseases at a regional scale has a direct impact on the risk posed at the national and global scale. That is, if a disease is highly transmissible, it is more likely to pose a global risk because travelers are more likely to be infected and will also pose greater harm at their travel destination. For these reasons, models which can characterize local outbreak behavior are highly relevant to global risk modeling and will therefore be introduced first.

For diseases which are transmitted through direct human-to-human contact, the progress of an epidemic in a large population can be mathematically represented using a generic compartmental model. The first compartmental models date back to the 1920s, proposed by Kermack and McKendrick (1927), the simplest of which included two main health states, susceptible (S), or previously unexposed to the pathogen, and infected (I), currently colonized by the pathogen. The SI model has since been expanded to include additional states such as recovered (R), or successfully cleared of the infection, and exposed (E), infected but not yet infectious. The flowchart in Fig. 1 illustrates the four compartmental states, the transitional state rates, and corresponding direction of flow through the states for an acute infectious disease, that is, those diseases where the immune system responds "rapidly" to remove pathogens within a short period after infection (days or weeks).

In traditional compartmental modeling, S, E, I, and R are defined as the *proportion* of the total population, N, in each disease state. The transition from S to E depends on the number of infected in the population (relationship is denoted by the dotted line) and can be defined as the force of infection, λ, which is the per capita rate at which susceptible people contact the infection. The force of infection, λ, is equal to the product of I and β, and β is the product of the transmission probability of the pathogen and the contact rate in the population. The transition rate from E to I can be simply defined as ε, the inverse of the average duration of the latent period or the number of days after an individual is infected and before they are infectious. The state change from I to R is based on the recovery rate or the average amount of time spent in the infectious state. The transitional rate of change is often assumed to be a constant, γ, the inverse of the "infectious period." It is important to note that this rate of change is actually variable, and a function of changes in the pathogen itself, as well as various intervention policies. The transitional rates from E to I and I to R can generally be estimated using clinical data, while the rate of change from S to E is much more complex and a function of the epidemic dynamics. Given predefined state transition rates between compartments, the state of the population (number of people falling into each compartment) at any time, t, can be defined using a set of ordinary differential equations.

A compartmental model can also be applied for the local transmission of vector-borne diseases, where a vector is any agent that carries and transmits an infectious pathogen to another living organism and includes mosquitoes, flies, sand flies, lice, fleas, ticks, and mites. Due to the role of the vector in the transmission process, the compartmental models which are used for contact-based diseases cannot be directly applied to vector-borne diseases. A compartmental model representing the infection dynamics of vector-borne diseases (such as a disease spread by mosquitoes) is illustrated by the flowchart in Fig. 2, where the top compartments refer to the states of the vectors and the bottom compartments refer to the states of humans. The transitional rate of change from S to I for both the human and vector is codependent and increases with the proportion of infected humans and vectors in the region. The dependency is a function of the mean rate of bites by a particular vector and the vector-to-human transmission probability per bite.

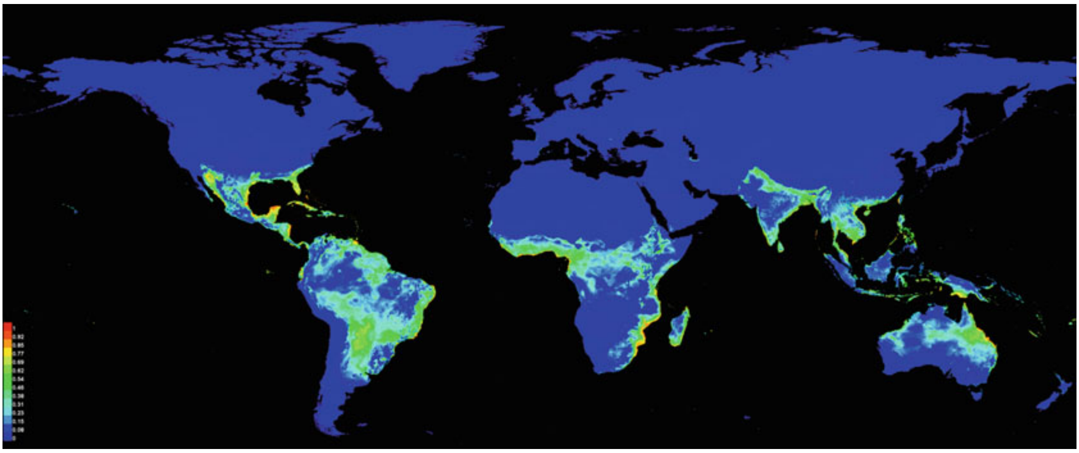
The use of compartmental models allows one to quantify the statistical properties of epidemic patterns by analytically predicting pathogen spread over time. Specific metrics of interest are the prevalence and duration of the epidemic. Additionally, these models can be applied to evaluate potential intervention strategies such as vaccination schemes, which can be accomplished by reducing the number of susceptibles in the population and comparing the epidemic metrics. For more on compartmental models, see May and Anderson (1991).



Modeling the Spread of Infectious Diseases in Global Transport Systems, Fig. 2 An example of a compartmental model for vector-borne diseases, illustrating the relationship between the infection dynamics of the human and vector populations

Species Distribution Models (SDMs)

In addition to the biting rate, a critical component of a vector-borne disease risk model is the likelihood of vector presence in a region. This is because a vector-borne disease cannot spread directly between humans; it must be passed through the vector. The probability of a given vector's presence in a region can be estimated using species distribution models (SDMs). SDMs predict the potential geographical distribution of a species based on occurrence points of a species and environmental data (i.e., climatic and topographic features). SDMs are sometimes interpreted as approximating the ecological niche for a species and can provide a robust framework to analyze the biogeographical determinants of vectorborne diseases (Peterson 2008). The output of SDMs can be interpreted as the probabilistic expectation of vector presence of a species in a given spatial cell. SDM outputs are a critical



Modeling the Spread of Infectious Diseases in Global Transport Systems, Fig. 3 Example of species distribution models for the yellow fever mosquito

component in modeling the risk of arbovirus importation and establishment.

For the last decade, SDMs have typically been constructed using machinelearning algorithms, including a maximum entropy-based software package (Maxent) (Phillips et al. 2006, 2008; Campbell et al. 2015) and boosted regression tree (BRT) models (Kraemer et al. 2015), GARP and BIOCLIM. An example of a global SDM is illustrated in Fig. 3 for the yellow fever mosquito *Aedes Aegypti*, one of the known spreading vectors of dengue and Zika virus. SDMs can correctly predict the known traditional ranges of the species, though they need to be continually validated and refined with updated data on the spread of these species.

Vector presence at both the travel origin and destination is a significant factor in estimating the risk of introduction and establishment of a disease into a region. For example, consider the case where there is an outbreak of dengue on a small island in the Caribbean. Assume, for simplification, there are only two departing travel routes with equal travel volumes; one route departs to New Orleans, LA, in the southern USA where the dengue vector is well established, and the other route departs to Toronto, ON in Canada, where the local conditions are unsuitable for harboring the same vector species. In this example, there is significantly less harm posed to Toronto because it is unlikely an individual infected with dengue will arrive in Toronto and further spread the disease. In contrast, a dengue-infected passenger arriving in New Orleans could be bitten by a local mosquito which could then spread the infection to other humans. Furthermore, if there is a large enough influx of infected cases in a given region at once, an autochthonous cycle could result. Thus, the inclusion of SDM data in disease models is critical for estimating the harm posed to a region.

Key Applications

So far, the focus of this chapter has been on modeling the infection dynamics between humans (and vectors) at a local scale. The local-level models are necessary, but not sufficient to model the risk of disease spread and establishment at a global scale. As was the case with the local-level models, contact-based and vector-borne diseases require separate modeling approaches at the global scale. Various methodologies have been proposed for each type of disease; a subset of which will be introduced in the following sections. The examples introduced are intended to provide the reader with a basic understanding of the critical components of the problem and some possible methodological approaches that have been used. It is however important for the reader to recognize that the models presented here represent a limited selection from a rapidly evolving field of literature. A comprehensive review of the literature is beyond the scope of this work.

The Global Spread of Contact-Based Diseases

In efforts to encapsulate the spread of disease through global transport systems in conjunction with ongoing disease propagation within a local population, compartmental models have been applied within a multilayer framework where the layers represent different levels of human mobility, such as air travel and daily commuting patterns. These models allow for a global-level analysis of future outbreaks and the ability to evaluate relevant intervention strategies such as closing airports, reducing air travel, closing schools, and limiting the number of people going to work, among other quarantine efforts.

Global-scale compartmental models have been applied in both an analytical and simulation-based framework. Analytical models are appealing because they require minimal computation efforts and can provide quantitative metrics about outbreak dynamics (Rvachev and Longini 1985; Balcan et al. 2009; Colizza et al. 2006). However, analytical models require simplifying assumptions to be made about the population structure and interaction dynamics and are therefore unable to incorporate information about explicit network structure properties.

Agent-based simulation, or individual-based models (IBMs), on the other hand, can incorporate specific network structure properties, i.e., contacts links between individual nodes, into their models, and utilizes agent-based simulation to recreate outbreak scenarios. The most advanced IBMs incorporate social contact data, as well as regional and international travel data in efforts to replicate interaction dynamics among "connected" populations at a global scale. The value of IBMs is they are able to replicate multiple possible spreading scenarios, predict average spreading behavior, and analyze various intervention strategies for a given network structure and disease. However, while they can capture a greater degree of detail in their predictions and analysis compared with analytical models, they require a highly detailed set of input data and significant computational resources (Balcan et al. 2010; Eubank et al. 2004; Broeck et al. 2011; Ajelli et al. 2010). Furthermore, due to the inherently stochastic nature of a disease outbreak, multiple simulations are required to compute *expected* outcomes. Given the data requirements and required run time, IBMs can be expensive and computationally taxing, but they have the potential to provide a greater degree of realism, and a means to evaluate very specific control strategies, and are therefore an invaluable tool for planning.

In addition to analytical models and IBMs, which provide insights into the expected spreading behavior of an outbreak, there currently exists a growing demand for a new paradigm of models which exploit the increasingly available real-time epidemiological, spatial, and clinical data in order to evaluate an ongoing outbreak and advise on real-time control strategies. Recent advances in scenario-based modeling have begun addressing this issue. For example, there are various models which use genetic sequence data to analytically infer the geographic history of a given virus' migration (Drummond and Rambaut 2007; Haydon et al. 2003; Jombart et al. 2009). A methodology presented by Gardner et al. (2014) was designed to infer outbreak patterns in social contact networks using case-report data and disease-specific properties by identifying the maximum probability spanning tree (MPST), and further extended in Fajardo and Gardner (2013) and Rey et al. (2015), to consider the case of partial case information availability. A similar model was implemented on a global air traffic network by Gardner et al. (2012) to infer the most likely air travel routes responsible for spreading the 2009 H1N1 (swine) influenza pandemic within the USA. These works represent a growing field of research which seeks to exploit network optimization tools and the types of real-time infection data and are becoming increasingly available during the onset of outbreaks.

The Global Spread of Vector-Borne Diseases

To model vector-borne disease spread at a global scale, analysis of surveillance and monitoring policies must consider the possible infection pathways of locally established vector populations becoming infected from new hosts (e.g., infected air travelers) and subsequently spreading the virus. Furthermore, the sustained presence of an arbovirus (a virus which is spread by arthropod vectors) in a (destination) region is dependent on the local ecological and environmental conditions, specifically existing populations of the spreading vector.

Both transport systems and climate change have contributed to the introduction and expansion of new disease carrying vector populations into previously uninhabited regions. Transport systems such as international air travel and maritime freight provide new intercontinental pathways for vectors, while climate change and urbanization result in extended transmission seasons and more suitable environments for the vectors. Furthermore, a greater likelihood of imported vector-borne infections among travelers returning to previously unexposed regions is expected in the near future due to a warming climate, increased arboviral activity in Asian and Pacific nations, in conjunction with increased travel to and from these regions.

Two different methodological approaches to estimate the global risk model of vector-borne disease spread are presented below. The first approach defines the link-level risk a priori as a function of variables and estimates the risk of disease spread on a relative scale across the network. The second approach is optimization based and instead seeks to calibrate a functional form that represents the link-level risk. Both models utilize data on the transport system, regional economic attributes, SDMs, and/or ecological and environmental factors.

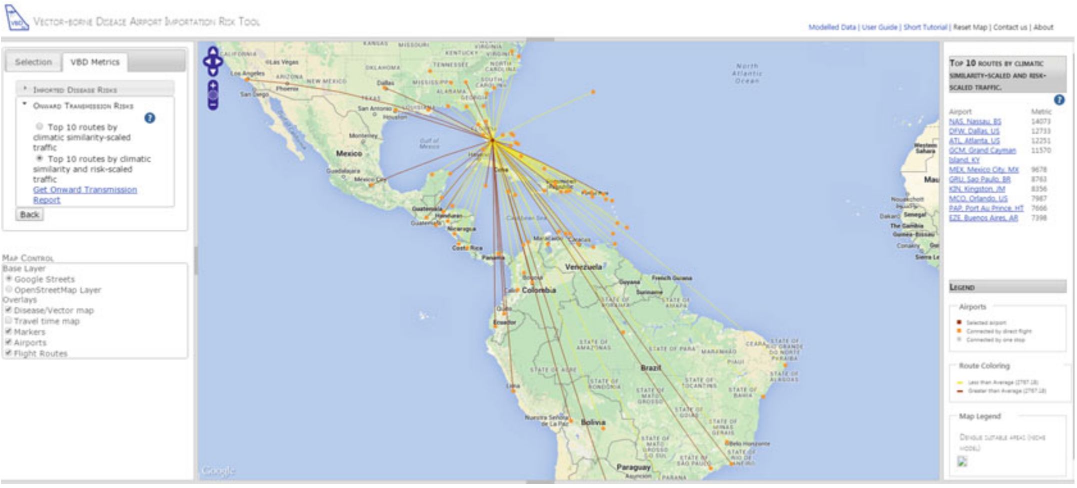
Relative Risk Models

One approach to quantify travel risk (at a route level) utilizing the type of data noted previously is to define the link-level risk as a product of variables. An example of such a relative risk model is the Vector-Borne Disease Airline Importation Risk Tool (VBD-AIR: http://www.vbd-air.com/), which is a web-based GIS tool for defining the role of airports and airlines in the transmission and spread of vector-borne diseases, including malaria, dengue, yellow fever, and chikungunya (Huang 2012). In this tool, the total volume of travel was determined by the passenger volume for air travel, and the climatic similarity was calculated as a distance-based vector. The risk of imported infections, imported vectors, and onward transmission is computed as the product of the *climate Euclidean distance* (CED – defined below), traffic capacity, and disease/vector prevalence at origin locations, which serves as an estimate for the relative risks between scheduled routes of incoming flights between origin i and destination j bringing exotic disease vectors and their consequent establishment. The risk function is computed as shown in Eq. (1):

$$R_{ij} = \frac{1}{(r_i - r_j)^2 + (t_i - t_j)^2 + (h_i - h_j)^2} v_{ij}\, e_i \tag{1}$$

where R_{ij} is the relative risk between origin i and destination j; r, t, and h are the monthly rainfall levels, temperature, and humidity, respectively, at the corresponding airport; v_{ij} is the air traffic capacity between i and j; and e_i is the endemicity at the origin or probability of vector presence. The first part of the function is referred to as the climate Euclidean distance (CED). Each of the three parts is individually normalized before being multiplied to compute the relative risk (Huang 2012); thus, R_{ij} always falls between 0 and 1. These results can be used to identify the set of travel routes entering a given airport which pose the highest risk to that airport and more generally to aid planners and decision makers responsible for allocating limited surveillance resources. An example of the tool visualization is presented in Fig. 4. This screenshot identifies all direct travel routes into Miami International Airport originating in dengue-endemic regions, as well as a list of the top ten routes which pose the highest risk.

Another global risk model was published by Gardner and Sarkar (2013), with the objective


Vector-borne Disease Airport Importation Risk Tool
Modelled Data | User Guide | Short Tutorial | Reset Map | Contact us | About
Selection
VBD Metrics
Imported Disease Risks
Onward Transmission Risks
○ Top 10 routes by climatic similarity-scaled traffic
● Top 10 routes by climatic similarity and risk-scaled traffic
Get Onward Transmission Report
Back
Map Control
Base Layer
● Google Streets
○ OpenStreetMap Layer
Overlays
☑ Disease/Vector map
☐ Travel time map
☑ Markers
☑ Airports
☑ Flight Routes
Mexico
Venezuela
Colombia
Brazil
Bolivia
Paraguay
North Atlantic Ocean
Legend
Airports
Selected airport
Connected by direct flight
Connected by one stop
Route Coloring
Less than Average (2787.18)
Greater than Average (2787.18)
Map Legend
Dengue suitable areas (niche model)
Google

Top 10 routes by climatic similarity-scaled and risk-scaled traffic.

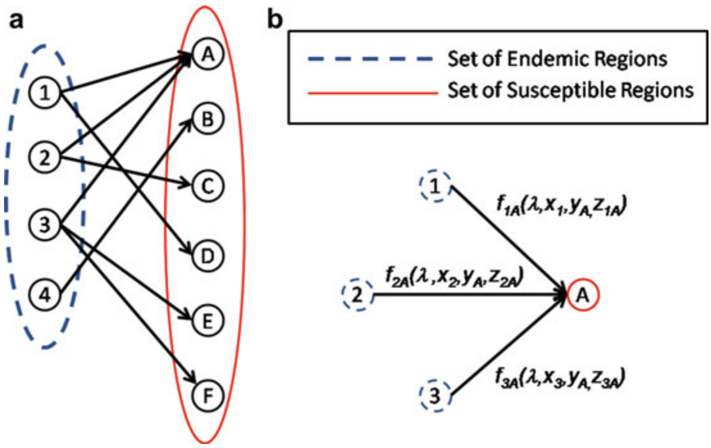
Airport	Metric
NAS, Nassau, BS	14073
DFW, Dallas, US	12733
ATL, Atlanta, US	12251
GCM, Grand Cayman Island, KY	11570
MEX, Mexico City, MX	9678
GRU, Sao Paulo, BR	8763
KIN, Kingston, JM	8356
MCO, Orlando, US	7987
PAP, Port Au Prince, HT	7666
EZE, Buenos Aires, AR	7398

Modeling the Spread of Infectious Diseases in Global Transport Systems, Fig. 4 Screenshot of the VBA-AIR tool, identifying all direct travel routes into Miami International Airport originating in dengue-endemic regions and a list of the top ten highest risk routes (http://www.vbd-air.com/)

of quantifying the relative risk of vector-borne disease spread by infected travelers arriving at or traveling through any given world airport. Similar to the network definitions in the previously introduced models, airports are represented as nodes, and the links in the network represent directed air travel connections between airports. One major difference between this model and the model previously introduced by Huang et al. (2012) is the inclusion of vector suitability data from SDM outputs at both the travel origin and destination. In the model proposed by Gardner and Sarkar (2013), the harm posed to a destination airport j from travel originating at airport i is defined in Eq. (2):

$$u_{ij} = \frac{\varepsilon_i s_i \sigma_i v_{ij} \alpha_j s_j}{D_{ij}}. \qquad (2)$$

Equation (2) is specific to the origin-destination (OD) pair i, j, and is dependent on the origin being in an endemic region (this is disease specific), the outbreak intensity at the origin (σ_i), the suitability at the origin (s_i), the total passenger volume (v_{ij}) traveling between i, j, the population at the destination (α_j), the suitability at the destination (s_j), and the travel distance (D_{ij}). The binary variable, ε_i, is set to 1 for all airports in endemic regions. The origin suitability, s_i, represents the relative ecological risk of the spreading vector (e.g., *Aedes aegypti* or *Aedes albopictus* – the Asian tiger mosquito) being present at the origin, provides a measure of the likelihood of an outgoing traveler being infected, and is computed from SDM models. The destination suitability, s_j, is included because, in order for a disease to spread further after introduction into a new region by an infected traveler, the destination habitat must be ecologically suitable for an insect vector population to establish itself. In the model presented in the paper, the expectations were aggregated to the city level by averaging overall the cells in each geographical unit to define the relative ecological risk in each city. The outbreak intensity, σ_i, is a function of the outbreak size and population density at the origin, which is assumed to be correlated with the probabilistic expectation that an outgoing traveler would be infected. The passenger flow variable, v_{ij}, or the total passenger volume originating at airport i and traveling to airport j captures the potential dispersal for the disease and includes travel on both direct routes and indirect routes with stopovers. The population at the destination, α_j, is a measure of the threat posed to a given region from the disease. The risks are normalized by dividing the highest value computed over all i, j combinations; thus, what is being


a
1
2
3
4
A
B
C
D
E
F
b
Set of Endemic Regions
Set of Susceptible Regions
1
2
3
A
$f_{1A}(\lambda,x_1,y_A,z_{1A})$
$f_{2A}(\lambda,x_2,y_A,z_{2A})$
$f_{3A}(\lambda,x_3,y_A,z_{3A})$

Modeling the Spread of Infectious Diseases in Global Transport Systems, Fig. 5 (**a**) Bipartite network connecting endemic regions to susceptible regions: the susceptible US and Europe nodes represent mutually exclusive sets. (**b**) Link-based functions: these predict the number of infections at susceptible node A, attributed to each adjacent endemic region (1, 2, and 3)

estimated is the *relative expected harm* posed to a destination airport j from travel originating at airport i. A similar model was proposed in the same paper to quantify the harm posed to stopover airports through traffic. Similar models are applied to estimate the relative risk of dengue spread posed by travelers out of the Philippines after Typhoon Haiyan (Gardner and Sarkar 2015) and to estimate the relative risk of Zika spread from Latin America in the 2015–2016 epidemic (Gardner et al. 2016).

Optimization-Based Modeling Framework

Optimization-based modeling frameworks are increasingly being applied to the field of epidemiology. As an alternative to the relative risk models discussed prior, which define the risk functions a priori, optimization methods can be applied to estimate risk functions for a given network (Gardner et al. 2012; Bóta et al. 2014). These models utilize available spatial-temporal case data and network properties to estimate risk functions, which can then be used to predict the likelihood of further disease spread in the network. (Optimization methods can also be applied in a planning context (Chen et al. 2016) to help make real-time control decisions at early stages in an outbreak, with limited information. However, this is a topic beyond the scope of this chapter.)

An optimization-based methodology which estimates risk functions for air travel routes likely to spread dengue into new regions is presented in Gardner et al. (2012). The outcome of the network model provides the expected number of dengue cases in each non-endemic region that can be attributed to a particular endemic region connected to it. In the proposed network structure, geographic areas are represented as nodes, belonging to either the set N of endemic nodes or the set G of susceptible nodes. The links in the network represent directed air travel connections between geographic areas (originating from G), while the measure P_{ji} represents the number of predicted infections at a susceptible node i attributed to an endemic node j. A directed bipartite network structure is used to connect the endemic countries to susceptible regions through directed arcs. Figure 5a provides an illustration of the bipartite network structure. Figure 5b illustrates a four-node extraction from the network to illustrate the generalized link-based functional form used in the model. The function f_{ji} $(\lambda, x_j, y_i, z_{ji})$ represents the number of cases observed at i for which j is responsible, where λ represents a vector of *calibrated* parameters, x_j represents the characteristics of origin j, y_i represents the characteristics of destination i, and z_{ji} represents the vector of parameters specific to directed link (j, i). The total predicted number of infections at i is given by

$$P_i = \Sigma_{\forall\, j \in A\ (i)}\ f_{ji}\ (\lambda, x_j, y_i, z_{ji}),$$

where $A(i)$ represents the set of endemic nodes adjacent to i.

The model seeks to find the unknown parameter vector λ. Attributes included in the function are travel volumes, outbreak data at the origin of travel, destination population size, and habitat suitability for the spreading vector (e.g., based on SDMs) at both ends of the route. The generic problem formulation is as follows:

$$\min_{\lambda} \sum_{\forall i \in N} (I_i - P_i)^2$$

$$\text{s.t.}$$

$$P_{ji} = f_{ji}\,(\lambda, x_j, y_i, z_{ji}) \quad \forall\, i \in N\ \forall j \in G \tag{3}$$

$$P_i = \Sigma_{\forall j \in A\,(i)}\, P_{ji} \qquad \forall\, i \in N \tag{4}$$

This type of model can be easily extended to model risk posed by maritime trade, as well as extended to model the risk posed by alternative vectors and diseases. The model is quantitatively calibrated using actual infection reports, thus providing a more reliable estimate of risk. However, the dependency on case data also represents the main limitation of this type of model; without complete infection data, the model cannot be properly calibrated. In addition to calibration, an additional challenge faced by this model as well as all the models presented in this chapter is validation. Proper validation of risk estimates would require comprehensive infection data on the actual transmission paths of the disease (i.e., travel routes which infected individuals were on and scenarios which resulted in further spread at the destination). Continued model development can help to provide guidance to public health authorities on the most valuable type of data collection which will in turn enhance the predictive accuracy of such models.

Future Directions

Increased international air travel volumes have increased the risk of introducing infectious diseases into new regions and thus increased the demand for quantifiable models to accurately predict disease-spreading behavior at the global scale. Typically, epidemiological models are used as a planning tool in preparation for possible pandemics, before an emerging infection event, and often rely on assumptions which cannot always be verified prior to the event. However, the inherent stochasticity of disease spreading makes it impossible to anticipate all outbreak scenarios. Experiences from the 2009 H1N1 pandemic, H5N1 and H7N9 avian influenzas, SARS, MERS-CoV, and Ebola, among others, have heightened concerns about possible severe global outbreaks of emerging infectious diseases and have highlighted the need for models which can be used in implementation of real-time containment and control strategies.

As illustrated in this chapter, network-based mathematical models can be prepared to aid in analysis and understanding of emerging infectious outbreaks. However, new methodologies are required which can be efficiently implemented during the acute phase of epidemics. Specifically, models which exploit the types of information now available from organizations such as the World Health Organization (WHO) and the International Health Regulations (IHR) including case reports, epidemiological and clinical characteristics, and laboratory testing, are in high demand and necessary to inform disease control and surveillance efforts in real time.

Future research should aim to further strengthen mathematical modeling of epidemiological risk assessment by developing an integrated risk model for real-time spatial and temporal tracking of infectious diseases. Such models should utilize real-time case reports and simultaneously incorporate contact networks, spatial networks, species distribution models, and multimodal transport systems, in efforts to capture the interaction between global transport systems and local transmission risk. The development of real-time predictive models are necessary to inform policy and practice through identifying infectious disease control needs and mitigate the burden of infectious diseases imported through travel.

References

Ajelli M, Goncalves B, Balcan D, Colizza V, Hu H, Ramasco J, Merler S, Vespignani A (2010) Comparing large-scale computational approaches to epidemic modeling: agent-based versus structured metapopulation models. BMC Infect Dis 10:190

Balcan D, Colizza V, Gonçalves B, Hu H, Ramasco JJ, Vespignani A (2009) Multiscale mobility networks and the spatial spreading of infectious diseases. Proc Natl Acad Sci 106(51):21484–21489

Balcan D, Gonçalves B, Hu H, Ramasco JJ, Colizza V, Vespignani A (2010) Modeling the spatial spread of infectious diseases: the GLobal Epidemic and Mobility computational model. J Comput Sci 1:132–145

Bóta A, Krész M, Pluhár A (2014) The inverse infection problem. In: Proceedings of the 2014 federated conference on computer science and information systems, ACSIS, vol 2, pp 75–83. doi:10.15439/2014F261

Broeck WV, Gioannini C, Gonçalves B, Quaggiotto M, Colizza V, Vespignani A (2011) The gleamviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale. BMC Infect Dis 11(1):37

Campbell LP, Luther C, Moo-Llanes D, Ramsey JM, Danis-Lozano R, Peterson AT (2015) Climate change influences on global distributions of dengue and chikungunya virus vectors. Philos Trans R Soc Lond B Biol Sci 370:1665

Chen N, Gardner L, Rey D (2016) A bi-level optimization model for the development of real-time strategies to minimize epidemic spreading risk in air traffic networks. Transp Res Rec J Transp Res Board 2569. doi:10.3141/2569-07

Colizza V, Barrat A, Barthelemy M, Vespignani A (2006) The modeling of global epidemics: stochastic dynamics and predictability. Bull Math Biol 68(8):1893–1921

Drummond AJ, Rambaut A (2007) BEAST: Bayesian evolutionary analysis by sampling trees. BMC Evol Biol 7(1):214

Eubank S, Guclu H, Kumar VS, Marathe M, Srinivassan A, Toroczkai Z, Wang N (2004) Modeling disease outbreaks in realistic urban social networks. Nature 429:180–184

Fajardo D, Gardner LM (2013) Inferring contagion patterns in social contact networks with limited infection data. Netw Spat Econ 1–28. doi:10.1007/s11067-013-9186-6

Gardner L, Sarkar S (2013) A global airport-based risk model for the spread of dengue infection via the air transport network. PloS one 8(8):e72129

Gardner L, Sarkar S (2015) Risk of Dengue spread from the Philippines through international air travel. Transp Res Rec J Transp Res Board 2501:25–30. doi:10.3141/2501-04

Gardner L, Fajardo D, Waller ST (2012) Inferring infection-spreading links in an air traffic network. Transp Res Rec J Transp Res Board 2300(1):13–21

Gardner L, Fajardo D, Waller ST (2014) Inferring contagion patterns in social contact networks using a maximum likelihood approach. ASCE Nat Hazards Rev. doi:10.1061/(ASCE)NH.1527-6996.0000135

Gardner L, Chen N, Sarkar S (2016) Global risk of Zika virus depends critically on vector status of *Aedes albopictus* [Letter]. Lancet Infect Dis. Accepted for Publication 11 Mar 2016. Published online 17 Mar 2016. http://dx.doi.org/10.1016/S1473-3099(16)00176-6

Haydon DT, Chase-Topping M, Shaw DJ, Matthews L, Friar JK, Wilesmith J, Woolhouse MEJ (2003) The construction and analysis of epidemic trees with reference to the 2001 UK foot–and–mouth outbreak. Biol Sci 270(1511):121–127

Huang Z, Das A, Qiu Y, Tatem AJ (2012) Web-based GIS: the vector-borne disease airline importation risk (VBD-AIR) tool. Int J Health Geogr 11:33. doi:10.1186/1476-072X-11-33

Jombart T, Eggo RM, Dodd P, Balloux F (2009) Spatiotemporal dynamics in the early stages of the 2009 A/H1N1 influenza pandemic. PLoS Curr 1: RRN1026

Kermack WO, McKendrick AG (1927) A contribution to the mathematical theory of epidemics. Proc R Soc A 115(772):700–721

Kraemer MU, Sinka ME, Duda KA, Mylne AQ, Shearer FM, Barker CM et al (2015) The global distribution of the arbovirus vectors *Aedes aegypti* and *Ae. albopictus*. eLife 4:e08347. doi:10.7554/eLife.08347

May RM, Anderson RM (1991) Infectious diseases of humans: dynamics and control. Oxford University Press, Oxford [Oxfordshire]. ISBN:0-19-854040-X

Peterson AT (2008) Biogeography of diseases: a framework for analysis. Naturwissenschaften 95(6):483–491. doi:10.1007/s00114-008-0352-5

Phillips SJ, Schapire RE, Anderson RP (2006) Maximum entropy modeling of species geographic distributions. Ecol Model 190(3–4):231–259. Available from: doi:10.1016/j.ecolmodel.2005.03.026

Rey D, Gardner L, Waller ST (2015) Finding outbreak trees in networks with limited information. Netw Spat Econ. doi:10.1007/s11067-015-9294-6

Rvachev L, Longini IM (1985) A mathematical model for the global spread of influenza. Math Biosci 75(1): 3–22

Recommended Reading

Gardner L, Fajardo D, Waller ST, Wang O, Sarkar S (2012) A predictive spatial model to quantify the risk of air-travel-associated Dengue importation into the United States and Europe. J Trop Med. Article ID 103679, 11p

Margules C, Sarkar S (2007) Systematic conservation planning. Cambridge University Press, Cambridge

Phillips SJ, Dudik M (2008) Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation. Ecography 31:161–175

Modeling with a UML Profile

Jugurta Lisboa Filho[1] and Cirano Iochpe[2]
[1]Department of Computer Science/Information Systems, Federal University of Vicosa (UFV), Vicosa, Brazil
[2]Institute of Computer Science/Information Systems, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil

Synonyms

Geographic Database Conceptual Modeling; Unified Modeling Language-Geoframe Modeling Language

Definition

A spatial database management system (SDBMS) provides storage structures and basic operations for spatial data manipulation, whereas geographic information systems (GIS) provide the mechanisms for analysis and visualization of geographic data (Shekhar and Chawla 2003). In this way, geographic databases (GeoDB) are collections of georeferenced spatial data, stored by SDBMS and manipulated by GIS.

GeoDB, as any database, must be designed following the traditional database design methodology that includes the conceptual, logical and physical design phases (Elmasri and Navathe 2000). To draw up a data schema during the conceptual phase, a conceptual modeling language must be used. A strong tendency exists in computer science to adopt the *Unified Modeling Language* (UML) (OMG 2007) as a system modeling standard based on the object-oriented paradigm, and more specifically the UML class diagram for database design. However, for GeoBD design, it is necessary to extend UML with new elements that enable the modeling of spatial-temporal characteristics of geographical phenomena. UML is a naturally extensible language, in other words, it has its own constructs allowing its extension. The stereotype concept, one of the *UML extension mechanisms*, allows the definition of new specific model elements generating a profile tailored for a particular problem domain (OMG 2007). There are some UML extensions for GeoDB modeling (Bédard et al. 2004; Borges et al. 2001; Lisboa Filho and Iochpe 1999). To exemplify a spatial UML profile, described here is the Spatialtemporal UML-GeoFrame modeling language, which extends the UML, generating a profile of stereotypes to support the GeoDB conceptual modeling.

Historical Background

GIS were originated outside the computer science field, unlike most software technologies developed in the last decades such as the operating systems based on windows, DBMS, fourth generation languages, CAD, CAM and CASE tools, Office Automation Systems (OIS), and more recently the World Wide Web (WWW) with the software revolution due to the explosion of internet use.

One of the consequences of this historical origin is that most GIS application designers are their own users, who have the evolutionary approach as their main software developmental methodology, and whose main focus of attention is geospatial data acquisition and analysis. Thus, the old raster-vector debate (Couclelis 1992) prevailed for a long time as an important theme in GIS conferences. Consequently, methodologies developed in the software engineering field are frequently not used in GIS application design, causing great losses in the quality of the produced systems and high maintenance costs.

An alternative to reduce these problems is the use of a database design methodology. Consequently, during the 1990s, several extensions of specific conceptual modeling languages for GIS applications were proposed in the literature. Initially these modeling languages were based on the entity-relationship model (ER), proposed by Peter Chen (1976) or one of its extensions (e.g., Merise and Enhanced Entity-Relationship, EER). A few modeling languages were based on the semantic data model IFO (Abiteboul and Hull

1987). At that time, the use of the object-oriented paradigm in system development was becoming more and more popular. Accordingly, several authors used as their base object-oriented design methods, such as OMT (Rumbaugh et al. 1991) and OOA (Coad and Yourdon 1991), proposing extensions for the modeling of spatiotemporal aspects of geographical phenomena (Bédard et al. 2004; Borges et al. 2001; Lisboa Filho and Iochpe 1999).

With the aim of standardizing the different existent graphic notations and defining a basic group of model constructs for software systems, in 1996 three great experts on object-oriented modeling joined their approaches to create the UML (Booch et al. 1998). Consequently, by 1999 some UML extensions to GeoDB modeling came out, some of them supported by CASE tools (e.g., Perceptory Bédard et al. 2004, ArgoCASEGEO Lisboa Filho et al. 2004).

The UML-GeoFrame modeling language (Lisboa Filho and Iochpe 1999) is described here, to exemplify a spatial UML profile, and show how UML can be naturally extended by its own extension mechanism, named stereotype. Unlike other conceptual modeling languages that seek constructs' completeness, so as to consider almost all modeling possibilities of geographical phenomena in different dimensions (descriptive, spatial and temporal), the UML-GeoFrame has as its inspiration the simplicity of the ER model and proposes the smallest possible group of stereotypes to assist the main requirements of GeoDB modeling, but at the same time allowing understanding by nonspecialized users, through a quite simple and instinctive graphic notation.

Scientific Fundamentals

UML-GeoFrame: A Modeling Language for Geographic Databases

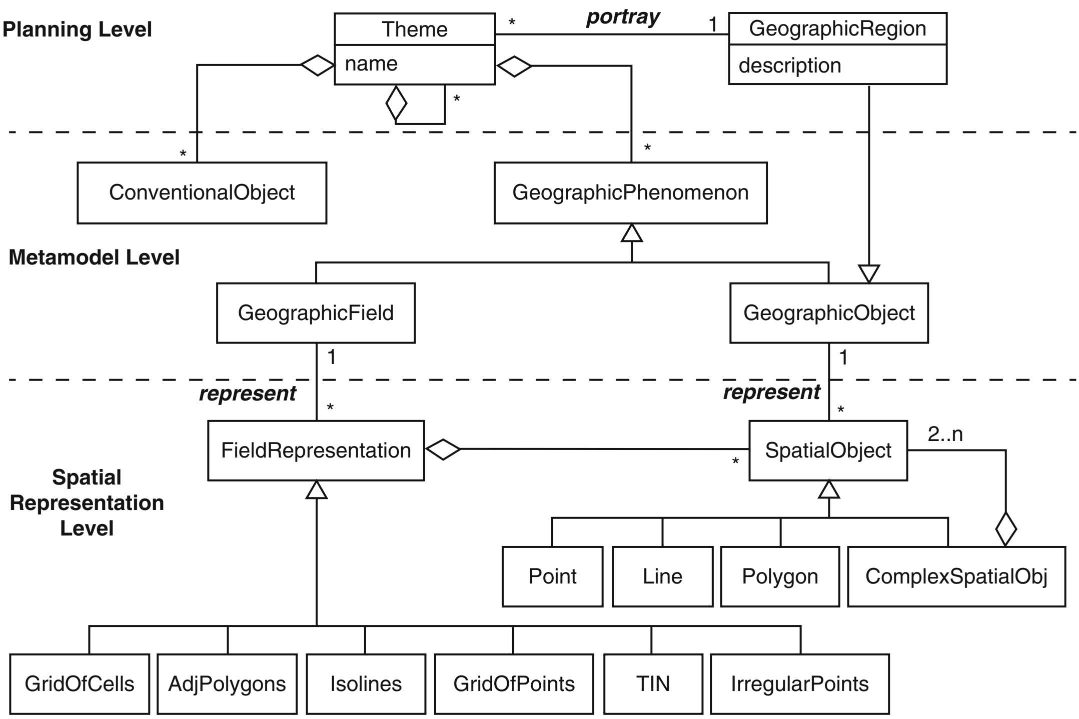
A conceptual data modeling language provides a formal base (notational and semantics) for tools and techniques used in data modeling. Data modeling is the abstraction process where only the essential elements of the observed reality are emphasized, the nonessential elements being discarded. The process of conceptual database modeling comprises the description of the possible data content, besides structures and constraints applicable to them. This database description is based on the semantic constructs provided by a conceptual data modeling language.

The UML-GeoFrame, originally presented in Lisboa Filho and Iochpe (1999), is based on a hierarchical class structure that makes up the conceptual GeoFrame framework. The GeoFrame provides the fundamental elements present in any GeoDB, whereas the UML class diagram provides the semantic constructs for a conceptual modeling language. This integration enables GeoDB design in a graphic language easily understandable by the users.

The result of the modeling process is a conceptual data schema that expresses "what" will be stored in the database and not "how" the data will be stored. A conceptual data schema becomes therefore an abstraction of the real world that is being modeled (miniworld). Consequently, every element of the reality to be modeled in the conceptual data schema must be stored in the GeoDB. In the same way, every object stored in a GeoDB must have been represented in the conceptual data schema, but this does not often happen.

A GeoDB stores three large data categories: conventional data without geographic reference (e.g., a property owner), geographic phenomena perceived in object view (e.g., cities, roads, parcels), and geographic phenomena perceived in field view (e.g., temperature, soil type, relief). The main UML-GeoFrame's contribution consists of providing a construct group that enables the designer to carry out the modeling of the geographic phenomena perceived in field view suitably. The geographic phenomena perceived in the object view and also the conventional objects are modeled in the same way as most existent modeling languages.

Therefore, the UML-GeoFrame uses the same constructs of the UML class diagram, such as classes and subclasses containing attributes and operations, and associations between classes, also enabling the specification of aggregations and compositions (Booch et al. 1998).



Modeling with a UML Profile, Fig. 1 The GeoFrame framework. *TIN* Triangular irregular network

The GeoFrame Framework

GeoFrame (Fig. 1) is a conceptual framework that provides a basic class diagram to assist the designer on the first steps of the conceptual data modeling of a new GIS application. The mutual use of a UML class diagram and GeoFrame allows the solution of most requirements of GIS application modeling.

The GeoFrame class diagram has three abstraction levels. The first is the planning level, which comprises the GeographicRegion class, whose instances correspond to the application interest areas, and the Theme class, describing the several themes that will portray this area. The metamodel level comprises the most generic classes of the geographic reality, which are divided in two categories, the conventional objects (without spatial representation) and the geographic phenomena, that comprise the geographic phenomena perceived in field view and the geographic phenomena perceived in object view. The third level includes the classes of objects that enable the designer to abstract the type of spatial representation that will be specified for each type of geographic phenomenon, multiple representations being possible.

To exemplify both the UML-GeoFrame's constructs and the respective GeoDB design methodology, a hypothetical support system for Brazilian agrarian reform will be used, described as follows.

The Brazilian government is initiating a process of land distribution for families of rural workers, in which nonproductive large landholdings are dispossessed to be divided and distributed. Each family of rural worker receives a parcel, size varying according to the country region and also depending on the available resources in the region, such as existent cropped areas, pastures, local roads, storage places, housing, or even natural resources availability such as water sources, streams, native vegetation, etc. A GIS application is used to assist in the demarcation of new parcels to be distributed. This distribution is carried out based on criteria that take into consideration, besides the resources previously mentioned, the relief, soil and vegetation type. Finally, effective environmental laws must be considered, as state laws prohibit the cultivation of agricultural crops in permanent protection areas (hill tops), areas with slope above 45° or close to water resources (lakes and rivers).

A short description of the classes belonging to each of the three GeoFrame abstraction levels follows. The way these classes are used during the modeling process is described in UML-GeoFrame Methodology section.

Planning Level

This comprises:

- GeographicRegion: this defines the geographic regions corresponding to the interest areas of a GIS application. For instance, the region corresponding to a large property that will be divided and the region of the municipal district in which the property is located.
- Theme: each geographic region can be portrayed by several themes. Two examples of themes in the context of the agrarian reform system are Allotment and Environmental Aspects. The themes can be organized in a theme and subtheme hierarchy. Hence, the Environmental Aspects theme could contain, for example, the subthemes Relief, Vegetation and Hydrography.

Metamodel Level

This comprises:

- ConventionalObject: this generalizes the classes of the application without spatial representation. An example could be the worker family class that will receive a parcel from the agrarian reform project.
- GeographicPhenomenon: this generalizes all application classes that define the geographic phenomena. This class is specialized in the GeographicField and GeographicObject classes.
- GeographicField: this class is a generalization of all application classes that are perceived in field view. These phenomena are also known as attributes varying in field, for example the classes Relief, Vegetation, Soil Use and Temperature.
- GeographicObject: this generalizes the application classes perceived in object view, in other words, classes whose instances have a single identity. Examples include Municipal districts, Farms, Parcels, Rivers and roads.

Spatial Representation Level

The spatial representation level in the GeoFrame, as part of the conceptual data modeling process that prioritizes "what" rather than "how", allows designers and users to specify the type of spatial representation used to abstract the spatial characteristics of each geographic phenomenon.

The purpose of GeoFrame is not to specify a type of data structure needed to store the spatial datum into the SDBMS, but only to model the spatial component of a particular geographic phenomenon as the user perceives or abstracts. For instance, in an urban water network application, the Hydrant class is associated with a spatial object of Point type (subclass of SpatialObject). This association only informs that the spatial characteristic of the geographic phenomenon hydrant will be zero-dimensional. However, as it corresponds to a vertex in a network structure, besides the *X* and *Y* coordinates, in SDBMS the vertex must be interlinked to other network elements (in order to maintain the topology) through a data structure of type arc-node. Nevertheless, this specification should only be detailed in the logical design phase of GeoDB. Following this approach, the GeoFrame classes in the spatial representation level are:

- SpatialObject: generalizes the classes of spatial representation of geographic phenomena in object view, such as Point, Line, Polygon or ComplexSpatialObject, which recurrently consists of two or more spatial objects. This last type of spatial representation is used when the geographic phenomenon presents a composed or complex characteristic (example: an archipelago).
- FieldRepresentation: the conceptual modeling of spatial representation of geographic phenomena in field view is the major

difference of the UML-GeoFrame compared with others. The FieldRepresentation class generalizes the main types of spatial representation used to abstract the spatial characteristic of phenomena in field view, which are: GridOfCells, AdjPolygons, Isolines, GridOfPoints, triangular irregular network (TIN) and IrregularPoints. This specification only deals with the way the designer/user frequently abstracts the spatial form of geographic fields. For example, many users imagine the relief as a geographic phenomenon usually represented by isolines, although other users work with the relief represented by a TIN or a digital elevation model. This basic group of six spatial representation models for phenomena in field view was identified and described in Goodchild (2002) as containing the models most commonly found in GIS. However, new models of spatial representation for fields can be added to GeoFrame.

A project methodology for GeoDB is presented next. GeoFrame consists of a class library that provides the fundamental elements present in any GeoDB. Nevertheless, the basic classes of GeoFrame are represented in the conceptual data schema only in an implicit way, which is done through stereotypes, as shown below.

The UML-GeoFrame Methodology

A methodology for GeoDB modeling based on the UML-GeoFrame is described here; in other words, the steps that should be followed during the GeoDB modeling process and how the GeoFrame elements are integrated with the UML class diagram constructs are presented.

The modeling process based on the UML-GeoFrame comprises five steps:

- Step 1: to identify themes and subthemes for each target region of the application
- Step 2: to draw a class diagram for each theme specified in step 1, associating classes of different themes, if this is the case
- Step 3: to model the spatial characteristic of each geographic phenomenon
- Step 4: to model spatial relationships
- Step 5: to model temporal aspects

The following subsections present each step in detail.

Step 1: To Identify Themes and Subthemes for Each Geographic Region

GIS applications are usually developed focusing on a particular geographic region, it can be an area of great extension such as a country, a state, a municipal district or a large river basin. Maps at small scales are usually manipulated in these applications. On the other hand, numerous applications focus on smaller areas such as a city, neighborhood, farm or small river basins, in which the degree of granularity of the manipulated data is usually much higher, with data usually represented in large-scale maps.

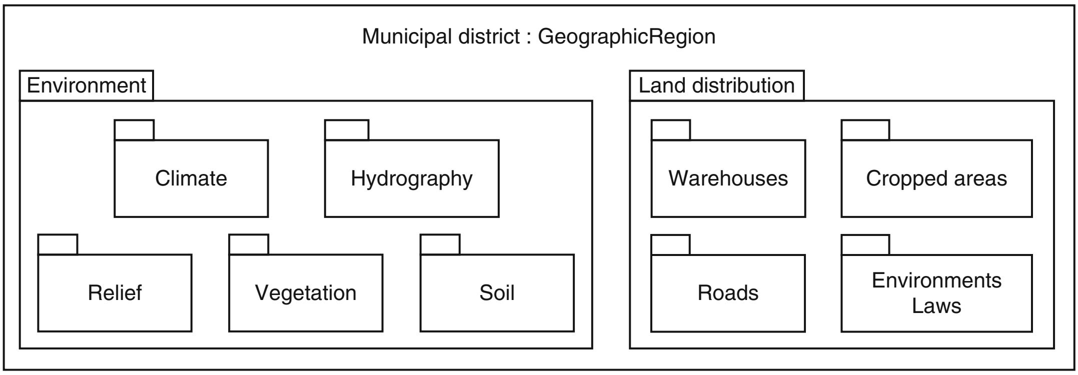
The specification of various themes that will portray each geographic region, besides allowing a top-down approach to the problem, also aims at facilitating the understanding of large data schema.

The elements identified at this stage are not directly transformed into a SDBMS structure; on the contrary, they are used only in the conceptual modeling phase, enabling the designer to plan and administer the data schema complexity. Theme modeling is done using the UML package construct. Figure 2 shows a possible hierarchical theme diagram for the agrarian reform support system.

Once the diverse themes are defined, the designer will be able to focus on a specific theme at a time to carry out data modeling (described in Step 2). For instance, one can chose Hydrograph and model all the classes of objects pertaining only to this theme. This process simplifies modeling and facilitates the understanding of the problem domain by the designer, as well as the communication with users.

Step 2: To Draw a Class Diagram for Each Theme

At this stage, modeling of the data is carried out. For each theme, the several elements of the real world that is being modeled are abstracted. This stage is similar to the traditional database modeling, in which the essential elements of the reality are classified and modeled with the UML



Modeling with a UML Profile, Fig. 2 Example of hierarchical theme diagram

class diagram constructs. At this stage, based on the GeoFrame, three classes of objects are identified: conventional, geographic phenomena perceived in object view and phenomena in field view. The existent associations are also modeled among the classes, however, without considering the spatial constraints, which will be discussed in Step 4.

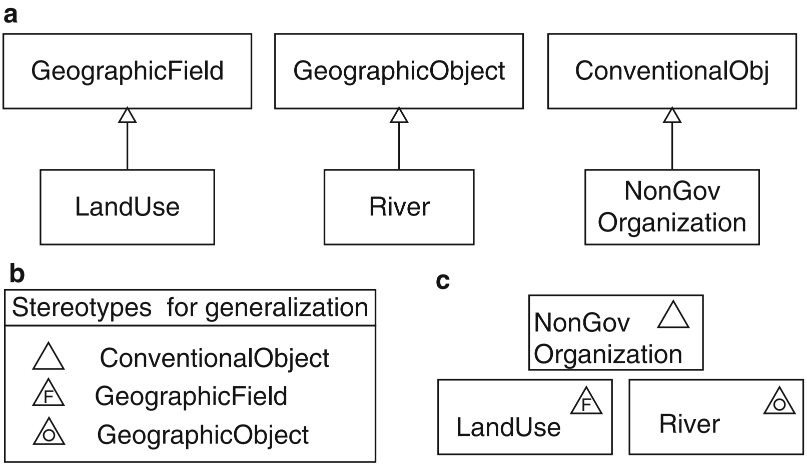
In a modeling process based on a class hierarchy, such as GeoFrame, the application classes should be modeled as subclasses of the GeoFrame classes (Fig. 3a). However, if all the application classes are represented as subclasses of only one of the three classes of the GeoFrame metamodel level, the data schema will be completely overloaded and difficult to read. Thus, three stereotypes were defined to replace these generalization-specialization relationships with graphic symbols (Fig. 3b), resulting in the representation shown in Fig. 3c. The idea of replacing relationships with graphic symbols was originally proposed by Bédard and Paquette (1989). The advantage of using these three stereotypes is that in large data schemas, the designers and users can easily identify the application classes according to their main category.

Step 3: To Model the Spatial Characteristic of Each Geographic Phenomenon

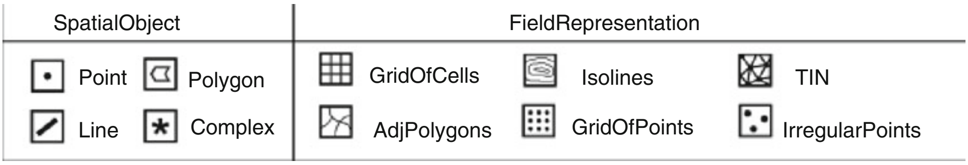
Geographic phenomena differ from conventional objects in that they have spatial properties (attributes and relationships), which are represented in a SDBMS by data structures containing primitive geometric object instances (e.g., point, line) or complex ones (e.g., multipoint, multipolygon), whose coordinates represent points in the geographic space based on a cartographic projection system (e.g., UTM).

During the conceptual modeling stage, the designer should not be stuck on details such "how" the geographic phenomena will be stored in SDBMS, but only on abstracting its spatial characteristics. For example, a pole in an electric network has the spatial characteristic of a point, while a lake has the characteristic of extension in the form of a region/polygon. In turn, the relief, which is perceived as a geographic phenomenon in field view, can have its spatial characteristic abstracted by a spatial model of the isoline type. Moreover, the spatial characteristic of a geographic phenomenon can be abstracted from different cartographic representation models, characterizing the existence of multiple representations for the same geographic phenomenon.

At the beginning of the 1990s, based on the object-oriented paradigm, a number of modeling languages proposed that geographic phenomena were modeled as specialized subclasses of a set of predefined classes representing geometric objects. Therefore, the class Street would be modeled as a subclass of the class Line, a class Pole should be modeled as a subclass of the class Point, etc. There was in this approach an incorrect use of the generalization-specialization concept, in which two distinct things were related by an



Modeling with a UML Profile, Fig. 3 Generalization-specialization stereotypes



Modeling with a UML Profile, Fig. 4 Stereotypes for spatial representation

IS-A relationship. That is, a road is not a line and a line is not a road, although a geometric object Line can be related to a geographic object Road to represent its spatial location.

With the UML-GeoFrame, the spatial characteristic of geographic phenomena is not abstracted in the form of spatial attributes, but by means of associations between the classes of geographic phenomena and the classes of spatial representation. This is specified by the *represent* association in the GeoFrame (Fig. 1). Again, in order not to overload the data schema, stereotypes are defined to replace these associations (Fig. 4).

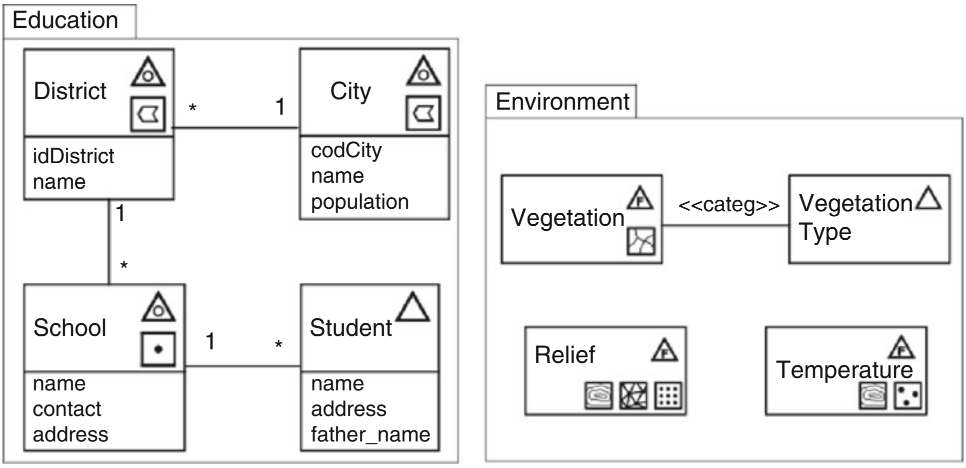
Therefore, in an UML-GeoFrame data schema, each geographic phenomenon class of the application domain will have at least two stereotypes, one for specialization and another for spatial representation. Figure 5 shows an example of a UML-GeoFrame data schema with two packages, one related to the Education theme and the other related to the Environment theme.

As can be seen in this example, the UML-GeoFrame enables a natural, integrated and consistent form of modeling phenomena perceived in field view. In this way, the conceptual data schema contains all the elements of the real world to be stored in the SDBMS and vice versa.

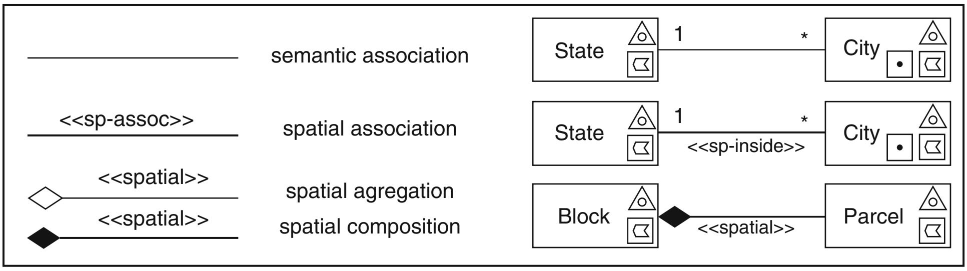
Step 4: To Model Spatial Relationships

A relationship modeled on a conceptual data schema semantically implies two facts; the first is that objects of the respective classes can be associated, which allows the user to consult the database based on this relationship, the second is that DBMS will have to guarantee the integrity constraint specified by the multiplicity of this relationship automatically. For example, an one-to-many association (1..* in UML) between the classes City and Farm, means that each farm must be associated at the most to one city, and that each city can be associated to zero or many farms.

In a GIS, this type of relationship can be captured in two ways: spatial and semantic. The spatial way is obtained by spatial operations between two geometric objects, for instance, verifying



Modeling with a UML Profile, Fig. 5 Example of a data schema using the UML-GeoFrame notation



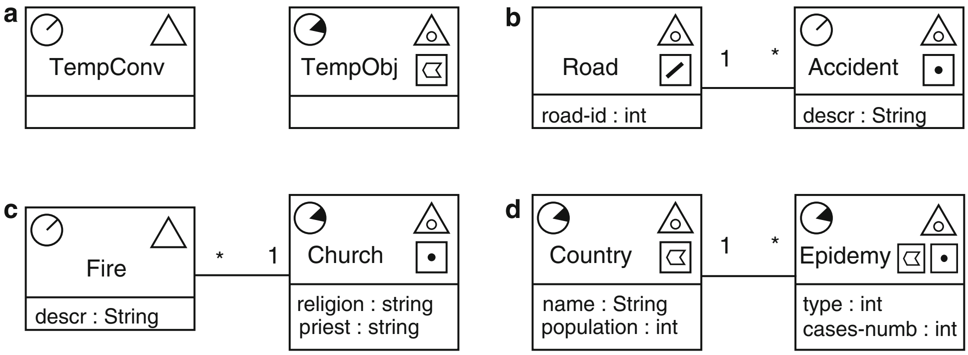
Modeling with a UML Profile, Fig. 6 Stereotypes for spatial relationship

whether a polygon is inside another polygon. The semantic way is the traditional database way, where the farm related register contains a field that stores a reference (e.g., foreign-key) for the city that the farm is associated with.

With the UML-GeoFrame the designer can specify the two types of relationships (spatial and semantic), meaning that the SDBMS must guarantee the specified integrity constraint. The semantic relationships correspond to the associations normally specified between classes in the data schema. In this case, referential integrity constraints can be transformed, for example, to foreign-key constraints in a relational DBMS. The integrity constraints of spatial relationships must be implemented in GIS or SDBMS specific procedures.

The possible types of spatial relationships are topological or metric. Engenhofer et al. (1994) described a set of types of spatial intersections that can occur between two regions: disjoint, contain, inside, equal, touch, covers, covered by and overlap. This set can be applied to other types of combinations such as point and polygon or line and polygon.

The specification of spatial associations in the UML-GeoFrame is done with textual stereotypes, i.e., a text between "…". The text corresponds to the type of spatial constraint one wants to impose, which can be of any type, including the relationships shown in Fig. 6. If there is no textual stereotype, then the association is semantic. Thus, in the logical design phase, the semantic association between State and City (Fig. 6) will be implemented as an attribute in the table City with foreign-key constraint for the table State, whereas the spatial association between State and City will be implemented by a procedure that will apply an *inside* spatial operation. These examples also show that an instance of City can



Modeling with a UML Profile, Fig. 7 **(a)–(d)** Examples of spatiotemporal classes

have multiple spatial representations (point or polygon). Finally, UML constructs of aggregation and composition correspond to WHOLE-PART relationships. Figure 6 shows a spatial composition where a Block is modeled as the whole and the Parcels correspond to the parts.

Step 5: To Model Temporal Aspects

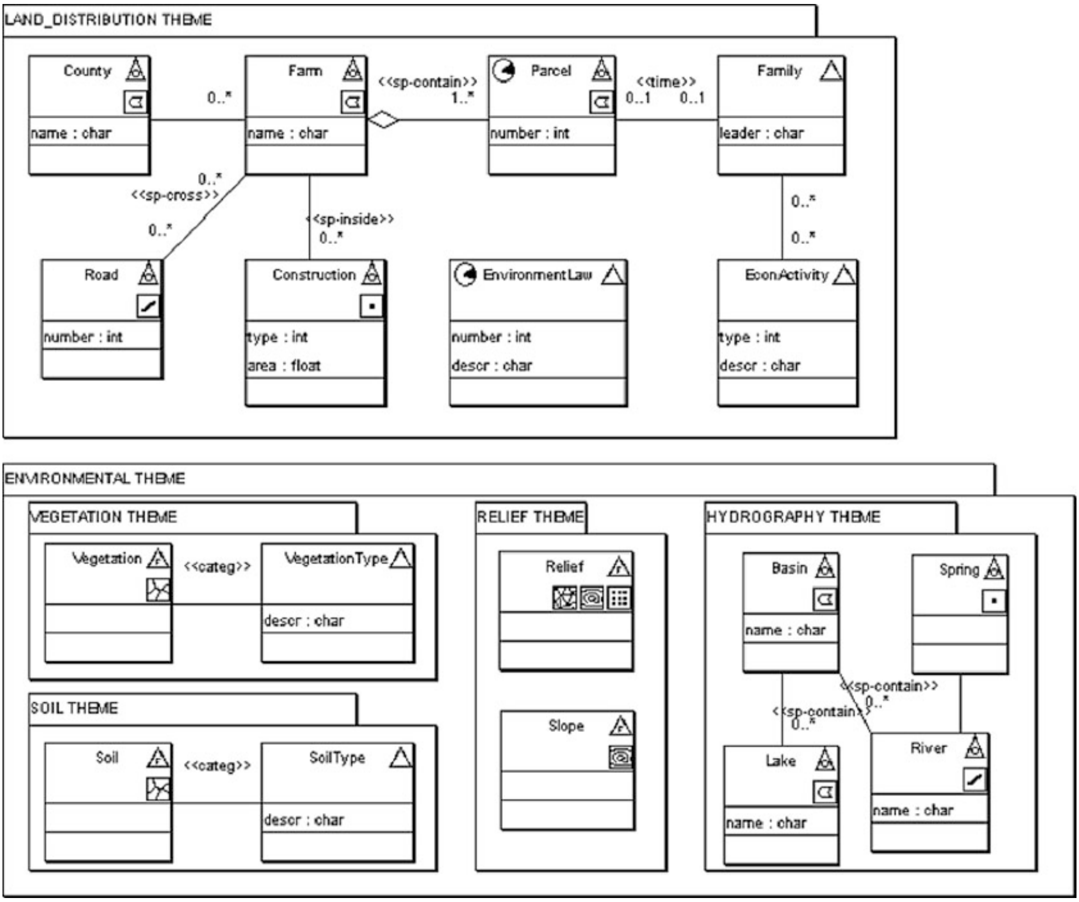
Numerous geographic phenomena are dynamic, i.e., their properties (descriptive and spatial) undergo changes with time. Although most databases applications reflect only the current state of real-world objects they represent, many applications need to keep the data evolution description as they are changed.

Descriptive, temporal and spatial dimensions are orthogonal. Temporal properties can therefore be defined for the three geospace data categories: conventional objects, geographic phenomena perceived in field view and geographic phenomena perceived in object view. In temporal DBMS, two types of time can be defined: valid time and transaction time. Rocha et al. (2001) presents an extension to the UML-GeoFrame comprising these types of time. However, the enormous possibility of combinations between the different dimensions (e.g., temporal and spatial) led to a highly complex model that was really very little used by GIS users/designers. Thus, only the valid time is presented here.

Valid time is the time instant or time interval when an object of the real world is considered valid. For example, the strike against the World Trade Center took place on September 11, 2001; in turn, the Gulf War occurred in the period between August 1990 and February 1991. Hence, another important factor is the granularity of temporal information. UML-GeoFrame considers three types of time granularity: Date, Time and Timestamp. Specifying the granularity of a temporal attribute is the same as defining the domain of a descriptive attribute value (e.g., CHAR or Boolean).

Besides granularity, the designer can specify two types of temporal occurrence: *Interval* and *Instant*, which is done using the stereotypes shown in Fig. 7a. An *Interval* temporal class indicates the need for storage of its evolution, that is, if its properties (descriptive or spatial) are changed, a new version of the same object will be created. In this case, temporal attributes will be inserted to each object version, indicating the initial and final valid time of this object version. In the case of *Instant* temporality, the class will contain temporal attributes, but new versions will not be created for its objects. The UML-GeoFrame also allows the specification of temporal relationships. Following, some examples of modeling of temporal aspects are presented.

Figure 7b shows an association between the Road and Accident classes. The Road class has only spatial aspects, whereas the Accident class is spatiotemporal of the *Instant* type, characterizing the need for storing the instant (e.g., timestamp) when the accident took place. In the case of *Instant* temporality, each occurrence (a new accident) creates a new object, and therefore temporal versions of the object will not be generated.



Modeling with a UML Profile, Fig. 8 Final class diagram for the hypothetical system of support to agrarian reform

Figure 7c illustrates the Fire class, whose instances are conventional objects with *Instant* temporality, associated with a spatiotemporal class Church. Each fire occurrence generates a new object instance that contains temporal data recording fire valid time (e.g., date or year). In turn, each change in Church properties, with *Interval* temporality, creates a new version of the same object.

Figure 7d shows two spatiotemporal classes of the *Interval* type. Each time that a property of the Country class (e.g., population) is changed, a new version of the object country will be created. The same occurs with the class Epidemic, in which both the number of cases and the affected geographic region can be changed.

The UML-GeoFrame methodology establishes the five steps described, but they do not need to be executed necessarily in this order. A more experienced designer can model the three aspects (descriptive, spatial and temporal) of a class at the same time. Nevertheless, considering in detail all the aspects involving a GeoDB design is not a trivial task. The objective of the methodology is to assist the designer to work methodologically, in an incremental and organized way.

An Example of the Application for Brazilian Agrarian Reform

Figure 8 shows the final UML-GeoFrame diagram, relative to the hypothetical system of support to agrarian reform, described at the beginning of this entry. This schema was depicted using the ArgoCASEGEO CASE tool (Lisboa Filho et al. 2004).

In Fig. 8, how the division of the data schema using packages describing the diverse themes facilitates the understanding of the project is shown. This example also shows how natural is modeling, in an integrated way, conventional objects and geographic phenomena in field and object views. Multiple spatial representations, spatial relationships and temporal aspects are also easily specified using the UML-GeoFrame.

Key Applications

The development of a GIS application, however simple it might be, will bring great benefits if a conceptual data modeling language is used during the system design phase. The use of these conceptual data modeling languages in medium- and large-size systems is fundamental. GIS applications have become more and more integrated with other corporative information systems, sharing diverse data bases and running not only as operational control systems (e.g.: cadastres of properties and infrastructure service networks), but as important decision-making support systems.

Future Directions

A CASE tool named ArgoCASEGEO has been developed (Lisboa Filho et al. 2004) to assist GeoDB modeling using UML-GeoFrame. This tool generates logical-spatial data schemas for the main data models related to GIS software (e.g., Shape File, Oracle Spatial). One of the innovative characteristics of this tool is to support a catalogue of analysis patterns (Lisboa Filho et al. 2002) aimed at enabling the reuse of GeoDB design solutions by different designers.

Acknowledgements This work has been partially supported by CNPq (Brazilian National Research Council) and Fapemig.

Cross-References

▶ Modeling with Pictogrammic Languages

References

Abiteboul S, Hull R (1987) IFO: a formal semantic database model. ACM Trans Database Syst 12:525–565

Bédard Y, Paquette F (1989) Extending entity/relationship formalism for spatial information systems. In: AUTO-CARTO 9, ninth international symposium on computer assisted cartography, ASPRS-ACSM, Baltimore, 2–7 Apr 1989

Bédard Y, Larrivée S, Proulx MJ, Nadeau M (2004) Modeling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on perceptory. In: Wang S et al (eds) Conceptual modeling for GIS (COMOGIS) workshop ER2004, Shanghai, 8–12 Nov 2004. Lecture notes in computer science, vol 3289. Springer, Berlin, pp 17–30

Booch G, Jacobson I, Rumbaugh J (1998) The unified modeling language user guide. AddisonWesley, Reading

Borges KA, Davis CD, Laender AHF (2001) OMT-G: an object-oriented data model for geographic applications. GeoInformatica 5:221–260

Chen PPS (1976) The entityrelationship model: towards a unified view of data. ACM Trans Database Syst 1:9–36

Coad P, Yourdon E (1991) Object-oriented analysis, 2nd edn. Prentice-Hall, New York

Couclelis H (1992) People manipulate objects (but cultivate fields): beyond the rastervector debate in GIS. In: Theories and methods of spatialtemporal reasoning in geographic space. Lecture notes in computer science, vol 639. Springer, Berlin, pp 65–77

Elmasri R, Navathe SB (2000) Fundamentals of database systems, 3rd edn. AddisonWesley, Reading

Engenhofer M, Clementini E, Felice P (1994) Topological relations between regions with holes. Int J Geogr Inf Syst 8:129–144

Goodchild MF (2002) Geographical data modeling. Comput Geosci 18:401–408

Lisboa Filho J, Iochpe C (1999) Specifying analysis patterns for geographic databases on the basis of a conceptual framework. In: ACM symposium on advances in geographic information systems, Kansas City, 5–6 Nov 1999, pp 9–13

Lisboa Filho J, Iochpe C, Borges KA (2002) Analysis patterns for GIS data schema reuse on urban management applications. CLEI Electron J 5:1–15

Lisboa Filho J, Sodré VF, Daltio J, Rodrigues MF, Vilela V (2004) A CASE tool for geographic database design supporting analysis patterns. In: Wang S et al (eds) Conceptual modeling for GIS (COMOGIS) workshop ER2004, Shanghai, 8–12 Nov 2004. Lecture notes in computer science, vol 3289. Springer, Berlin, pp 43–54

OMG – Object Management Group (2007) Unified modeling language. Available at http://www.uml.org

Rocha LV, Edelweiss N, Iochpe C (2001) GeoFrame-T: a temporal conceptual framework for data modeling. In:

ACM symposium on advances in geographic information systems, Atlanta, 9–10 Nov 2001, pp 124–129

Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W (1991) Object-oriented modeling and design. Prentice-Hall, Englewood Cliffs

Shekhar S, Chawla S (2003) Spatial databases: a tour. Prentice Hall, New York

Modeling with Enriched Model-Driven Architecture

André Miralles[1] and Thérèse Libourel[2]
[1]Centre for Agricultural and Environmental Engineering Research, Earth Observation and GeoInformation for Environment and Land Development Unit, Montpellier Cedex 5, France
[2]University of Montpellier II – National Center for Scientific Research, Montpellier Laboratory of Computer Science, Robotics, and Microelectronics, Montpellier Cedex 5, France

Synonyms

MDA; MDE; Model Driven Architecture; Model Driven Development (MDD); Model Driven Engineering

Definition

Model Driven Architecture, formalized in 2001, is a software design approach proposed by the Object Management Group (OMG) with the objective of improving application development. It was conceived of in order to improve the productivity of software development but also to resolve problems of software portability, software integration and software interoperability encountered during development (Kleppe et al. 2003).

To achieve this objective, the MDA approach recommends separating the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform. For that, the authors of this approach suggest the use of two types of model groups: the Platform Independent Models (PIM) and the Platform Specific Models (PSM).

PIMs are models providing a description of the structure and functions of a system without technical specifications of data-processing nature. PSMs are models defining how structure and functions of a system are implemented on a specific platform.

In fact, the MDA approach introduces a separation between concepts and specifications needed to develop software. PIMs only contain business concepts. PSMs contain business concepts and also implementation concepts. Since all of the PIM business concepts are included in PSMs, a PIM can be seen as a modified subset of a PSM. Therefore, a PSM always derives from a model PIM through one or more transformations. Figure 1 illustrates this separation and transformation. If different platforms are used for the implementations (e.g., same standardized model implemented into different organizations), then more than one PSM may be derived from the same PIM.

The previous transformations, called PIM/PSM transformations, are not the only ones. In fact, the authors of MDA mention, on the one hand, the existence of PSM/PIM transformations converting a PSM into PIM and, on the other hand, transformations whose models source and target are in the same fashion standard: PIM/PIM transformations or PSM/PSM transformations.

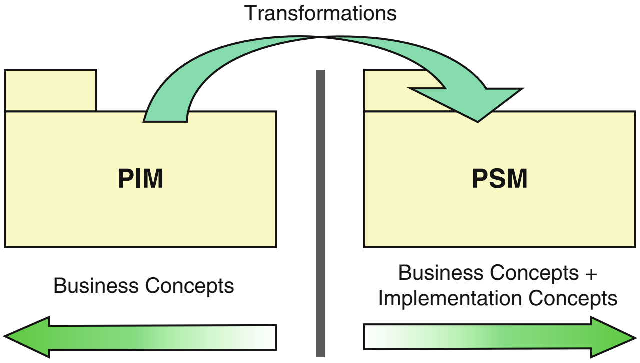
In the process of development, the PSM is not the last step since it is then necessary to project this model into a programming language. This projection is often considered to be a transformation.

In summary, the separation introduced by MDA can be seen as a solution of a more fundamental preoccupation: the capitalization of knowledge.

Historical Background

The Model Driven Architecture is an approach which expands the Object Management Architecture (Miller and Mukerji 2001) without replacing it. The Object Management Architecture provides a framework to implement distributed systems (Miller and Mukerji 2003) whereas the

Modeling with Enriched Model-Driven Architecture, Fig. 1 Concepts separations and transformations



Model Driven Architecture defines an approach specifying how to use the models during the development of an application.

The founding text of Model Driven Architecture is a recent OMG work that was approved in 2001 (Miller and Mukerji 2001). In 2003, this text was supplemented by a methodological guideline defining the main directives to apply this approach.

In 2003, Anneke Kleppe et al. wrote in their book entitled, *MDA Explained: The Model Driven Architecture-Practice and Promise* (Kleppe et al. 2003), that this approach was "still in its infancy". The following year, Anneke Kleppe, when interviewed about Model Driven Architecture (Kleppe 2004), was quoted as saying, "Can we realistically look to build all of the J2EE code for an application from UML diagrams?" and "Yes, it may take five, ten, or maybe twenty years before it will be common practice, but I believe it will become just that."

These two statements explain that a lot of research has to be carried out so that this approach reaches its maturity. For over 20 years, R&D and commercial software engineering software packages have offered semi-automated transformations between so-called "conceptual" and "physical" models, with more or less success with regards to their widespread usage. Nowadays, with the standardization impacts of MDA and UML, hope is returning. Currently, research works are mainly concentrated on the transformations of models to automate these transformations in order:

- Firstly, to carry out a *Full* MDA process (Kleppe 2004), i.e., a process automating the evolution of the models from the analysis to the implementation
- Secondly, to increase the productivity during the development and facilitate code reuse.

Scientific Fundamentals

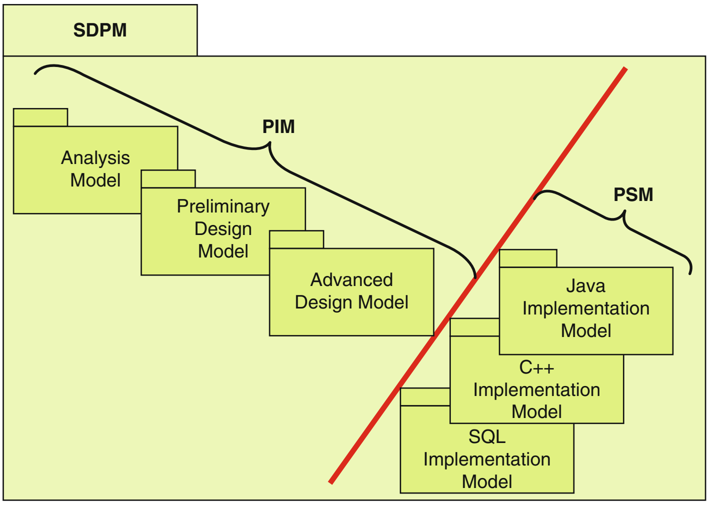
The Enriched Model Driven Architecture is a framework for designing and implementing spatio-temporal databases following a MDA approach into a Full MDA process. This process includes the generation of the SQL code necessary to implement the spatio-temporal database. For that, a multimodel artifact and a panoply of transformations has been conceived and implemented.

In order to describe the spatial and temporal properties of the business concepts, the pictogrammic language of the Perceptory case tool (Bédard et al. 2004) has also been adopted and implemented.

Principle of the Software Development Process Model (SDPM)

When an application is developed, one of the main preoccupations for a project manager and for the company in charge of the software development is the capitalization and reuse of the knowledge accumulated during the development.

The capitalization of knowledge is not just the problem of separating the business concepts



Modeling with Enriched Model-Driven Architecture, Fig. 2 Principle of the multimodel artifact Software Development Process Model

and implementation concepts according to the MDA vision. *The capitalization of knowledge is problematic throughout the development of an application*. The *Software Development Process Approach* (SDPA) is a framework based on this report (Miralles 2006). As with each phase of the development, the type of mobilized knowledge is different. The idea of this new approach is to capitalize the knowledge of each one of the phases. For that, this new approach recommends dedicating a model at each one of the phases.

In order to materialize this new approach, a multimodel artifact, called the *Software Development Process Model* (SDPM), has been conceived (Miralles 2006). This multimodel artifact contains the different models corresponding to the phases of the software development process. In this vision, the Software Development Process Model is the MODEL of the application under development. Figure 2 shows the Software Development Process Model for the development of software following the Two Track Unified Process method (Roques and Vallée 2002) derived from the Unified Process method. This figure also shows that the PIM/PSM separation introduced by MDA occurs when the project moves from the advanced design phase to the implementation phase.

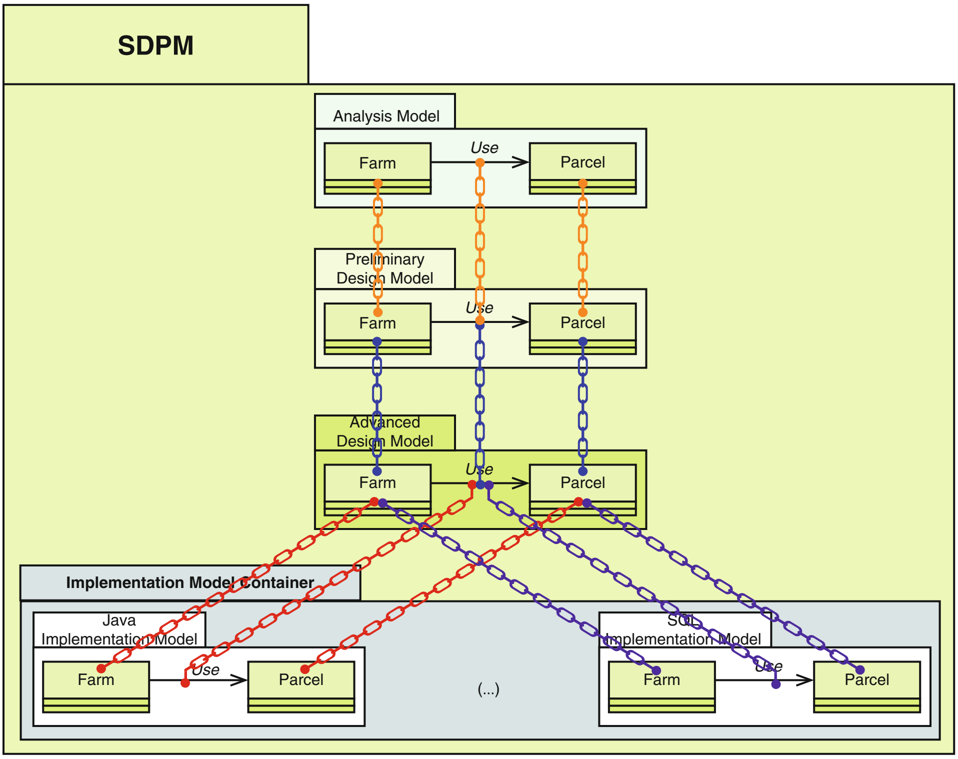
Panoply of Transformations for a Full MDA Process

Diffusion Transformation and Management of the Software Development Process Model

In order to manage the different models, the Software Development Process Model has been endowed with a Diffusion transformation (Miralles 2006). This transformation first clones a concept from a source model into the following model. Step by step, the concepts that are captured in the analysis phase and added into the analysis model are transferred into the implementation models.

In order to guarantee the coherency of the multimodel artifact, a Cloning Traceability Architecture is automatically built by the Diffusion transformation. After cloning, this transformation establishes an individual cloning traceability link between each one of the source concepts and the cloned concepts. Figure 3 illustrates the Cloning Traceability Architecture.

In an iterative development process, the Diffusion transformation adds, at every iteration, a new clone of the same source into the following model. To avoid this problem, when an individual cloning traceability link exists, the Diffusion transformation does not clone the concepts, but carries out only one update of the clone.



Modeling with Enriched Model-Driven Architecture, Fig. 3 Example of Cloning Traceability Architecture (to keep the figure simple, differing details of the models have been discarded)

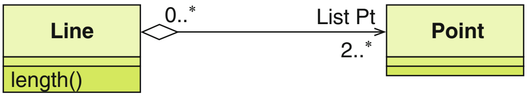
Full MDA Process for GIS Design and Development

After a thorough analysis, the pictogrammic language of Perceptory (Bédard et al. 2004) has been adopted to describe the spatial properties (Point, Line, and Polygon) and temporal properties (Instant and Period) of the business concepts. This pictogrammic language is applied on the analysis model of the SDPM. These pictograms are added to the business concept (i.e. classes) via stereotypes, one of extensibility mechanisms in UML. This mechanism allows to associate an icon or a pictogram on business concepts.

In the analysis model, the stereotype/pictogram couple only has an informative value. To be able to generate an application's code, these pictograms have to be reified into UML modeling elements. To do this, two geomatic transformations have been designed and implemented: the first generates a design pattern (A design pattern systematically names, motivates, and explains a general design that addresses a recurring design problem in object-oriented systems. It describes the problem, the solution, when to apply the solution, and its consequences. It also gives implementation hints and examples. The solution is a general arrangement of objects and classes that solve the problem. The solution is customized and implemented to solve the problem in a particular context (Gamma et al. 2001)) based on spatial concepts and temporal concepts (Instant and Period) and the second converts the stereotype/pictogram couple into a relationship.

GIS Design Pattern Generation Transformation

The spatial and temporal concepts have stable relationships that are entirely known. They constitute recurrent mini-models having the design



Modeling with Enriched Model-Driven Architecture, Fig. 4 Example of a GIS Design Pattern

pattern properties. Figure 4 shows an example of a design pattern of the Geographic Information domain. The set of these patterns are called GIS Design Patterns.

Given that the design patterns are always identical, they can be automatically generated without any difficulty. The GIS Design Pattern Generation transformation is the transformation in charge of generating the GIS Design Pattern.

Pictogram Translation Transformation

Once the GIS design patterns have been created, the business and the spatial or temporal concepts represented by the pictogram are totally disassociated (Fig. 5). The goal of the Pictogram Translation Transformation is to automatically establish a relationship between the Parcel and Polygon concepts (cf. Fig. 5). This transformation creates an association called Spatial Characteristic.

During the capture of the pictogram, two tagged values are added to the business concept in order to specify the role of the spatial concept ({Gis S: Spatial Role(Geometry)}) and its cardinality ({Gis S: Spatial Cardinality(1)}). By default, this role and this cardinality have the values Geometry and 1, respectively, but the designer can subsequently modify them. In this association, the entity name has been allocated to its role, Parcel in this example, and its cardinality's value is 0..1. Once the association has been created, the stereotype/pictogram and the two tagged values are deleted since this information becomes redundant.

To ensure traceability, the Pictogram Translation Transformation creates a traceability link, called Translation Traceability Link, between the pictogram of the business entity of the analysis model and the Spatial Characteristic association.

SQL Transformations

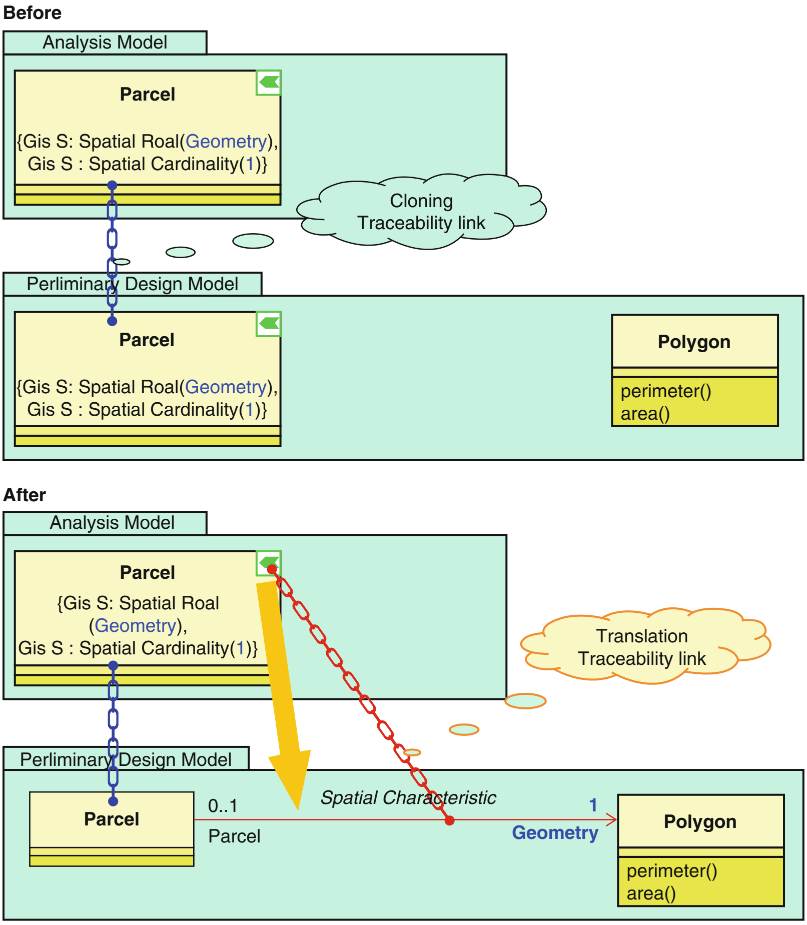
In order to achieve a Full MDA process, SQL Transformations have conceived and implemented. They are applied on the SQL Implementation model. The objective of these transformations is to adapt the SQL Implementation Model after cloning to the SQL code generator of the Case Tool.

The main SQL transformation adds the persistence and primary key attributes (main features of relational database) on UML concepts. Annotated with information specific to SQL, it is possible to generate SQL code to create the spatial database. For that, the SQL code generator available in Case Tool supports is used.

Conclusion

The Software Development Process Approach is a recent contribution to the design and development of spatial databases. It enriches the MDA by subdividing the PIM level into three levels based on the development process. The reification of this approach is the Software Development Process Model, a multimodel artifact that meets the previously stated requirements of capitalization. This is so because each of the four modeling levels include concepts that are their very own and independent of those of the preceding level. Without the cloning traceability architecture and the Diffusion Transformation, the functioning and the maintenance of coherence of the Software Development Process Model would become impossible.

Structuring models into SDPM offers a very powerful mutation capability; it becomes very easy to change the hardware or software platform in the course of the development and to improve performance calculations, for example. Such changes can be made at a minimum cost since the analysis, preliminary design and advanced design models constitute the application's **capital**, capital that is available and mobilizable at any time. Thus, only the new implementation concepts will require reworking within the new implementation model. This mutation capability can also be called upon during the application's



Modeling with Enriched Model-Driven Architecture, Fig. 5 Rule for transforming the stereotype/pictogram couple into an association

life cycle to correct bugs, improve features, mitigate obsolescence of hardware or software platforms, to create new versions of the application, etc.

The two geomatic transformations that have been designed use spatial, temporal and spatio-temporal properties of business concepts annotated with Perceptory pictogrammic language. Some SQL transformations adapt the SQL Implementation Model to the code generator of the Case Tool.

This panoply of transformations automates, for the first time in a full MDA process, the generation of SQL code that allows the Geographical Information System's database to be reified.

Key Applications

Software Development and Information Technology

The potential benefits in terms of productivity, portability, integration and interoperability confer a transversality that should be of interest to the software industry regarding the Software Development Process Approach; more so because this approach can be implemented irrespective of the typology of the software application to be created.

Geographic Information Systemsand Spatial Databases

The field of Geographic Information Systems, including the design of associated spatial or spatio-temporal databases, benefit from Enriched Model Driven Architecture.

Future Directions

MDA is an approach that is still young; its broad principles were only enunciated by OMG in 2001. It will surely evolve in the coming years, either at the generalization level – the present solution is an example – or at the level of transformation languages (QVT, ATL, etc.).

Acknowledgements The authors thank Professor Yvan Bédard of the Geomatics Research Center at Laval University and his team for stimulating our thought processes and for contributing to our research.

Cross-References

▸ Computer Environments for GIS and CAD
▸ Movement Patterns in Spatio-Temporal Data

References

Bédard Y, Larrivée S, Proulx MJ, Nadeau M (2004) Modeling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 Years of research and experimentations on perceptory. In: Proceedings of ER Workshops 2004 CoMoGIS, Shanghai

Gamma E, Helm R, Johnson R, Vlissides J (2001) Design patterns – elements of reusable object-oriented software, 1st edn. Addison Wesley Professional, London

Kleppe A (2004) Interview with Anneke Kleppe. Code generation network. http://www.codegeneration.net/tiki-read_article.php?articleId=21. Date cited: Aug 2006

Kleppe A, Warmer J, Bast W (2003) MDA explained: the model driven ArchitecturePractice and promise. AddisonWesley Professional, London

Miller J, Mukerji J (2003) MDA guide version 1.0.1. OMG. http://www.omg.org/cgi-bin/doc?omg/03-06-01

Miller J, Mukerji J (2001) Model driven architecture (MDA). OMG. http://www.omg.org/cgi-bin/apps/doc?ormsc/01-07-01.pdf. Date cited: Sept 2004

Miralles A (2006) Ingénierie des modèles pour les applications environnementales. Dissertation, Université Montpellier II, Montpellier

Roques P, Vallée F (2002) UML en Action – De l'analyse des besoins à la conception en Java, 2nd edn. Eyrolles, Paris

Modeling with ISO 191xx Standards

Jean Brodeur[1] and Thierry Badard[2]
[1]Center for Topographic Information, Natural Resources Canada, Sherbrooke, QC, Canada
[2]Department of Geomatic Science, Center for Research in Geomatics (CRG), Université Laval, Québec, QC, Canada

Synonyms

Conceptual modeling of geospatial databases; Modeling geospatial application database; Modeling geospatial databases

Definition

Application Schema

ISO19109-Rules for application schema (ISO/TC211 2005a) defines an *application schema* as a conceptual schema for data required by one or more applications. In the context of geographic information, an application schema documents the content and the structure of geographic databases along with manipulating and processing operations of the application to a level of details that allows developers to set up consistent, maintainable, and unambiguous geographic databases and related applications (Brodeur et al. 2000). As such, an application schema contributes to both the semantics of geographic data and describes the structure of the geographic information in a computer-readable form. It also supports the use of the geographic data appropriately (i.e., fitness for use). Typically, an application schema is depicted in a formal conceptual schema language.

Feature Catalog

As in ISO19110-Methodology for feature cataloguing (ISO/TC211 2005b), a *feature catalog* presents the abstraction of reality represented in one or more sets of geographic data as a defined classification of phenomena. In the context of geographic information, it is a collection of metadata that provides the semantics and the structure of the objects stored in a geographic database. A feature catalog includes (1) the names and definitions of feature types, (2) their properties' name and definition including feature attributes, geometry (shapes and specifications, datum, map projection, etc.), temporality (dimensions and specifications, datum, units, resolutions, etc.), operations, and roles, (3) descriptions of attribute values and domains, relationships, constraints, and so on. An application schema may be described in various forms such a text document, a database, a spreadsheet, etc. Typically, a feature catalog is available in electronic form to support interoperability of geographic information. Although both an application schema and a feature catalog address the same content basically, they are complementary in the manner they represent it.

Historical Background

International standardization in the geographic information field has taken place during the last decade. These works have been coordinated by the ISO Technical Committee 211 (ISO/TC211 2007) and the Open Geospatial Consortium (OGC 2007). Standardization in geographic information aims at (1) facilitating and increasing the understanding and use of geographic data, (2) increasing the availability, the access, the integration, the exchange, and the sharing of geographic data (i.e., interoperability of geographic information), (3) enhancing efficiency when using geographic data, and (4) developing a worldwide orientation of geographic information in support of global problems (ecological, humanitarian, sustainable development, etc.) (ISO/TC211 2007).

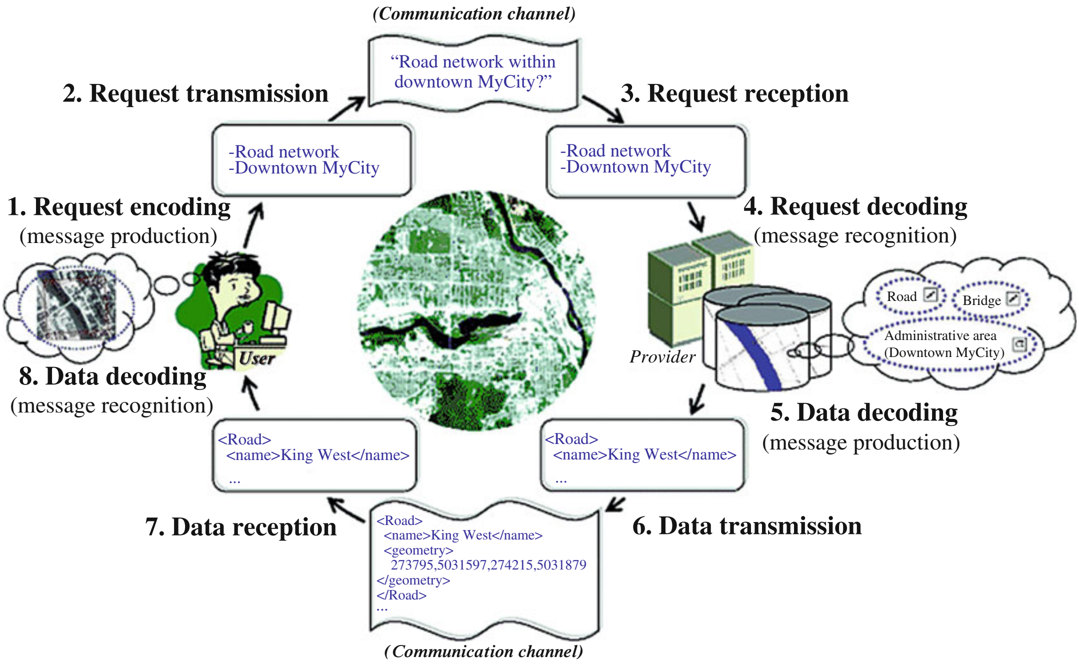
Scientific Fundamentals

Interoperability

As introduced in Institute of Electrical and Electronics Engineers (1990), interoperability can be seen as the ability of two or more systems or components to exchange information and to use the information that has been exchanged. As such, interoperability adheres to the human communication process (see Fig. 1) where agents (e.g., human beings, systems, etc.) interact together at the system, syntactic, schematic, and semantic levels to share information. Each agent has its own conceptual representation of reality and uses it to encode (Fig. 1, steps 1 and 5) and decode messages (e.g., queries and responses about geographic information, Fig. 1, steps 4 and 8), which are transmitted (Fig. 1, steps 2 and 6) to or received (Fig. 1, steps 3 and 7) from another agent through the communication channel. Interoperability happens only when both agents engaged in a communication have the same understanding about the message (Brodeur et al. 2003). Therefore, interoperability agrees to a bidirectional communication process including a feedback mechanism in both directions to control the good reception and understanding of messages.

Accordingly, topics of interest to standardization in geographic information are organization, content, access and technology, and education. Standards cover methods, tools, and data management services (including data definition and description) that are related to interoperability in geographic information including data acquisition, processing, analysis, access, portrayal, and transfer of data between users, systems, and places. They provide a structure for application development related to geographic data and refer to existing standards in information technology and communication (W3C (World Wide Web Consortium), OMG (Object Management Group), IETF (The Internet Engineering Task Force), OASIS (Organization for the Advancement of Structured Information Standards), and so on) when needed. A large number of standards are published as part of the ISO191xx suite of standards (see http://www.iso.ch/iso/en/stdsdevelopment/tc/tclist/TechnicalCommitteeDetailPage.TechnicalCommitteeDetail?COMMID=4637).

As ISO/TC 211 and OGC share common objectives, a similar program and have complementary approaches, they are now collaborating closely. On the one hand, the OGC wants their



```
<Road>
  <name>King West</name>
  ...
```

```
<Road>
  <name>King West</name>
  ...
```

```
<Road>
  <name>King West</name>
  <geometry>
   273795,5031597,274215,5031879
</geometry>
</Road>
...
```

Modeling with ISO 191xx Standards, Fig. 1 Framework for interoperability

specifications to obtain official recognition of international standards. On the other hand, ISO/TC 211 wants to benefit from OGC's work and that OGC's specifications comply with ISO 191xx standards. By sharing certain of their resources and developing standardization projects jointly, they aim at reducing the inconsistency between de jure (i.e., official, approved by an official standardization body like ISO) and de facto or industrial standards (i.e., adopted by users, industry and/or professional sector, because of its popularity and its market share). As an example, ISO 19115-Metadata has been adopted by OGC as its Topic 11-Metadata, and the specifications of the Web Map Service (WMS) edited by OGC are now an ISO international standard (ISO 19128).

As such, the first part of this chapter concerns the role of conceptual modeling in the field of geographic information related to interoperability, while the second concerns the contribution of standards with respect to conceptual modeling in geographic information, before concluding remarks and future considerations are presented.

Geographic Information Modeling and Interoperability

Typically, organizations have assembled geographic databases for their explicit needs (e.g., censuses, land inventory and management, homeland security, sustainable development, etc.). Over time, a huge amount of geographic information has been accumulated. Considering the high demand for geographic information and the enhancement of web technologies, an increasing number of organizations have begun to disseminate of data they have collected. It is now well known that geographic information is widely available from multiple providers (i.e., governmental agencies, private organizations, etc.) and accessible on the web. Spatial data infrastructures (SDIs) have grown rapidly and, today, support easy access and use of geographic information (e.g., NSDI in the US, CGDI in Canada). Currently, those searching for geographic information can rely on SDIs, which offer a one-stop shop for geographic information in many countries.

However because the variety of available geographic information disseminated and further used without any formal definition, users experience problems in finding and obtaining the data that fit their needs, in analyzing such data, and in making sound decisions in different situations, thus limiting interoperability. Consequently, knowledge about data has become vital.

Knowledge about geographic information is collected in term of metadata. Metadata has been traditionally defined as "data about data" (ISO/TC211 2003c) and constitute a description of captured or modeled data in databases or applications. Basically, metadata refers to the content, the structure, the semantics, the lineage (source, collecting process, etc.), the quality (positional and content accuracy, etc.), the vintage, the resolution, the distribution format, the persons or institutions responsible for the data, etc. Accordingly, conceptual models, conceptual schemas, application schemas, database models, data dictionaries, feature catalogs, and repositories consist of so many approaches to document metadata specifically about the content and definition, the description of the structure, and the semantics of the geographic information. Considering the availability and accessibility to such descriptions of geographic information in a standardized manner, users of geographic information can interpret the information correctly and then identify the suitability of the data for their specific application (i.e., fitness for use). Therefore, metadata and especially conceptual models are crucial to support interoperability of geographic information.

Moreover in term of database design, it is an excellent practice to define properly the semantics, the geometry, the temporality, and integrity constraints of objects to be included in geographic databases and datasets. Referring to the interoperability framework depicted in the previous section, the content description of geographic databases consists in a fundamental agent's knowledge for reasoning and communicating with other agents, since it serves as its ontology. Ontology consists in a formal representation of phenomena with an underlying vocabulary including definitions and axioms that make the intended meaning explicit and describe phenomena and their interrelationships (Brodeur et al. 2003).

Database modeling benefits greatly from standards. Amongst existing standards, the ISO191*xx* suite of standards addresses specifically geographic information. The following section describes these standards, which contribute to the development of geographic database models and applications and to the standardized description of geographic information.

Contribution of Standardizations in Geographic Information to Spatiotemporal Modeling

Standards in geographic information contribute at various levels to the modeling of geographic information. The contributions range from the definition of types for data, which support the description of features and attributes, to the definition of standardized methodologies for handling descriptions of concepts of geographic databases or applications.

More specifically, the following standards take part either directly or indirectly to the modeling of geographic information:

- ISO/TS 19103 Conceptual Schema Language
- ISO 19107 Spatial Schema
- ISO 19108 Temporal Schema
- ISO 19109 Rules for Application Schema
- ISO 19110 Methodology for Feature Cataloging
- ISO 19111 Spatial Referencing by Coordinates
- ISO 19112 Spatial Referencing by Geographic Identifiers
- ISO 19115 Metadata
- ISO 19135 Procedures for Item Registration

This section is a review of the above standards' contributions to the modeling of spatiotemporal information. Hereafter, the expression "application schema" is used to refer to conceptual model, database model, or any other similar expressions. For a presentation of database modeling levels of abstraction, see the chapter entitled "▶ Modeling with Pictogrammic Languages."

Conceptual Modeling Language

Although ISO/TS 19103 Conceptual Schema Language (ISO/TC211 2005d) is intended to provide rules and guidelines for the use of a conceptual modeling language in ISO 191xx geographic information standards, namely the Unified Modeling Language (UML), it also includes definitions of a number of data types which provide a common ground for the representation of attributes and values.

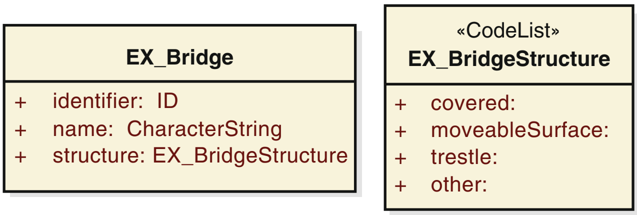
These rules are a help to the harmonization of conceptual models. Basically, ISO/TS 19103 rules follow standard UML (Object Management Group 2005), but introduce restrictions on the definition of classes, use of multiple inheritance, association role and multiplicity, and multiple class association. For instance, ISO/TS 19103 specifies a naming convention to increase readability and consistency: each word of class and relationship names are capitalized without space between them; each word of attribute, relation role, operation, and parameter names are capitalized except the first, without spaces between words. All examples in this chapter follow ISO/TS 19103 rules.

New stereotypes are also introduced: "CodeList", "Leaf", and "Union". "CodeList" is basically an extensible enumeration of character string values that an attribute can take. As Fig. 2 illustrates, the class *EX_Bridge* has an attribute *structure* of the type *EX_BridgeStructure*, which is a CodeList that enumerates its acceptable values, but is not limited to them. "Leaf" identifies packages that have no subpackages. A "Union" class is a class that gives to users a choice between multiple alternatives for the description of an attribute, but the attribute must use one, and only one, choice.

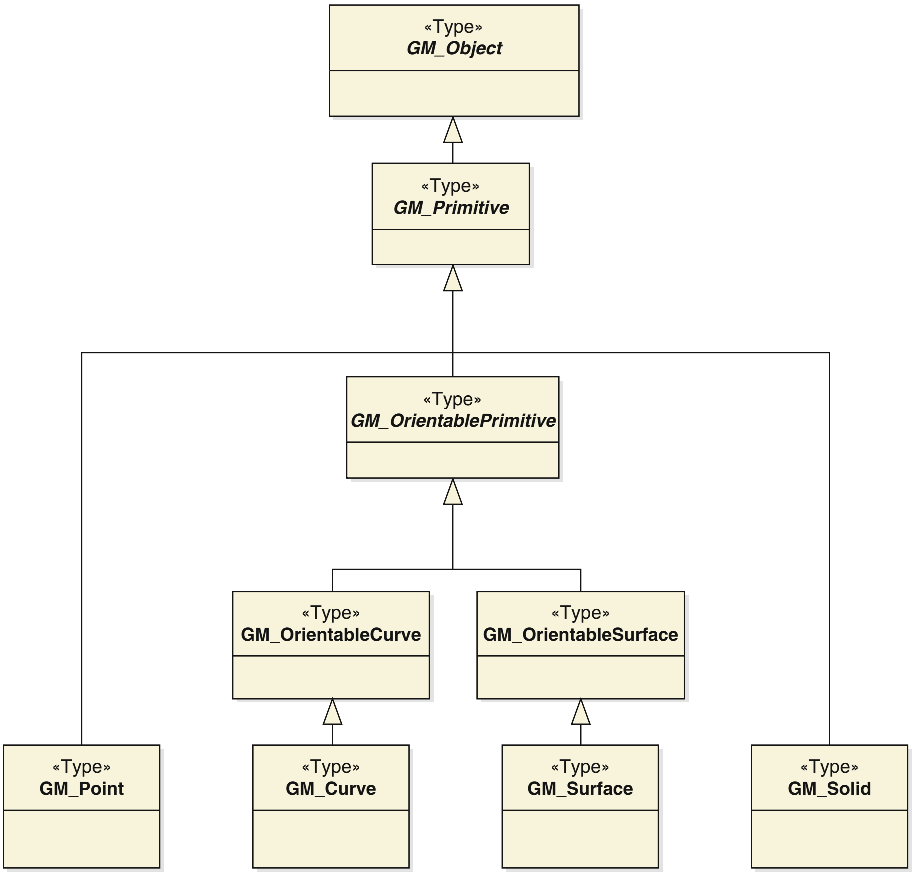
In addition, the use of standardized data types when building application schemas for geographic information supports attributes' meaning and better understanding of geographic information and, hence, enhances interoperability of geographic information. In ISO/TS 19103, these data types include primitive types, implementation and collection types, and derived types. Primitive types are those that are basic for the representation of elementary values. More specifically, they cover types for the representation of numeric, text, date and time, truth, multiplicities, and elementary enumerations:

- Numeric: Number, Decimal, Real, Integer, UnlimitedInteger, and Vector
- Text: Character, Sequence<Character>, and CharacterString
- Date and time: Date, Time, and DateTime
- Truth: Boolean {TRUE =1, FALSE = 0}, Logical {TRUE = 1, FALSE = 0, MAYBE = 0,5}, and Probability
- Multiplicity: Multiplicity, MultiplicityRange
- Elementary enumerations: Sign {+, −}, Bit {0, 1}, Digit {zero = 0, one = 1, two = 2, three = 3, four = 4, five = 5, six = 6, seven = 7, eight = 8, nine = 9}

A collection type is a container of multiple instances of a given type. Collections may have different characteristics with respect to ordering, duplication, and operations. Four types of collection are defined: *Set*, *Bag*, *Sequence*, and *Dictionary*. A *Set* consists of a definite number of objects of a given type. In a *Set*, an object appears once and only once (no duplicate are allowed)



Modeling with ISO 191xx Standards, Fig. 2
CodeList example



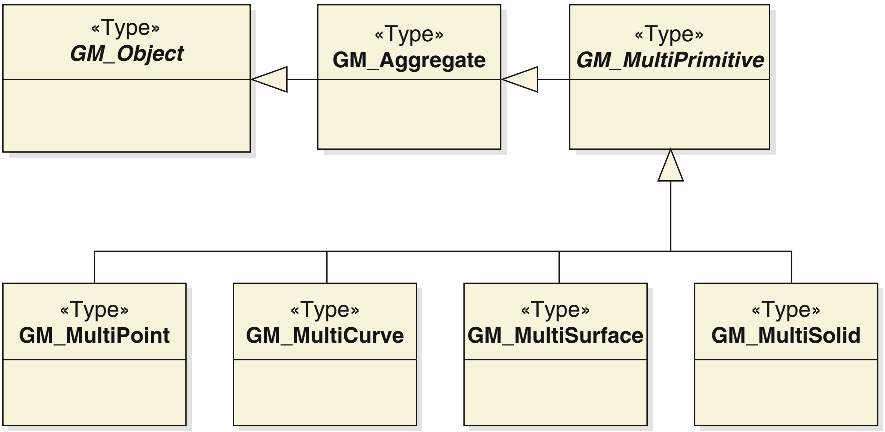
Modeling with ISO 191xx Standards, Fig. 3 Basic geometric primitives specified by ISO 19107 (ISO/TC211 2003a)

and objects are not ordered. A *Bag* is a similar structure to *Set*, but accepts duplicates. A *Sequence*, commonly known as a List, is a structure similar to a *Bag* in which objects are ordered. The *CircularSequence* is a specific type of *Sequence*, which does not define any last element making it circular. A *Dictionary* is an array-like structure that binds a key with a value, as a "hash table".

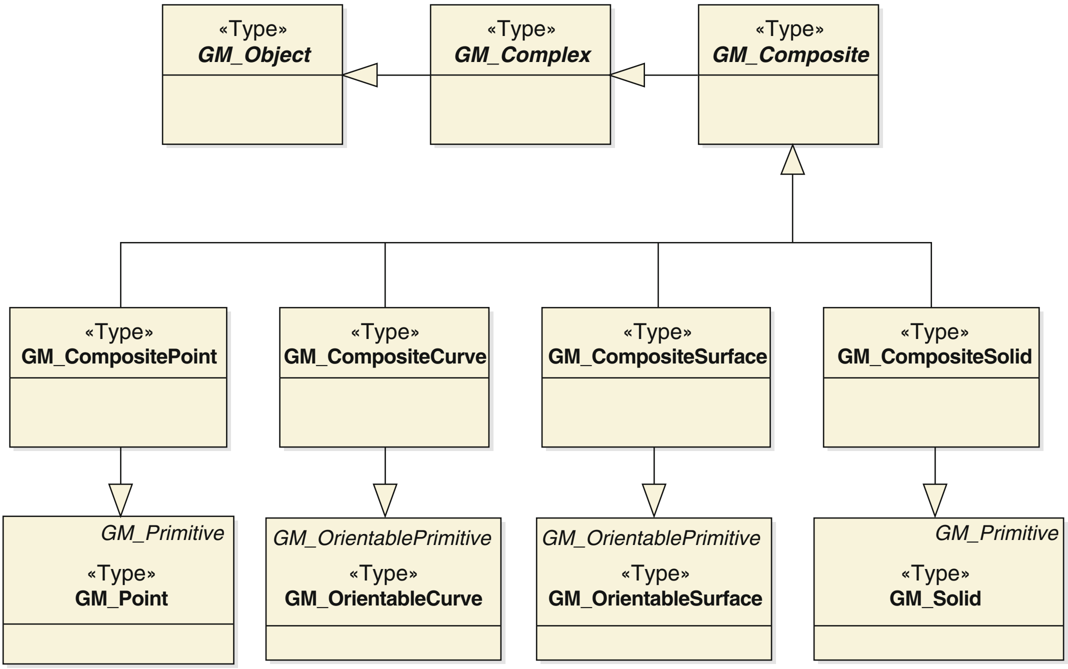
Derived types refer specifically to units of measure. A generic class UnitOfMeasure is introduced with a number of subclasses, which specify units for the different measures such as length, area, angle, time, etc.

Spatial Schema

The representation of spatial characteristics is fundamental in geographic information for the description of geographic feature, either in an application schema or a geographic database. Spatial characteristics encompass the geometry of the feature, its location with respect to a coordinate reference system, and its topological properties with other features. ISO 19107 Spatial Schema (ISO/TC211 2003a) defines in detail the geometric and topological characteristics that are needed to describe geographic features spatially. In ISO 19107, geometric characteristics are of three types: primitive (GM_Primitive), aggregate (GM_Aggregate), and complex (GM_Complex). Figure 3 shows ISO 19107 basic geometric primitives: GM_Point, GM_Curve, GM_Surface, and GM_Solid. They provide all components needed to depict the shape and the location of simple geographic features such as buildings, towers, roads, bridges, rivers, etc.



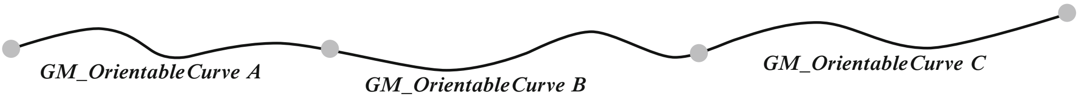
Modeling with ISO 191xx Standards, Fig. 4 Aggregate geometries specified by ISO 19107 (ISO/TC211 2003a)



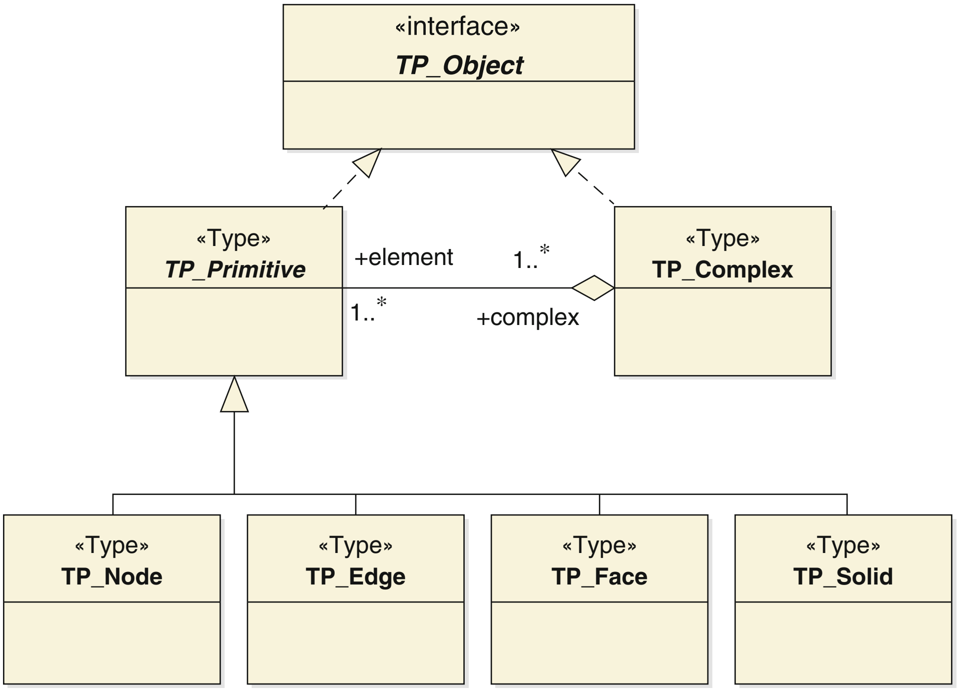
Modeling with ISO 191xx Standards, Fig. 5 Complex geometries specified by ISO 19107 (ISO/TC211 2003a)

Aggregate geometries (Fig. 4) depict features composed of multiple geometric primitives such as an archipelago, composed of multiple surfaces, or a campus, composed of small (or point like) and large (or surface like) buildings. Accordingly, ISO 19107 defines GM_Aggregate, which is a set of any kind of geometric types (GM_Object); GM_MultiPoint, a set of GM_Points; GM_MultiCurve, a set of GM_OrientableCurves; GM_MultiSurface, a set of GM_OrientableSurfaces; and GM_MultiSolid, a set of GM_Solids.

In some cases, geographic features have a more complicated geometric structure. That is the case for a road or a hydrographical network. Consequently, ISO 19107 has developed geometries for complexes: GM_Complex, GM_CompositePoint, GM_CompositeCurve, GM_CompositeSurface, GM_CompositeSolid (Fig. 5). They all consist of a set of GM_Primitives,



Modeling with ISO 191xx Standards, Fig. 6 Complex curve

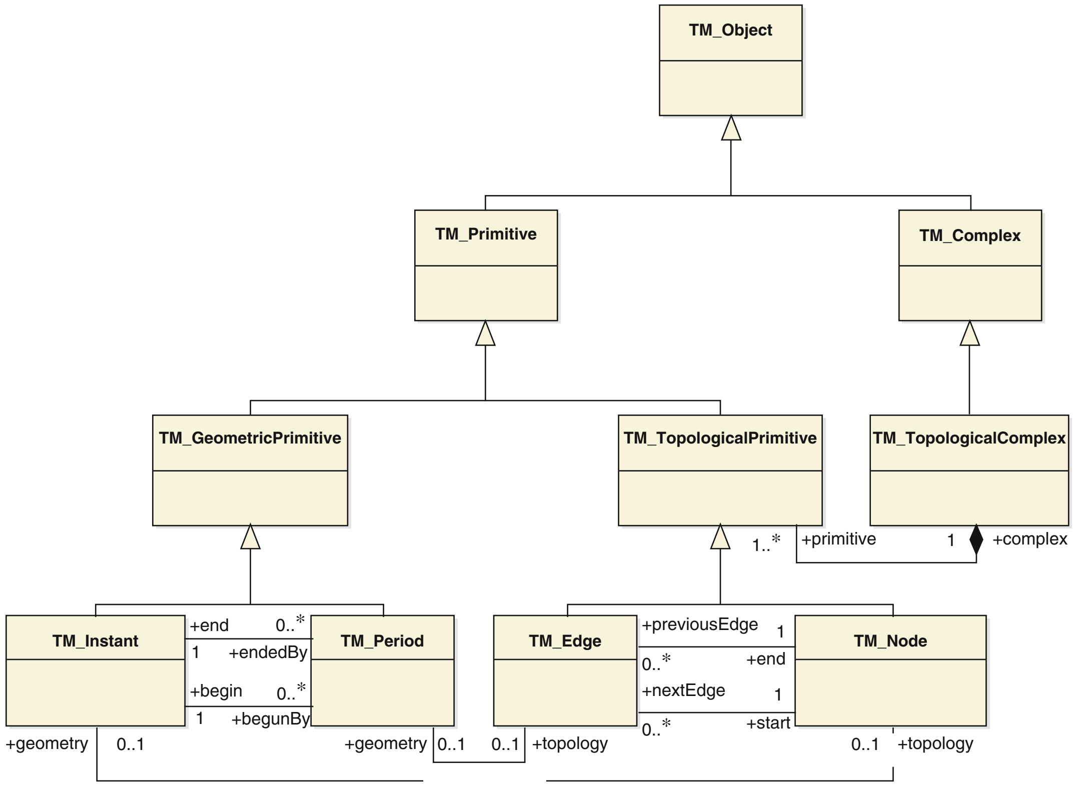


Modeling with ISO 191xx Standards, Fig. 7 Topological primitives and complex specified by ISO 19107

which have disjoint interiors, i.e., the interior of one geometry does not intersect with any other geometry. In a GM_Complex, primitives are joined together only by the way of a common boundary in order to form a unique geometry. For example, a GM_CompositeCurve is made of a set of GM_OrientableCurves where the first point of each curve, except for the first, corresponds to the last point of the previous one (Fig. 6) and the final GM_CompositeCurve possesses all the properties of a GM_OrientableCurve. Similarly, a GM_CompositePoint behaves as a GM_Point, a GM_CompositeSurface as a GM_OrientableSurface, and a GM_Composite Solid as a GM_Solid.

The location of all types of geometric primitives, aggregates, and complexes is described with coordinates that are referenced to a coordinate reference systems. This will be discussed in section "Spatial Referencing".

Topological primitives are needed to support complex geometric calculations, such as adjacency, boundary, and network analysis between geometric objects. In ISO 19107, topological primitives are structured similarly to the geometric structure with the definition of topological objects (i.e., TP_Objects). Accordingly, TP_Object is found at the top level of the structure and defines operations that are inherited by all its subordinates. There are two types of TP_Objects: TP_Primitive and TP_Complex (Fig. 7). TP_Primitives include TP_Node, TP_Edge, TP_Face, and TP_Solid, which parallel the geometric primitives presented earlier. TP_Complex aggregates TP_Primitives of different types up to the dimension of the



Modeling with ISO 191xx Standards, Fig. 8 Temporal primitives, topological primitives and complex specified by ISO 19108 (ISO/TC211 2002a)

complex. For example, a complex referring to a road network would include TP_Nodes and TP_Edges.

Temporal Schema

The representation of geographic features or attributes in an application schema or a geographic database may also have a temporal definition. For example, a bridge could be accessible only within specific periods of time in a day. ISO 19108 Temporal Schema (ISO/TC211 2002a) defines primitives, topological primitives, and topological complexes for the description of temporal characteristics. Basically, there are two temporal primitives: TM_Instant and TM_Period (Fig. 8). TM_Instant is used to describe a point in the time dimension whereas TM_Period represents a temporal extent, i.e., duration. A topological primitive, TM_Node or TM_Edge, provides information about its linkage or connectivity with other topological primitives. A set of topological primitives linked together forms a temporal topological complex, i.e., a TM_TopologicalComplex.

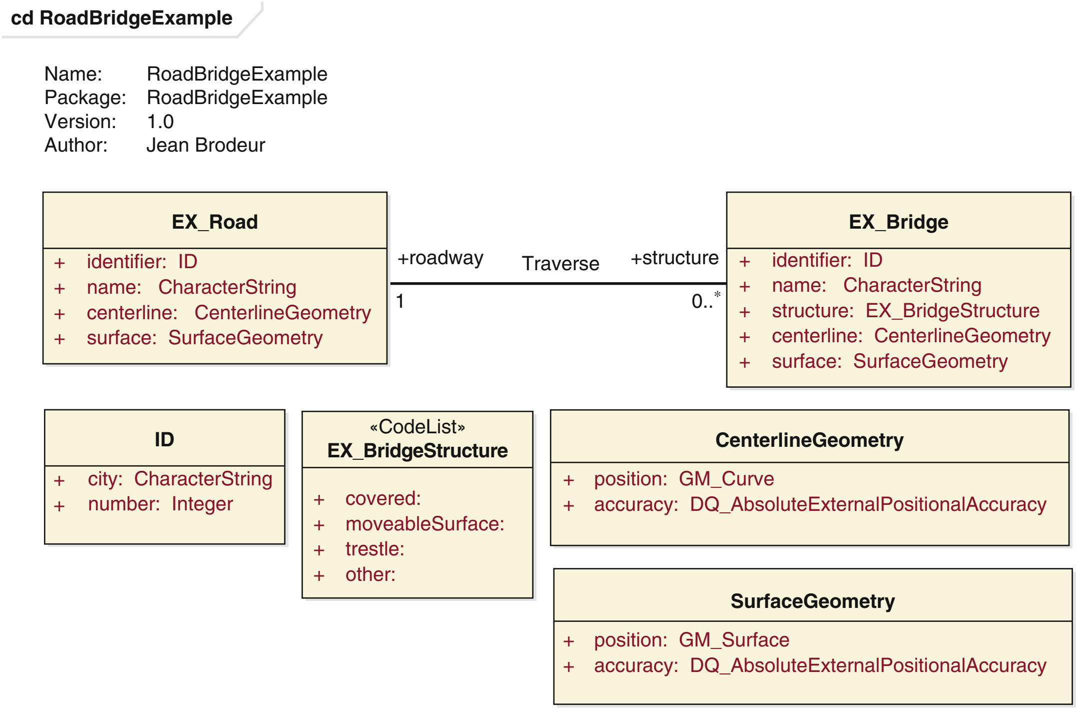
Application Schema

Geographic databases are collections of representations of geographic real world phenomena, i.e., geographic features that hold in a specific context and are of interest for users and/or applications. The manner in which a real world phenomenon is described depends on the perception of the observer about that phenomenon from which he/she abstracts characteristics of importance. Typically, an application schema is used to identify geographic as well as nongeographic features and their characteristics that are maintained in a geographic database or processed in an application.

For the same universe of discourse, observers may come to different conceptual models. Consequently, it is neither desirable nor of interest to standardize conceptual models. However, ISO 19109 Rules for Application Schema defines a number of rules to develop and maintain consistent application schemas. An application schema is a conceptual schema or model for geographic data that is required by one or more applications. Consistent application schemas help in the understanding, the acquisition, the processing, the analysis, the access, the representation, and the exchange of geographic information, in other words in the interoperability. On the one hand, ISO 19109 defines a metamodel called the *General Feature Model* (GFM). The GFM includes all the necessary elements for the description of features (GF_FeatureType), their properties (GF_PropertyType, GF_AttributeType, GF_AssociationRole, and GF_Operation), the different types of associations (GF_Association Type, GF_AggregationType, GF_SpatialAssociationType, and GF_TemporalAssociationType), generalization/specialization (GF_Inheritance-Relation) relationships, and constraints (GF_Constraint). The description of an attribute encompasses thematic (GF_ThematicAttributeType), location (GF_LocationAttributeType), and temporal attributes (GF_TemporalAttributeType). On the other hand, ISO 19109 defines rules to end up with consistent application schemas. These rules explain:

- The steps to create application schemas
- The general information of an application schemas (e.g., name and version)
- The integration of application schemas
- The use of UML to build application schemas
- The inclusion of metadata in an application schemas (e.g., quality of the geometry)
- The use of geometric, topological, and temporal primitives

Figure 9 illustrates an example of an application schema complying with ISO 19109 rules. It shows two feature classes: *EX_Road* and *EX_Bridge*. *EX_Road* has an *identifier* and a *name* thematic attribute. The attribute



Modeling with ISO 191xx Standards, Fig. 9 Road-bridge application schema example *identifier* is of the type *ID*, which is defined in the class *ID*, and the attribute *name* of the type *CharacterString* from ISO 19103. *EX_Road* has also two geometric attributes: *centerline* and *surface*. The attribute *centreline* is of the type *CenterlineGeometry*, which is described in the class *CenterlineGgeometry*. *CenterlineGeometry* has an attribute *position*, which describes the geometry of the road, and an attribute *accuracy* for the description of the positional accuracy of the geometry. The class *EX_Bridge* has a similar attribute arrangement to *EX_Road* with the addition of the attribute *structure* that takes its value from the code list *EX_BridgeStructure*. Finally, *EX_Road* is linked to *EX_Bridge* through the *Traverse* association. In this association, *EX_Bridge* plays the role *structure* in *EX_Road* and, reciprocally, *EX_Road* plays the role *roadway* for *EX_Bridge*. An instance of *EX_Bridge* must always be associated to an instance of *EX_Road*.

Feature Cataloging

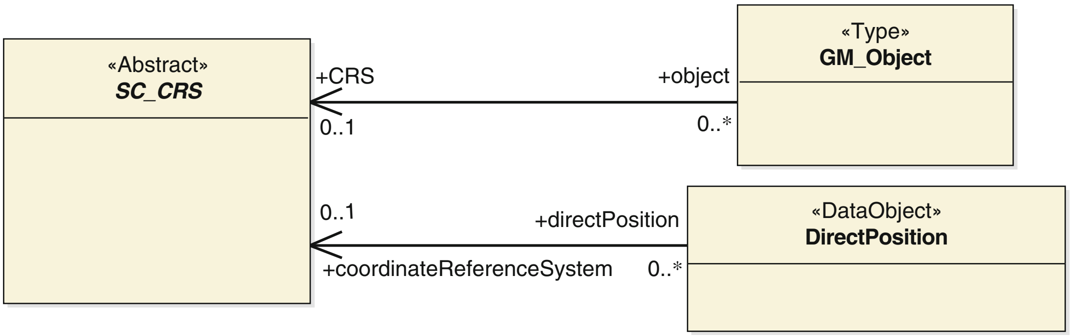
To describe an application schema completely, it is imperative to provide the semantics of each of its elements: classes, attributes, relationships, and so on. The object of ISO 19110 Methodology for Feature Cataloging is to define a mechanism for the documentation of the semantics of all application schema elements. Basically, the methodology for cataloging features agrees to the GFM discussed in section "Application Schema". A feature catalog includes a description of itself, feature types, feature property types, feature operations, feature attributes and attribute values, relationships, and association roles. As an example, the feature catalog of the road-bridge example could be depicted as in Table 1.

Spatial Referencing

Typically, the representation of a geographic feature position is done according to a coordinate reference system (CRS). In ISO19111 Spatial Referencing by Coordinates (ISO/TC211 2002b), a CRS is defined as a coordinate system which is related to an object by a datum. For the purpose of geodesy, the object of concern is the Earth and consists in the reference for the geometric depic-

Modeling with ISO 191xx Standards, Table 1 Road-bridge feature catalog example

FC_FeatureCatalogue
• Name: RoadBridgeExample
• Scope: Transportation network
• Field of application: Tracking and routing
• Version number: 1.0
• Producer: – Name: Jean Brodeur – Role: Custodian
• Functional language: English (ISO639_2.eng)
• Feature type: EX_Road, EX_Bridge
FC_FeatureType
• Name: EX_Bridge
• Definition: Structure erected along a travelled route to span a depression or obstacle and ensure the continuity of the road and railway network
• Code: 2139
• Abstract: False
• Feature catalogue: RoadBridgeExample
• Chracteristics: identifier, name, structure, centerline, surface
FC_FeatureAttribute
• Name: identifier
• Definition: unique identifier of the object
• Cardinality: 1
• Code: 2139.2
• ValueType: ID
FC_FeatureAttribute
• Name: name
• Definition: place-name of the feature
• Cardinality: 1
• Code: 2139.2
• ValueType: CharacterString
FC_FeatureAttribute
• Name: structure
• Definition: kind of construction
• Cardinality: 1
• Code: 2139.3
• ListedValue: covered, moveable surface, trestle, other
FC_ListedValue
• Label: covered
• Code: 1
• Definition: A bridge that has a building like cover to protect the bridge deck
FC_ListedValue
• Label: moveable surface
• Code: 2
• Definition: A bridge of which section can be moved to allow passage of vessels . . .



Modeling with ISO 191xx Standards, Fig. 10 Geometric object relationships with coordinate reference systems (*CRS*s)

tion of geographic features. ISO19111 provides the mechanism for describing a CRS, which can be either simple or compound. A simple CRS includes a description of the coordinate system (i.e., its axes) and the datum. A compound CRS is an aggregation of two or more simple CRSs. For example, a compound CRS may include a horizontal and a vertical CRS, which both have specific coordinate axes and datum.

CRS is of interest to conceptual modeling of geographic databases as it provides the description in which the positions of geographic features are known. Accordingly, it is important to note that each geometric object as defined in ISO 19107 (primitive, aggregate, or complex) is associated to a CRS and, as such, carries all parameters describing the CRS in which its coordinates are recorded (see Fig. 10).

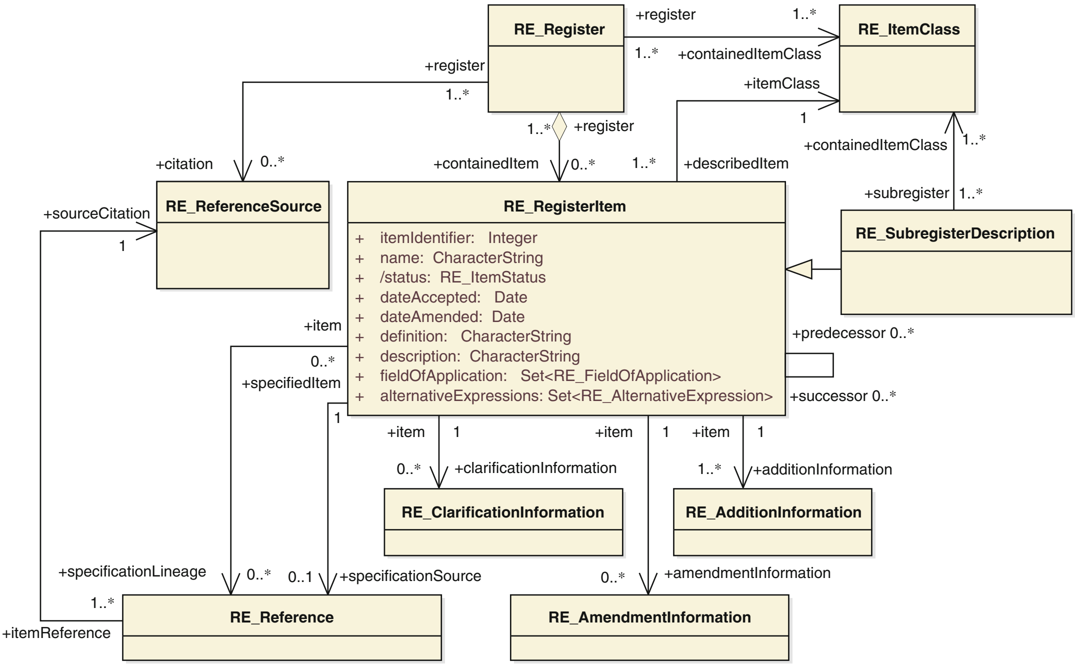
A geographic feature can also be located by the way of a geographic identifier, which provides a geographic reference through a label (e.g., King West street) or a code (e.g., postal code "J1H 1P1"). Geographic identifiers are usually organized in gazetteers, which are dictionaries of geographic identifiers. ISO 19112 Spatial Referencing by Geographic Identifiers (ISO/TC211 2003b) specifies a mechanism and components to describe geographic references based on geographic identifiers.

In this standard, a geographic identifier is represented by a SI_LocationInstance data type, which can be used in an application schema for documenting the location of a geographic feature. For example, "J1H 1P1" constitutes a SI_LocationInstance from the gazetteer of postal code of Canada and locates the north side of a portion of the *King West street* in *MyCity*.

Metadata

Metadata, commonly known as data about data, is meaningful information to better understand geographic data elements and data sets. ISO 19115 Metadata (ISO/TC211 2003c) sets the content and structure of geographic metadata. It covers topics like identification, constraints, quality, lineage, maintenance, spatial representation, reference system, content, portrayal, distribution, and application schema. Metadata elements are used to describe data sets and to provide additional information about geographic feature characteristics. For example, the road-bridge application schema illustrated in section "Application Schema" uses an attribute accuracy of DQ_AbsoluteExternalPositionAccuracy type from ISO 19115 in classes CenterlineGeometry and SurfaceGeometry to report the geometric accuracy of road and bridge instances. This data type and others from ISO 19115 can be used to report metadata about different facets of geographic data and therefore to benefit conceptual modeling.

In addition, ISO 19115 includes the application schema (MD_ApplicationSchemaInformation) and a reference to the feature catalog (MD_ContentInformation) as metadata items for the description of datasets, data collections, or series. This makes the application schema and the feature catalog living components for the



Modeling with ISO 191xx Standards, Fig. 11 Geometric object relationships with CRSs

use of geographic information to support better understanding of the data but also analysis and reasoning purposes. Consequently, metadata contributes to semantic interoperability of geographic information.

Register

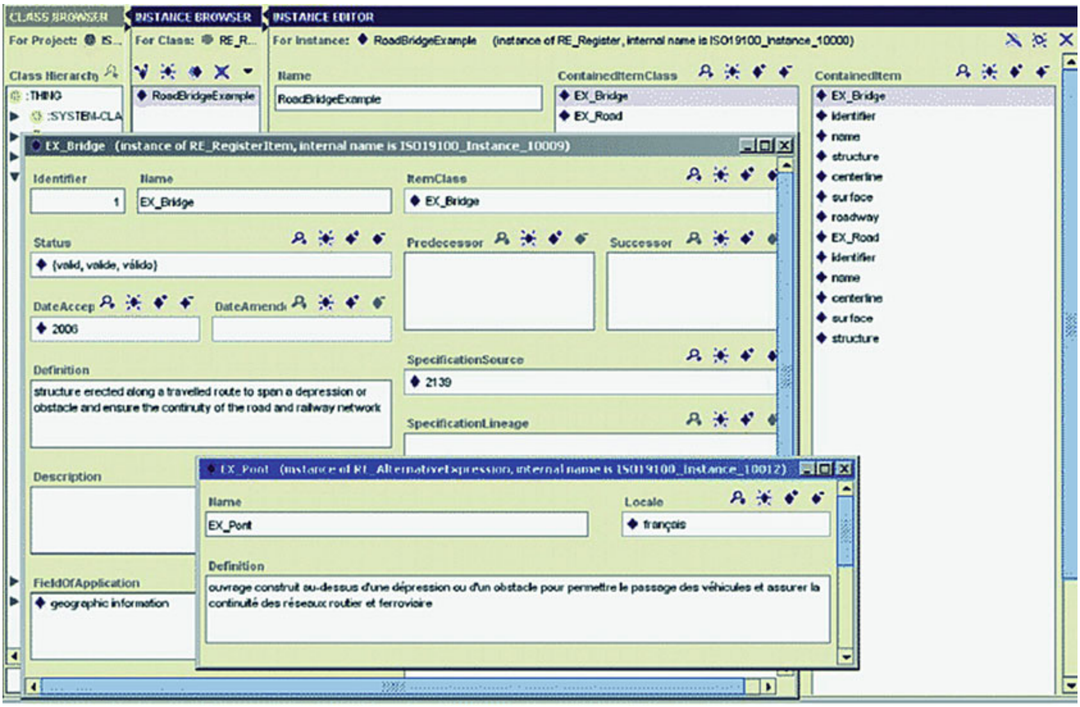
According to ISO 19135 Procedures for Item Registration (ISO/TC211 2005c), a register consists of a collection of object identifiers with definitions recorded in a file or set of files. It establishes the identity of concepts of interest within a namespace for geographic applications and databases. A registry is a complete system to ensure the appropriate management of a register including a register owner, a register manager, and submitting organizations, which sponsor the register.

ISO 19135 establishes the structure of a register (RE_Register) and the rules for the proper management of registers. A register contains items that exist as components of an item Class (Fig. 11). An item (RE_RegisterItem) is described by an identifier, a name, a status, an acceptance date, an amendment date, a definition, a description, a field of application, and a set of alternative expressions. Items may evolve in time and a register allows for maintenance and tracking of modifications.

Registers contribute to geographic database modeling in maintaining semantics of concepts that application schemas, feature catalogs, or geographic databases may reuse and associate to model elements or data. They support concepts' multilingual representation via alternative expressions that are associated with a specific locale (i.e., the language and country identification). Figure 12 illustrates a register of the road-bridge example that shows EX_Bridge as a contained item of the register with EX_Pont as an alternative expression in French.

Key Applications

The importance of spatial database modeling based on the ISO191xx suite of standards lies mostly in increasing interoperability of geographic information. Geographic data described in a common structure using the same



For Project: IS...

For Class: RE_R...

◆ RoadBridgeExample

For Instance: ◆ RoadBridgeExample (instance of RE_Register, internal name is ISO19100_Instance_10000)

Name: RoadBridgeExample

ContainedItemClass: ◆ EX_Bridge ◆ EX_Road

ContainedItem: ◆ EX_Bridge ◆ identifier ◆ name ◆ structure ◆ centerline ◆ surface ◆ roadway ◆ EX_Road ◆ identifier ◆ name ◆ centerline ◆ surface ◆ structure

EX_Bridge (instance of RE_RegisterItem, internal name is ISO19100_Instance_10009)

Identifier: 1

Name: EX_Bridge

ItemClass: ◆ EX_Bridge

Status: ◆ {valid, valide, válido}

Predecessor

Successor

DateAccep: ◆ 2006

DateAmend

Definition: structure erected along a travelled route to span a depression or obstacle and ensure the continuity of the road and railway network

SpecificationSource: ◆ 2139

SpecificationLineage

Description

FieldOfApplication: ◆ geographic information

EX_Pont (instance of RE_AlternativeExpression, internal name is ISO19100_Instance_10012)

Name: EX_Pont

Locale: ◆ français

Definition: ouvrage construit au-dessus d'une dépression ou d'un obstacle pour permettre le passage des véhicules et assurer la continuité des réseaux routier et ferroviaire

Modeling with ISO 191xx Standards, Fig. 12 Road-bridge register example

data types facilitates their access, understanding, integration, and use through SDIs. It is easier for users to find and get geographic data that fit their application when, for example, they are published on a web portal such as the CGDI Discovery portal in Canada (http://geodiscover.cgdi.ca/gdp/) or Geospatial One Stop in the US (http://gos2.geodata.gov/wps/portal/gos) using these standards where all the information is described and presented similarly. From a user point of view, it supports applications such as disaster management, global warming, sustainable development, traffic management, etc. which typically need to integrate data from different sources, themes, levels of details, such as road, drainage, railways, satellite images, relief and so on.

Future Directions

The contribution of standard-based geographic data modeling to interoperability of geographic information and the contribution of international standards in geographic information to geographic data modeling has been explained in this chapter.

Standards in geographic information play an important role in geographic data modeling. As noted here, a number of standards provide data types for the representation of geographic information. These data types include basic types, such as numbers, texts, and dates, geometric types (e.g., point, line and surface), temporal types (e.g., instant and period), etc. This is the case for ISO/TS 19103 (conceptual schema language), ISO 19107 (spatial schema), and ISO 19108 (temporal schema). Other standards provide data types for the description of ancillary information. This is the case for ISO 19111 (spatial referencing by coordinates), ISO 19112 (spatial referencing by geographic identifiers), ISO 19115 (metadata), and ISO 19131 (data product specifications). Finally, there are standards that concern the elaboration of geographic data models. These standards provide structures and rules for the elaboration of application schemas and documentation of concepts. This is the case for ISO 19109 (rules for application schema), ISO 19110 (methodology for feature cataloging), and ISO 19135 (procedures for item registration).

Additionally, standards in geographic information bring a common domain ontology, which describes a set of concepts that are needed for the overall aspects of geographic information. This sets the foundation for the Semantic Web in the geographic information realm. Some open frameworks like GeOxygene (Badard and Braun 2004) are already available to help in the interoperable development and deployment of geospatial applications over the internet.

Cross-References

- ▶ Geospatial Semantic Web, Interoperability
- ▶ Metadata and Interoperability, Geospatial
- ▶ Modeling with Pictogrammic Languages
- ▶ OGC's Open Standards for Geospatial Interoperability
- ▶ Vector Data

References

Badard T, Braun A (2004) OXYGENE: a platform for the development of interoperable geographic applications and web services. In: Proceedings of the 15th international workshop on database and expert systems applications (DEXA'04), Zaragoza, 30 Aug–3 Sept 2004

Brodeur J, Bédard Y, Proulx MJ (2000) Modeling geospatial application databases using UML-based repositories aligned with international standards in geomatics. In: Proceedings of the 8th ACM symposium on advances in geographic information systems (ACMGIS), Washington, DC, 6–11 Nov 2000

Brodeur J, Bédard Y, Edwards G, Moulin B (2003) Revisiting the concept of geospatial data interoperability within the scope of a human communication process. Trans GIS 7:243–265

Institute of Electrical and Electronics Engineers (1990) IEEE standard computer dictionary: a compilation of IEEE standard computer glossaries. IEEE Computer Society, New York

ISO/TC211 (2002a) ISO19108:2002 geographic information-temporal schema. ISO, Geneva

ISO/TC211 (2002b) ISO19111:2002 geographic information-spatial referencing by coordinates. ISO, Geneva

ISO/TC211 (2003a) ISO19107:2003 geographic information-spatial schema. ISO, Geneva

ISO/TC211 (2003b) ISO19112:2003 geographic information-spatial referencing by geographic identifier. ISO, Geneva

ISO/TC211 (2003c) ISO19115:2003 geographic information-metadata. ISO, Geneva

ISO/TC211 (2005a) ISO19109:2005 geographic information-rules for application schema. ISO, Geneva

ISO/TC211 (2005b) ISO19110:2005 geographic information-methodology for feature cataloguing. ISO, Geneva

ISO/TC211 (2005c) ISO19135 geographic information-procedures for item registration. ISO, Geneva

ISO/TC211 (2005d) ISO/TS19103:2005 geographic information-conceptual schema language. ISO, Geneva (2005)

ISO/TC211 (2007) Geographic information/geomatics homepage. http://www.isotc211.org/. Accessed 2 Sept 2007

Object Management Group (2005) Unified modeling language: superstructure, v2.0. OMG, Needham

OGC (2007) Open geospatial consortium homepage. http://www.opengeospatial.org/. Accessed 2 Sept 2007

Modeling with Pictogrammic Languages

Yvan Bédard and Suzie Larrivée
Department of Geomatics Sciences, Centre for Research in Geomatics, Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support, Laval University, Quebec City, QC, Canada

Synonyms

Perceptory pictograms; Spatial modeling language extension; Spatiotemporal modeling language extension

Definition

"Spatial databases" consist of large groups of data structured in a way to represent the geographic features of interest to the users of a system. Spatial database models are schematic representations of these data. Database models are created to design and document the system, to facilitate communication and to support programming. They are created using CASE tools (computer-assisted software engineering). CASE tools support schema drawing, dictionaries and code generation. Database schemas are typically represented with a graphical language such as UML (Unified Modeling Language; see http://www.uml.org and Fowler 2004).

"Database models" can represent (1) users' real-life views of the data of interest, (2) developers' views of the potential organization of these data for a family of technologies, or (3) their final implementation on a specific platform. For example, in the standard Model-Driven Architecture (MDA) method (http://www.omg.org/mda/), these three models represent three levels of abstraction and are respectively called CIM (computation-independent model), PIM (platform-independent model) and PSM (platform-specific model). In other methods, they may be called conceptual, logical and physical models as well as analysis, design and implementation models.

"Pictograms" are symbols aimed at facilitating modeling. Different sets of pictograms have been proposed. This chapter presents those used by the CASE tool Perceptory (http://sirs.scg.ulaval.ca/perceptory) since they are the most widely used, they were designed to allow developers to keep their method, and they were thoroughly tested as implementations of UML stereotypes. In Perceptory, they aim at hiding the complexity of geometric primitives in CIM and PIM models. They can serve other purposes as well and have been implemented in other CASE tools (Miralles 2006).

Historical Background

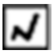
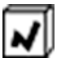
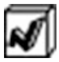
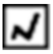
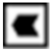
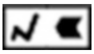
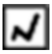
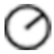
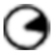
In the field of GIS, pictograms were first proposed in 1989 by Bédard and Paquette to simplify how Entity-Relationship (E/R) models depicted the geometry of cartographic features. It was then called "Sub-Model Substitution" technique as the main goal was to remove from the spatial database model those geometric primitives with their data elements and relationship (considered of no interest to the user) and to replace them by simple symbols showing only the information of interest to the users (i.e. the features' shape). This first solution was tested in several projects and enhanced over time to lead to the development of Modul-R (Bédard et al. 1996, 1992; Bédard and Larrivée 1992), the first spatio-temporally extended E/R which led to Orion, the first GIS-compatible CASE tool in 1992 (Bédard et al. 1992). This first solution has influenced several researchers afterwards. Examples of methods or tools using pictograms for spatial databases include Perceptory (Bédard et al. 2004; Bédard 1999) which is used in over 30 countries, Software Development Process Model with Objecteering (Miralles 2006), MADS (Parent et al. 2006), CONGOO (Pantazis and Donnay 1996), UML-Geoframe with ArgoCASEGEO (Filho et al. 2004), and STER (Tryfona et al. 2003).

In 1996, Modul-R pictograms were revisited to integrate three paradigms: object-orientation (OO), plug-in (module, blade, cartridge) and a pragmatic symbiotic approach (Bédard 1999). Object-orientation allowed for more expressive power and was first tested with UML in its prerelease days. The plug-in approach led to define the pictograms and their syntax as a module, i.e. a specialized language designed to extend standard languages (e.g. UML, E/R, English). This allowed for enriching one's modeling language and tool rather than requiring to adopt new ones. For instance, in addition to Perceptory, these pictograms have been used with commercial and open-source CASE tools such as Oracle Designer, Objecteering and others while being also used to describe spatial integrity constraints, to compare database semantics and to improve software user-interfaces. With regards to the symbiotic approach, it came from cognitive studies and pragmatics lessons resulting from several projects with practitioners, including very complex ones. It helped to find a better balance between human abilities, language requirements, database design methods and commercial software constraints. Practical projects clearly indicated the need to better support unexpected complex situations, to simplify the pictograms along with their syntax, and to better balance the content of the graphical schema with the ontological content of the dictionary (i.e. simpler schemas, increased use of natural and formal languages in the dictionary). This was a departure from the trend of that period to rely increasingly on graphical depictions. Such novel approach and the arrival of UML led to developing Perceptory. This approach also goes beyond the leading tendency to perceive "modeling" solely as a schema-building exercise since in fact it is not; a schema without clear and complete semantics is meaningless and its robustness cannot be validated. Accordingly, good spatial database modeling becomes an ontological exercise. For example, Perceptory provides specialized spatial and temporal sections in its dictionary (as can be added to other CASE tools). In the remaining of this chapter are presented the scientific fundamentals of modeling spatial databases with pictograms, using examples from the UML-based Perceptory CASE tool.

Scientific Fundamentals

"Pictograms" aim at supporting the expression of any feature's spatial and spatio-temporal properties into a consistent manner that is compatible with various human-oriented languages (e.g. UML, Entity-Relationship, English, French).

"Syntax rules" dictate the way to combine and position pictograms in a model or document. These rules also dictate how to use special characters (0–9 N,). Properly combining pictograms, with or without characters, makes it possible to express complex cases of geometry and spatio-temporality, namely: facultative, mandatory, alternate, aggregate, multiple, and derived.

A "pictogrammic expression" includes one or several pictograms which are positioned in a precise manner with pertinent digits according to a syntax. Such a pictogrammic expression completely describes the spatial, temporal or spatio-temporal properties of either (1) a feature, (2) where and when an attribute value is valid within an object geometry or existence, or of (3) a relationship between features. For example, in Perceptory, the simple expression  is made of only one pictogram and represents a simple 1D geometry in a 2D universe. Similarly, the expression  represents the same geometry in a 3D universe while the expression  adds thickness to this geometry. On the other hand, the expression   (i.e. 1D OR 2D) has a different meaning from the previous ones and from the expression  (i.e. 1D + 2D) or from the expression  0, N. In a similar manner, the simple expression  represents one instant, the expression  represents one period of time. More complex temporal and spatio-temporal expressions can be made.

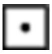
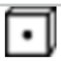
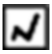
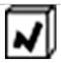
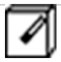
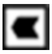
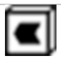
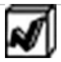
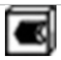
Grouping pictograms and syntactic rules commonly used together allows one to form a specialized graphical language called "PVL" (Plug-in for Visual Languages). A PVL, as introduced in Bédard (1999), allows extending a modeling language with a tested method that is compatible with other PVLs if needed. For example, one may decide to use only a small group of Perceptory pictograms to make a 2D spatial PVL (i.e. a language to depict plane geometries of geographic features) while later on, if needed, use additional Perceptory pictograms to have a 3D spatio-temporal PVL. A pictogrammic expression is sometimes called a PVL expression.

The pictograms high level of abstraction facilitates the making of database models, reports, specifications, spatio-temporal integrity constraints, user interfaces, and similar tasks of a system development workflow. They hide the complexity inherent to the description of geometric and temporal primitives and relationships as well as implementation and standard-related issues. In particular, they facilitate the building, editing, communication and validation of spatio-temporal database models as well as their translation into efficient data structures. In spite of such translation rules, the PVL are independent from commercial software and numerous standards.

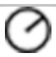
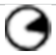
The pictograms were first created for spatial database modeling and are best described in such a context. Accordingly, the present chapter describes the pictograms implemented as UML stereotypes in Perceptory object class model. In such a context, the PVL allows the analyst or designer to describe the spatial and temporal properties of the elements depicted in an object class schema. Perceptory pictograms support 0D, 1D, 2D and 3D geometries for objects located in 2D or 3D universes (see Table 1). Supported temporalities are 0D (instant) and 1D (period) (see Table 2). Supported combinations are simple, complex (aggregate), alternate (exclusive OR), multiple (AND), spatio-temporal and hybrid (combinations of any of the above) (see Tables 3, 4 and 5). Supported minimum multiplicities include facultative (0), mandatory (1), specific number, and many (N), while maximum multiplicities include the three latter. Special cases are "any possibility", "not yet defined" and "complicated", the latter pointing to a textual description in the repository (when easier to read). All geometries and temporalities can be indicated as "measured" or "derived from other attributes, objects, relationship using calculations, spatial or temporal analysis". Having no geometry or temporality is also accepted. Pictogrammic expressions may describe object classes, association classes, attributes, and may be used within operations.

Examples of the use of pictogrammic expressions for UML object classes are presented hereafter. Figure 1 describes an accident as a an instantaneous event with a geometry defined as a point. Figure 2 shows a case where users want to keep information about the existence of commercial buildings (dates of construction and destruction), about the evolution of their commercial value (attribute data with their period of validity) and about the evolution of its polygonal representation if it is enlarged or modified. Figure 3 illustrates a case of aggregated complex geometry while Fig. 4 shows cases of simple and of alternate geometries. At last, Fig. 5 shows a case of multiple geometry where the first pictogram expresses the fact that every building is represented by simple polygon at large scales and the second

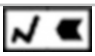
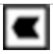
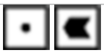
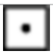
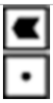
Modeling with Pictogrammic Languages, Table 1 Simple pictogrammic expression for geometry

	2D space	3D space	Examples of cases
0D geometry			Hydrants when they are all represented by points
1D geometry			Road segments when they are all represented by lines
			Electric poles when they are all represented by vertical lines
2D geometry			Lakes when they are all represented by polygons
			Walls when they are all represented by vertical plans
3D geometry			Buildings when they are all represented by solids

Modeling with Pictogrammic Languages, Table 2 Simple pictogrammic expressions for temporality

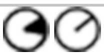
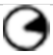
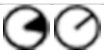
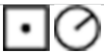
		Examples of cases
0D temporality		Existence of accidents; traffic flow of a road segment
1D temporality		Existence of a building; duration of its commercial use; duration of its ownership by a given person

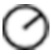
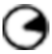
Modeling with Pictogrammic Languages, Table 3 Syntax for advanced 2D and 3D spatial pictogrammic expressions

Geometry	Examples of syntax	Examples of cases
Aggregate geometry		
(Complex)		Hydrographic networks composed of 1D rivers and 2D lakes (i.e. aggregate of different geometries)
(Simple)	 1,N	Some municipalities may include several 2D geometries such as islands (i.e. aggregate of similar geometries)
Alternate geometry (on same line)		Buildings having a 0D shape if area <1 ha OR a 2D shape if area >1 ha (Exclusive OR)
Facultative geometry	 0,1	Buildings in database may have no geometry if area <0.2 ha, or a 0D shape if area >0.2 ha
Multiple geometry (on different lines)		Every municipality has a 2D shape AND a OD location (e.g. downtown). See Bédard et al. (2004) for detailed examples.

N.B. same syntax for 2D and 3D pictograms

Modeling with Pictogrammic Languages, Table 4 Syntax for advanced temporal and spatiotemporal pictogrammic expressions

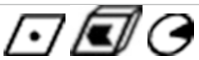
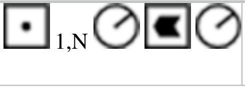
Temporality	Examples of syntax	Examples of cases for feature existence and states
Alternate temporality (on same line)		Forest fires lasting several days OR 1 day (if temporal resolution is 1 day); water level data varying continuously when opening/closing the dam OR remaining stable for a period once a level is reached
Facultative temporality	 0,1	Houses in database may need NO construction and demolition dates IF area <0.2 ha
Multiple temporality (on different lines)		Hurricane existence defined by a date of beginning and a duration for some purposes, AND by a unique date of maximum peek for other purposes. Buildings commercial value considered stable for the whole year for tax purposes but as being valid only the day when the building was assessed for market analysis purposes.
Spatio-temporality		Position of a moving vehicle. The temporal pictogram affects the spatial pictogram on its left

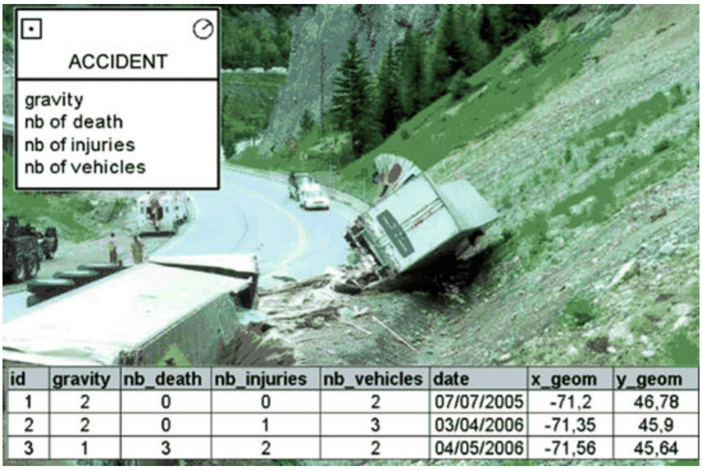
N.B. Selecting between  or  depends on the temporal granularity defined into the repository for each class, attribute and geometry

line of pictograms indicates that some (but not all) buildings may have a second geometry, either a point or a line, depending on their size, for small scale maps (usually to properly place symbolic representations). See Bédard et al. (2002) for more details.

From a UML point of view, these pictogrammic expressions are implemented as stereotypes

Modeling with Pictogrammic Languages, Table 5 Syntax and pictograms for special cases

Derived geometry or temporality "italic pictogram"		Municipality centroids derived from their polygons; 3D buildings derived from 2D buildings with number of floors; duration of commercial use derived from permits
Hybrid expression (combination of any pictos above)		A set of individual cyclists continuously moving during a race or forming a group that changes its size during the race
Default multiplicity	If no multiplicity is written immediately after a pictogram, the 1,1 multiplicity is implied	
Any possibility	* * *	"wildcard pictogram" meaning no predefined shape or temporality, and no restriction on the geometry or temporality
Complicated	! ! !	Better explained textually in the dictionary than using a complicated PVL expression in a schema. Replaces a long hybrid expression if desired
Not yet defined	? ? ?	During the process of designing a database, one may anticipate a need for geometry or temporality, but ignore which one and will replace it later by a regular pictogram

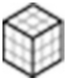
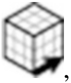
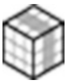
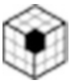


ACCIDENT

gravity
nb of death
nb of injuries
nb of vehicles

id	gravity	nb_death	nb_injuries	nb_vehicles	date	x_geom	y_geom
1	2	0	0	2	07/07/2005	-71,2	46,78
2	2	0	1	3	03/04/2006	-71,35	45,9
3	1	3	2	2	04/05/2006	-71,56	45,64

Modeling with Pictogrammic Languages, Fig. 1 Example of simple pictogrammic expressions for the geometry and existence of a UML object class accident

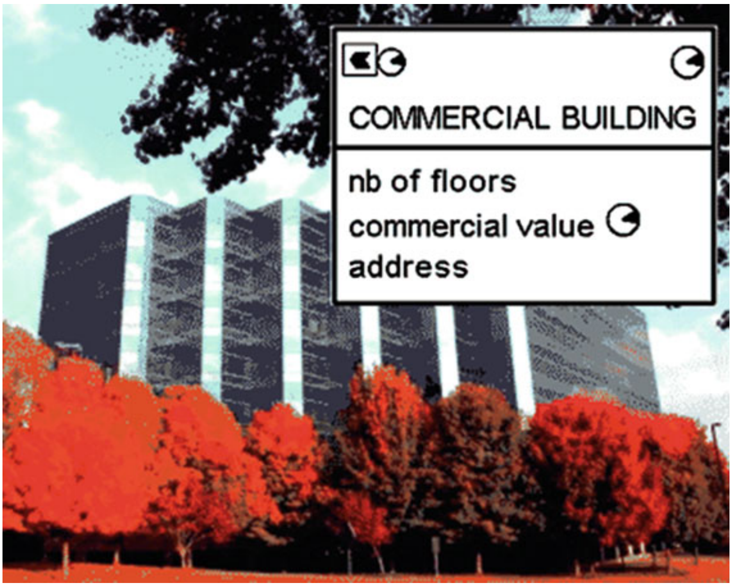
(a formal way of extending UML) and are built on-the-fly in Perceptory. Using such pictogrammic expressions has also proved to be useful to model spatial multidimensional databases (or datacubes) as used in spatial data warehousing and SOLAP (Spatial On-Line Analytical Processing). These datacubes pictogrammic expressions include datacube , data dimension , member , measure  and are compatible with the previous spatial and temporal pictograms. They are all supported by Perceptory.

Key Applications

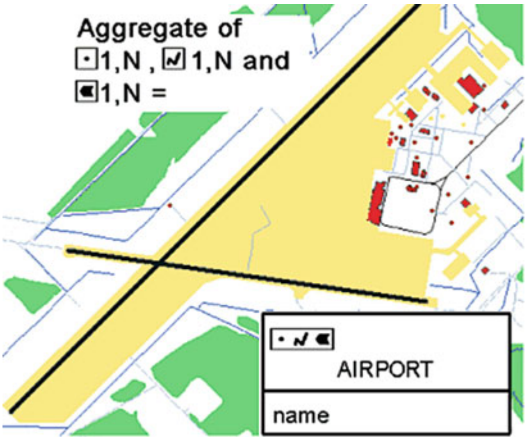
Pictogrammic languages, if sufficiently expressive and usable, can serve several purposes. The following paragraphs further describe the primary key application, i.e. spatial database modeling, plus other applications of interest.

Using Pictogrammic Languages for Spatio-temporal Database Modeling

Modeling databases for GIS applications has always posed several challenges for system analysts, system developers as well as for their clients whose involvement into the development



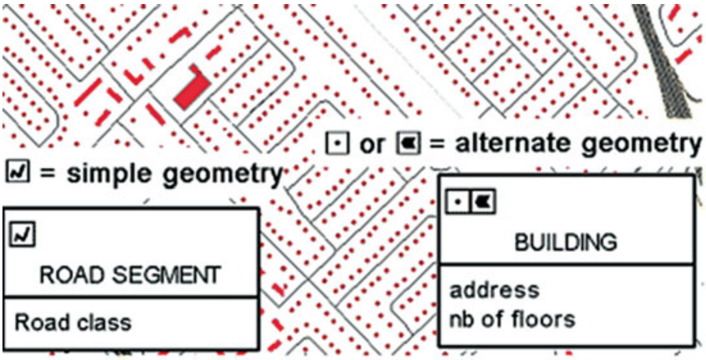
Modeling with Pictogrammic Languages, Fig. 2 Example of a spatio temporal pictogrammic expression, a temporal expression for the existence of the UML object class and of another one to keep track of the evolution of one attribute



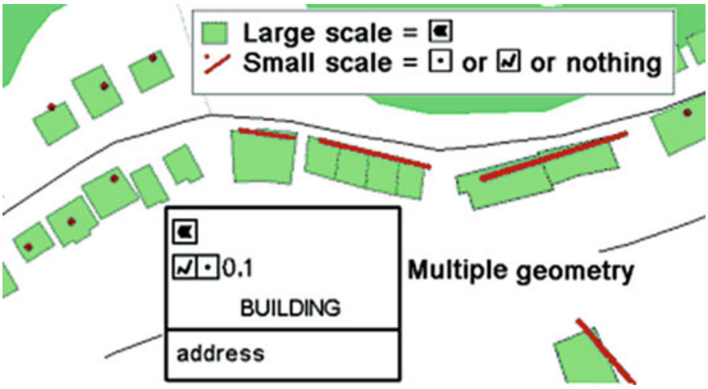
Modeling with Pictogrammic Languages, Fig. 3 Example of a complex aggregate geometry for Airport object class, that is an aggregate of points, lines and polygons (Data from ministère des Ressources naturelles et de la faune du Québec)

of such a project is not a familiar endeavor. Used with well-known modeling techniques, pictogrammic expressions help to meet these challenges (Bédard et al. 2004; Filho et al. 2004; Pantazis and Donnay 1996; Parent et al. 2006) and are commonly used in different methods such as relational database design with UML (cf. the UML relational stereotypes in Naiburg and Maksimchuk 2001). Extending CASE tools and modeling methods in such a way allows analysts and designers to work at a higher level of abstraction for the first steps of a spatial database project. As presented in Fig. 6, once high-level models are completed (e.g. Perceptory CIM), they can be translated and enriched to give more technical models which are closer to implementation (e.g. OGC PIM, models by Brodeur and Badard in the chapter entitled "▶ Modeling with ISO 191xx Standards") and finally highly technical models which specific to one implementation on one platform (e.g. ESRI Shapefiles PSM). Such multi-level approach is typical of good software engineering methods as exemplified in Fig. 6 and by the following landmark methods:

- The Object Management Group (OMG) Model-Driven Architecture (MDA) which has three levels of models: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM);
- Zachman Framework (Frankel et al. 2003; Sowa and Zachman 1992) which has a business or enterprise model, a system model and a technology model (also called semantic, logical and physical models);



Modeling with Pictogrammic Languages, Fig. 4 Example of a single geometry pictogrammic expression (where each instance is represented by one line) and of an alternate geometry (where small buildings are represented by a point and large ones by a polygon) (Data from ministère des Ressources naturelles et de la faune du Québec)

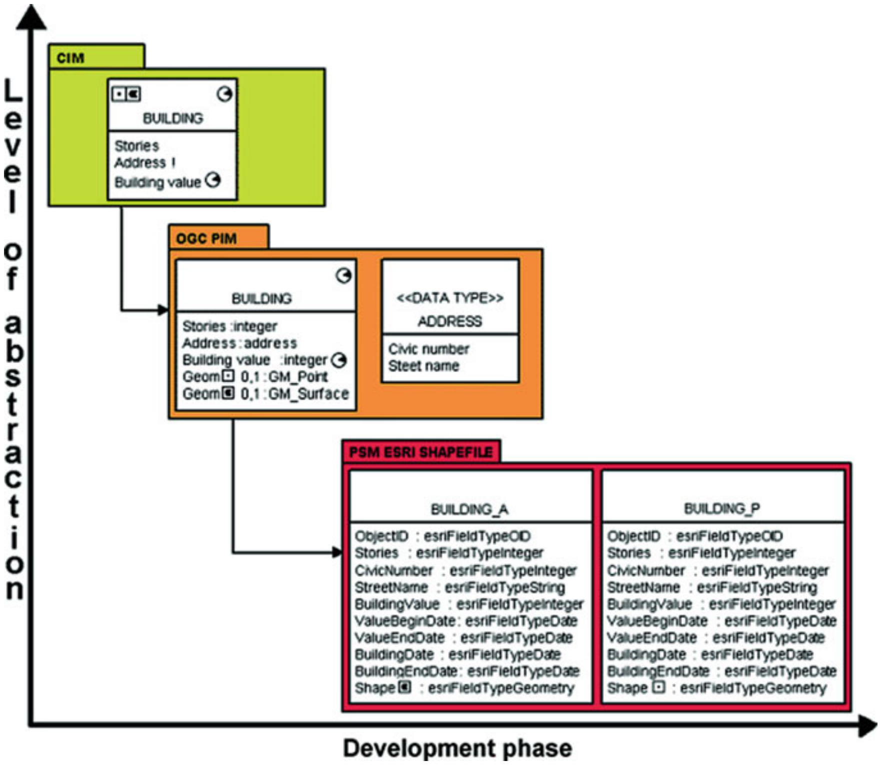


Modeling with Pictogrammic Languages, Fig. 5 Example of a multiple geometry pictogrammic expression, where at large scale (e.g. 1:1000), buildings are represented by a polygon and at small scale (e.g. 1:20,000), they are represented by a point, a line or nothing (Data from Research and Development Defence Canada and from ministère des Ressources naturelles et de la faune du Québec)

- Rational Unified Process (RUP) which has a domain model, an analysis model, a design model and an implementation model.

Since the pictograms are aimed at facilitating modeling by being closer to human language than typical modeling artefacts, they are primarily used in high-level models. Regarding the MDA method, pictogrammic expressions are more widely used for CIM than for PIM and PSM:

- CIM: “A *computation independent model* is a view of a system from the computation independent viewpoint. A CIM does not show details of the structure of systems. A CIM is sometimes called a domain model and a vocabulary that is familiar to the practitioners of the domain in question is used in its specification.” (Miller and Mukerji 2003)
- PIM: “A *platform independent model* is a view of a system from the platform independent viewpoint. A PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type.” (Miller and Mukerji 2003)
- PSM: “A *platform specific model* is a view of a system from the platform specific viewpoint. A PSM combines the specifications in the PIM


Level of abstraction
CIM
BUILDING
Stories
Address !
Building value
OGC PIM
BUILDING
Stories :integer
Address : address
Building value :integer
Geom 0,1 : GM_Point
Geom 0,1 : GM_Surface
<<DATA TYPE>>
ADDRESS
Civic number
Steet name
PSM ESRI SHAPEFILE
BUILDING_A
ObjectID : esriFieldTypeOID
Stories : esriFieldTypeInteger
CivicNumber : esriFieldTypeInteger
StreetName : esriFieldTypeString
BuildingValue : esriFieldTypeInteger
ValueBeginDate : esriFieldTypeDate
ValueEndDate : esriFieldTypeDate
BuildingDate : esriFieldTypeDate
BuildingEndDate : esriFieldTypeDate
Shape : esriFieldTypeGeometry
BUILDING_P
ObjectID : esriFieldTypeOID
Stories : esriFieldTypeInteger
CivicNumber : esriFieldTypeInteger
StreetName : esriFieldTypeString
BuildingValue : esriFieldTypeInteger
ValueBeginDate : esriFieldTypeDate
ValueEndDate : esriFieldTypeDate
BuildingDate : esriFieldTypeDate
BuildingEndDate : esriFieldTypeDate
Shape : esriFieldTypeGeometry
Development phase

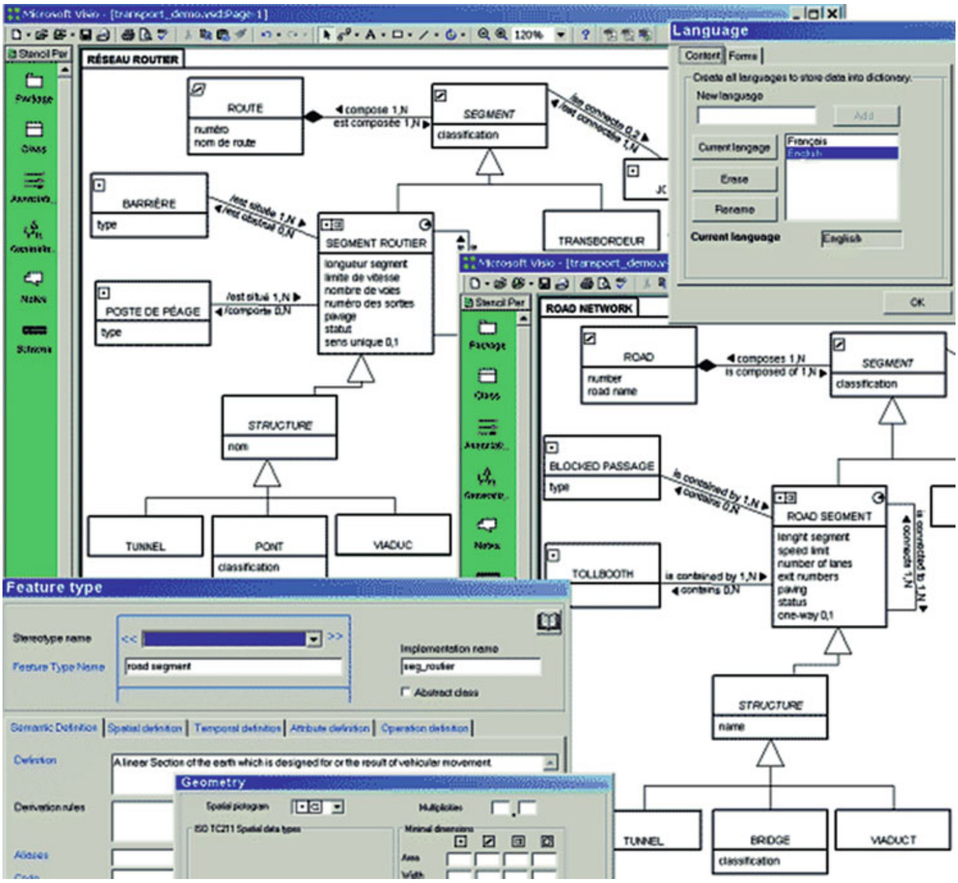
Modeling with Pictogrammic Languages, Fig. 6 Examples of CIM, PIM and PSM levels of abstraction of the MDA method for a same application, where the information encapsulated in the higher levels using pictograms is expanded in the lower levels

> with the details that specify how that system uses a particular type of platform." (Miller and Mukerji 2003)

Furthermore, since pictograms are not tied to a specific natural language, they facilitate the translation of database models. For example, in Canada, several schemas and repositories are available in English and French. Figure 7 shows such French and English schemas that are synchronized thru the same repository and pictograms. The use of formal ISO-19110 labels (in blue) further facilitates communication while the use of pictograms facilitates automatic GIS code generation and bilingual reporting.

At the CIM level, pictogrammic expressions are intuitive and independent of domain ontologies and technology-oriented standards. No technology artefacts nor standardization elements must appear unless they are useful and intuitive. When the CIM is well defined, it can be translated and enriched to produce lower-level models semi-automatically. Then, technology-oriented artefacts and standard-based elements replace the pictogrammic expressions. For example, in Fig. 6, the CIM evolves in a PIM where the geometry is expressed according to ISO/OGC. Then, the PSM shows the structure of two shapefiles needed to implement Building Points and Building Areas.

In addition to hiding the technical complexities of GIS and Universal server database engines, using pictogrammic expressions also hides the intricacies of international standards such as ISO/TC-211 and OGC. For example, ISO jargon doesn't express directly all possible geometries (e.g. alternate and facultative geometries) and they are not cognitively compatible with clients' conceptual view who assumes a topologically consistent world (e.g. GMPoint vs. TPNode,



Create all languages to store data into dictionary.

New language

Français
English

Current language English

Stereotype name

Feature Type Name road segment

Implementation name seg_routier

□ Abstract class

Definition A linear Section of the earth which is designed for or the result of vehicular movement

Derivation rules

Aliases

Code

Modeling with Pictogrammic Languages, Fig. 7 Example of common pictograms in a French and an English CIM synchronized for a same spatiotemporal database using Perceptory multi-standard and multi-language capabilities

GMCurve vs. TPEdge, GMSurface vs. TPFace, Aggregate vs. Multi).

Using Pictogrammic Expressions to Define Spatial Integrity Constraints

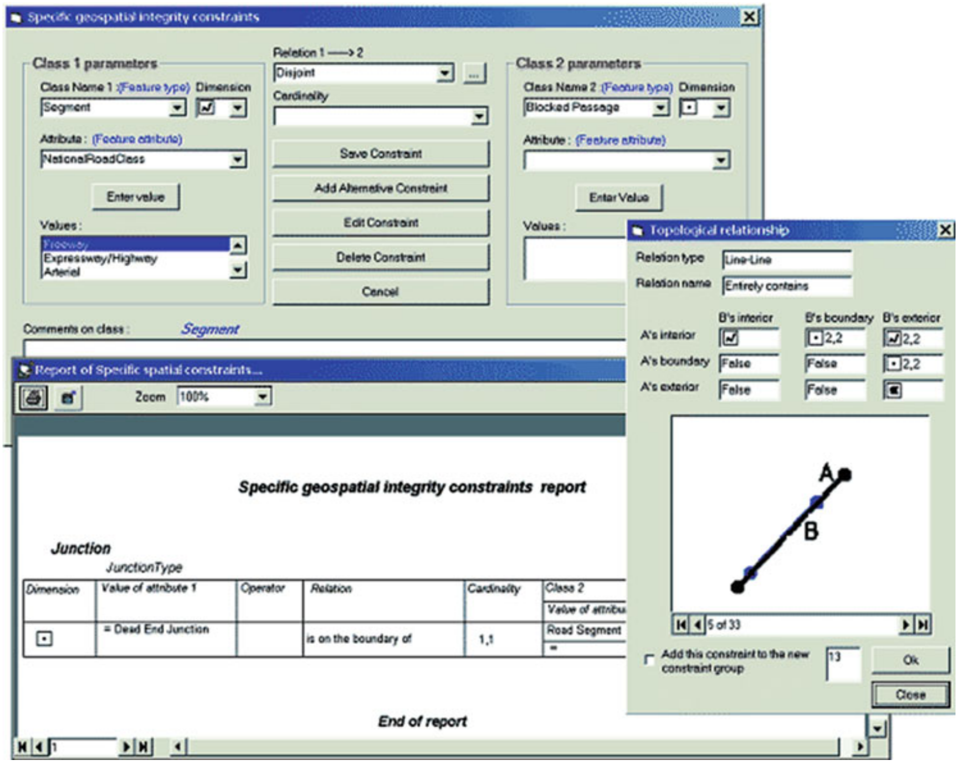
Spatial integrity constraints can also be defined efficiently with pictogrammic expressions. For example, in Fig. 8, the upper window shows a user interface for the definition of spatial integrity constraints between two object classes, with or without considerations to specific attribute values. The lower window shows a report showing the defined spatial integrity constraints. The last window shows an example of using pictogrammic expressions in a 3 × 3 e-relate matrix.

Additional Usages of Pictogrammic Expressions: Software User Interfaces, Reports and Semantic Proximity Analysis

Pictogrammic expressions are regularly used in a text to express the spatiality and temporality of objects. They have been used in reports, data dictionaries and data acquisition specifications. They were also used for semantic proximity analysis (Brodeur et al. 2003) and integrated in a commercial package (JMap SOLAP, Fig. 9).

Future Directions

Over the last two decades, different pictogrammic languages have emerged to improve the efficiency



Specific geospatial integrity constraints

Class 1 parameters
Class Name 1 :(Feature type): Segment
Dimension
Attribute : (Feature attribute): NationalRoadClass
Values : Freeway, Expressway/Highway, Arterial

Relation 1 ——> 2: Disjoint
Cardinality

Class 2 parameters
Class Name 2 :(Feature type): Blocked Passage
Dimension
Attribute : (Feature attribute)
Values :

Comments on class : Segment

Specific geospatial integrity constraints report

Junction
JunctionType

Dimension	Value of attribute 1	Operator	Relation	Cardinality	Class 2 Value of attribu
⊡	= Dead End Junction		is on the boundary of	1,1	Road Segment =

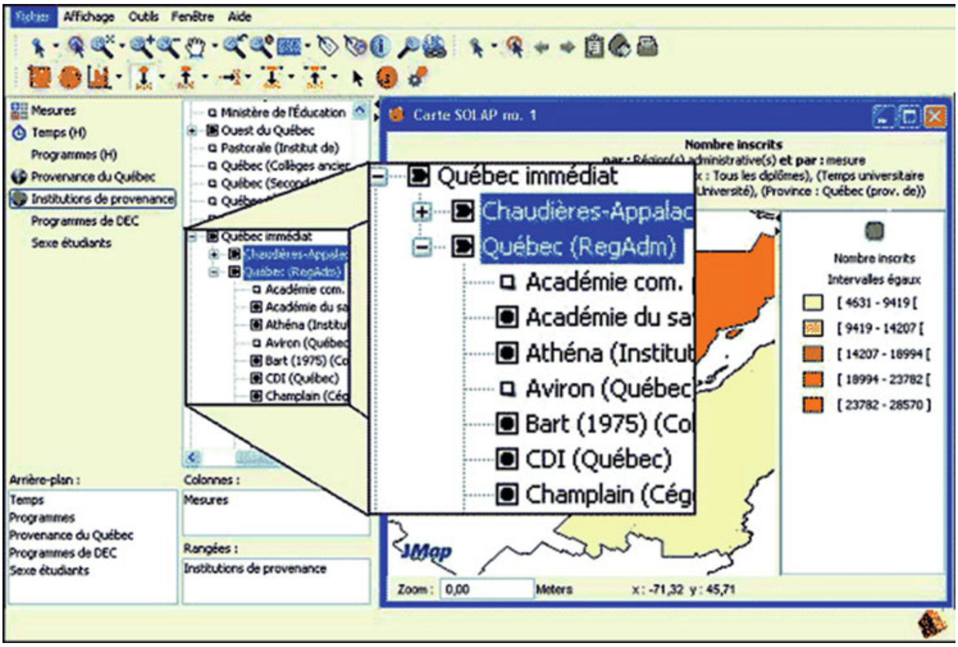
End of report

Topological relationship
Relation type: Line-Line
Relation name: Entirely contains

	B's interior	B's boundary	B's exterior
A's interior	☑	⊡ 2,2	☑ 2,2
A's boundary	False	False	⊡ 2,2
A's exterior	False	False	▣

□ Add this constraint to the new constraint group: 13

Modeling with Pictogrammic Languages, Fig. 8 Examples of pictogrammic expressions to define topological constraints between two object classes (*upper left*), to print them in a report (*lower left*) and to describe them in an extended ISO e-related 3×3 matrix



Ministère de l'Éducation
Ouest du Québec
Pastorale (Institut de)
Québec (Collèges ancien
Québec (Second
Québec immédiat
Chaudières-Appalac
Québec (RegAdm)
Académie com.
Académie du sa
Athéna (Institut
Aviron (Québec
Bart (1975) (Co
CDI (Québec)
Champlain (Cég

Nombre inscrits
par : Région(s) administrative(s) et par : mesure
x : Tous les diplômes), (Temps universitaire
Université), (Province : Québec (prov. de))

Nombre inscrits
Intervalles égaux

□	[4631 - 9419 [
▨	[9419 - 14207 [
■	[14207 - 18994 [
■	[18994 - 23782 [
■	[23782 - 28570]

Arrière-plan : Temps, Programmes, Provenance du Québec, Programmes de DEC, Sexe étudiants
Colonnes : Mesures
Rangées : Institutions de provenance

Zoom : 0,00 Meters x : -71,32 y : 45,71

Modeling with Pictogrammic Languages, Fig. 9 JMap SOLAP interface using pictogrammic expressions of systems analysts and to improve the quality of spatial database design. The language presented in this chapter was the first such language and has become the most widely used one, not only within Perceptory but also in other CASE tools and in diverse applications. It uses a downloadable font (http://sirs.scg.ulaval.ca/YvanBedard/english/others.asp). Such languages will likely evolve in two major directions. First, they will accommodate the most recent spatial database trends, that is spatial datacube structures for data warehousing and SOLAP applications. Second, as they can be translated into ISO and OGC primitives (Brodeur et al. 2000), official adoption of such a language should be put forward to improve interoperability between spatial application database schemas, between ontologies and between other documents.

Cross-References

- ▸ Modeling and Multiple Perceptions
- ▸ Modeling with Enriched Model-Driven Architecture
- ▸ Modeling with a UML Profile
- ▸ Modeling with ISO 191xx Standards
- ▸ Spatiotemporal Database Modeling with an Extended Entity-Relationship Model

References

Bédard Y (1999) Visual modeling of spatial database, towards spatial PVL and UML. Geomatica 53(2): 169–185

Bédard Y, Larrivée S (1992) Développement des systèmes d'information à référence spatiale: vers l'utilisation d'ateliers de génie logiciel. CISM J ACSGC Can Inst Geomat Can 46(4):423–433

Bédard Y, Paquette F (1989) Extending entity/relationship formalism for spatial information systems. In: AUTO-CARTO 9, 9th international symposium on automated cartography, ASPRS-ACSM, Baltimore, 2–7 Apr 1989, pp 818–827

Bédard Y, Pageau J, Caron C (1992) Spatial data modeling: the Modul-R formalism and CASE technology. In: ISPRS Symposium, Washington DC, 1–14 Aug 1992

Bédard Y, Caron C, Maamar Z, Moulin B, Vallière D (1996) Adapting data model for the design of spatiotemporal database. Comput Environ Urban Syst 20(l):19–41

Bédard Y, Proulx, M-J, Larrivée S, Bernier E (2002) Modeling multiple representations into spatial data warehouses: a UML-based approach. ISPRS WG IV/3, Ottawa, 8–12 July 2002, p 7

Bédard Y, Larrivée S, Proulx M-J, Nadeau M (2004) Modeling geospatial databases with plug-ins for visual languages: a pragmatic approach and the impacts of 16 years of research and experimentations on perceptory. In: Wang S et al (eds) Conceptual modeling for advanced application domains. Lecture notes in computer science, vol 3289. Springer, Berlin/Heidelberg, pp 17–30

Brodeur J, Bédard Y, Proulx M-J (2000) Modeling geospatial application database using UML-based repositories aligned with international standards in geomatics. In: ACM-GIS, Washington DC, 10–11 Nov 2000, pp 36–46

Brodeur J, Bédard Y, Edwards G, Moulin B (2003) Revisiting the concept of geospatial data interoperability within the scope of human communication processes. Trans GIS 7(2):243–265

Filho JL, Sodre VDF, Daltio J, Rodrigues Junior MF, Vilela V (2004) A CASE tool for geographic database design supporting analysis patterns. Lecture notes in computer science, vol 3289. Springer, Berlin/Heidelberg, pp 43–54

Fowler M (2004) UML 2.0. Campus Press, Pearson Education, Paris, p 165

Frankel DS, Harmon P, Mukerji J, Odell J, Owen M, Rivitt P, Rosen M, Soley R (2003) The Zachman framework and the OMG's model driven architecture. Business Process Trends, Whitepaper

Miller J, Mukerji J (2003) MDA guide version 1.0. In: Miller J, Mukerji J (eds) OMG document: omg/2003-05-01. http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf

Miralles A (2006) Ingénierie des modèles pour les applications environnementales. Ph.D. thesis, Université des Sciences et Techniques du Languedoc, Département d'informatique, Montpellier, p 338

Naiburg EJ, Maksimchuk RA (2001) UML for database design. Addison-Wesley, Boston, p 300

Pantazis D, Donnay JP (1996) La conception SIG: méthode et formalisme. Hermès, Paris, p 343

Parent C, Spaccapietra S, Zimányi E (2006) Conceptual modeling for traditional and SpatioTemporal applications: the MADS approach. Springer, Berlin/Heidelberg, p 466

Sowa JF, Zachman JA (1992) Extending and formalizing the framework for information systems architecture. IBM Syst J 31(3). IBM Publication G321-5488

Tryfona N, Price R, Jensen CS (2003) Conceptual models for SpatioTemporal applications. In: Spatiotemporal databases: the CHOROCHRONOS approach. Lecture notes in computer science, Chapter 3. Springer, Berlin/Heidelberg, vol 2520

Modifiable Areal Unit Problem

▸ Error Propagation in Spatial Prediction

Monitoring

▸ Data Collection, Reliable Real-Time

Monte Carlo Simulation

▸ Uncertain Environmental Variables in GIS

Moran Coefficient

▸ Moran's I

Moran Eigenvector Spatial Filtering

▸ Eigenvector Spatial Filtering and Spatial Autoregression

Moran's *I*

Xiaobo Zhou and Henry Lin
Department of Crop and Soil Sciences, The Pennsylvania State University, University Park, PA, USA

Synonyms

Moran coefficient; Moran's index

Definition

Moran's I, based on cross-products, measures value association and is calculated for n observations on a variable x at locations i, j as

$$I = \frac{\sum_i \sum_{j \neq i} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{S^2 \sum_i \sum_{j \neq i} w_{ij}} .$$

Where x_i denotes the observed value at location i, $\bar{x}$ is the mean of the x variable over the n locations,

$$S^2 = \frac{1}{n} \sum_i (x_i - \bar{x})^2 ,$$

and w_{ij} is the element of the spatial weights matrix for locations i and j, defined as 1 if location i is contiguous to location j and 0 otherwise. Other more complicated definitions of spatial weights matrices allow for the computation of the Moran's I at various levels of proximity or distance.

Main Text

Moran's I is one of the oldest indicators of spatial autocorrelation and is still a widely accepted measure for determining spatial autocorrelation. It is used to estimate the strength of interdependence between observations of the variable of interest as a function of the distance by comparing the value of x_i at location i with the value x_j at all other locations ($j \neq i$). Moran's I varies from -1 to 1. Positive signage represents positive spatial autocorrelation, while the negative signage represents negative spatial autocorrelation. The Moran's I will approach zero for a large sample size when the variable values are randomly distributed and independent in space.

Cross-References

▸ Autocorrelation, Spatial

Moran's Index

▶ Moran's I

Motion Patterns

▶ Movement Patterns in Spatio-Temporal Data

Motion Tracking

▶ Moving Object Uncertainty

Moved Planning Process

▶ Participatory Planning and GIS

Movement

▶ Temporal GIS and Applications

Movement Patterns in Spatio-Temporal Data

Joachim Gudmundsson[1], Patrick Laube[2], and Thomas Wolle[3]
[1]NICTA, Sydney, NSW, Australia
[2]Department of Geomatics, University of Melbourne, Melbourne, VIC, Australia
[3]Sydney, Australia

Synonyms

Association Rules, Spatiotemporal; Converging; Collocation, Spatiotemporal; Exploratory data analysis; Flocking; Indexing, native space; Indexing, parametric space; Indexing trajectories; Motion patterns; Pattern, encounter Pattern, flock; Pattern, leadership; Pattern, moving cluster; Pattern, periodic; R-tree, multi-version; TPR-trees; Trajectory patterns;

Definition

Spatio-temporal data is any information relating space and time. This entry specifically considers data involving point objects moving over time. The terms *entity* and *trajectory* will refer to such a point object and the representation of its movement, respectively. *Movement patterns* in such data refer to (salient) events and episodes expressed by a set of entities.

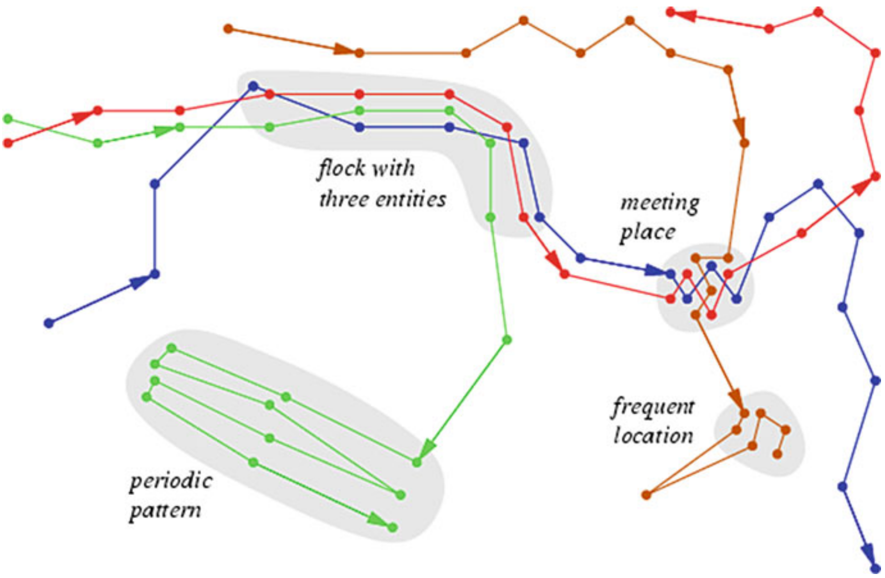
In the case of moving animals, movement patterns can be viewed as the spatio-temporal expression of behaviors, as for example in flocking sheep or birds assembling for the seasonal migration. In a transportation context, a movement pattern could be a traffic jam.

Only formalized patterns are detectable by algorithms. Hence, movement patterns are modeled as any arrangement of subtrajectories that can be sufficiently defined and formalized, see for example the patterns illustrated in Fig. 1. A pattern usually involves a certain number of entities. Furthermore a pattern starts and ends at certain times (temporal footprint), and it might be restricted to a subset of space (spatial footprint).

Historical Background

The analysis of movement patterns in Spatio-temporal data is for two main reasons a relatively young and little developed research field. First, emerging from static cartography, geographical information science and theory struggled for a long time with the admittedly substantial challenges of handling dynamics. For many years, occasional changes in a cadastral map were challenging enough, not to mention the constant change of location as is needed for modeling movement.

Second, only in recent years has the technological advancement in tracking technology reached a level that allowed the seamless tracking of individuals needed for the analysis of movement patterns. For many years, the tracking of movement entities has been a very cumbersome



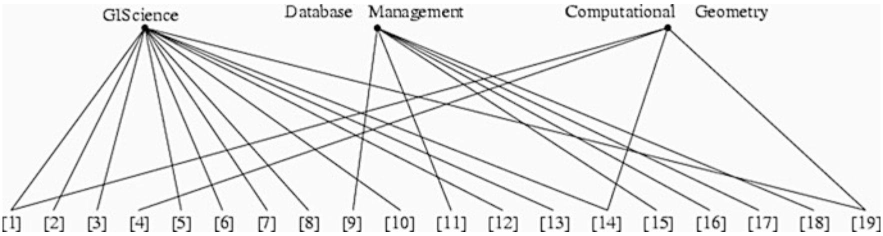
Movement Patterns in Spatio-Temporal Data, Fig. 1 Illustrating the trajectories of four entities moving over 20 time steps. The following patterns are highlighted: a *flock* of three entities over five time-steps, a *periodic pattern* where an entity shows the same Spatiotemporal pattern with some periodicity, a *meeting place* where three entities meet for four time steps, and finally, a *frequently visited location* which is a region where a single entity spends a lot of time

and costly undertaking. Hence, movement patterns could only be addressed for single individuals or very small groups. Hägerstrand's time geography (Hägerstrand 1970) may serve as a starting point of a whole branch of geographical information science representing individual trajectories in 3D. The two spatial dimensions combined with an orthogonal temporal axis proved to be a very powerful concept for exploring various kinds of spatio-temporal relationships, including movement patterns.

With GPS and various other tracking technologies movement pattern research entered a new era, stepping from 'thread trailing' and 'mark and recapture' approaches to low cost, almost continuous capture of individual trajectories with possibly sub-second sampling rates. Within a few years the situation completely reversed from a notorious data deficit to almost a data overkill, with a lack of suited analytical concepts coping with the sudden surge of movement data. Consequently, the huge potential of analyzing movement in spatio-temporal data has recently attracted the interest of many research fields, both in theory and application, as is outlined in the next two sections.

Scientific Fundamentals

Assume that the entities in Fig. 1 are sheep on a pasture and that they are observed by a geographer, a database expert and a computational geometer. Even though all three experts see the very same sheep, they may all perceive totally different things. The geographer might interpolate a sheep density surface of the pasture. For the database expert in contrast, each sheep may represent a leaf in a dynamic tree optimized for fast queries. Finally, the computational geometer might triangulate the sheep locations in order to detect a flocking pattern. Even though the sheep will not care, their grazing challenges various research fields handling spatio-temporal data. The following overview bundles the different perspectives addressing movement patterns into the three sections exploration, indexing and data mining.



Movement Patterns in Spatio-Temporal Data, Fig. 2 Access guide to the references in the recommended reading section below

See Fig. 2 for a comprehensible access guide to recommended reading.

GIScience: Exploratory Data Analysis and Visualization

In GIScience the term 'pattern' is used in various contexts and meanings when addressing movement. However, as a common denominator, movement patterns are generally conceptualized as salient movement events or episodes in the geospatial representation of moving entities. Given GIScience's legacy in cartography, it is not surprising that movement patterns are often addressed by a combination of geovisualisation and exploration. Exploratory analysis approaches combine the speed and patience of computers with the excellent capability of humans to detect the expected and discover the unexpected given an appropriate graphical representation.

Salient movement patterns may emerge from (i) two-dimensional maps of fixes aligned in trajectories, (ii) movie-like animated maps or even (iii) three-dimensional representations of movement, if time is used as a third, orthogonal axis.

(i) Basic movement patterns are obvious from simple plotting of movement trajectories on a two-dimensional map. Trajectories bundled in narrow, directed bottlenecks represent often used corridors. Less focused trajectory footprints represent more arbitrary movement, such as in grazing animals or visitors at a sports event strolling around a stadium. The application of GUS analysis tools on points and lines representing moving entities has proven to be a very effective approach. For example, GIS tools for generalization, interpolation and surface generation may be applied to support the detection of movement patterns in trajectory data. Brillinger et al. (2004) use a regularly sampled vector field to illustrate the overall picture of animals moving in their habitat, with each vector coding in orientation and size for mean azimuth and mean speed at that very location. Dykes and Mountain (2003) use a continuous density surface and a 'spotlight' metaphor for the detection of activity patterns. Again, common GIS tools such as algorithms initially designed for the analysis of digital terrain models can easily be adopted for the search for salient movement patterns, for instance to identify 'peaks' of frequent visitation and 'ridges' of busy corridors (Dykes and Mountain 2003).

(ii) Animation is suited to uncover specific movement behaviors of individuals and groups. Animating moving entities with a constant moving time window in the so-called dynamic view uncovers speed patterns of individuals (Andrienko and Andrienko 2003; Dykes and Mountain 2003). Flocking or converging are more complex patterns of coordination in groups. Such group patterns are very striking when animating even large numbers or individuals in a movie-like animation.

(iii) The extension of a two-dimensional map with a third orthogonal time axis produces a very powerful tool for uncovering movement patterns. Such ideas go back to Hägerstrand's time geography (Hägerstrand 1970) and have often been adopted in present day geocomputation (Kwan 2000). In the specific geometry in such a three-dimensional space-time aquarium episodes of immobility and certain speed behaviors produce distinctive patterns of vertical and inclined time lines, respectively. Furthermore, patterns of spatio-temporal collocation can be identified from vertical bottleneck structures in sets of time lines (Kwan 2000).

Indexing Spatio-Temporal Trajectories

In the database community considerable research has been focusing on spatial and temporal databases. Research in the spatio-temporal area in many ways started with the dissertations by Lorentzos (1988) in 1988 and Langran (1999) in 1989. Not surprisingly research has mainly focused on indexing databases so that basic queries concerning the data can be answered efficiently. The most common queries considered in the literature are variants of nearest neighbor queries and range searching queries. For example:

- Spatiotemporal range query, e.g. 'Report all entities that visited region S during the time interval $[t_1, t_2]$.'
- Spatial nearest neighbors given a time interval, e.g. 'Report the entity closest to point p at time t.'
- Temporal nearest neighbors given a spatial region, e.g. 'Report the first entity visiting region S.'

In general one can classify indexing methods used for spatio-temporal data into Parametric Space Indexing methods (PSI) and Native Space Indexing methods (NSI). The PSI method uses the parametric space defined by the movement parameters, and is an efficient approach especially for predictive queries. A typical approach, described by Šaltenis et al. (2000) is to represent movement defined by its velocity and projected location along each spatial dimension at a global time reference. The parametric space is then indexed by a new index structure referred to as the TPR-tree (Time Parametrized R-tree). The TPR-tree is a balanced, multi-way tree with the structure of an R-tree. Entries in leaf nodes are pairs of the position of a moving point and a pointer to the moving point, and entries in internal nodes are pairs of a pointer to a subtree and a rectangle that bounds the positions of all moving points or other bounding rectangles in that subtree. The position of a moving point is represented by a reference position and a corresponding velocity vector. To bound a group of d-dimensional moving points, d-dimensional bounding rectangles are used that are also time parametrized, i.e. their coordinates are functions of time. A time-parametrized bounding rectangle bounds all enclosed points or rectangles at all times not earlier than the current time. The search algorithm for a range query also performs computation on the native space by checking the overlap between the range of the query and the trapezoid representation of the node.

The NSI methods represent movement in d dimensions as a sequence of line segments in $d + 1$ dimensions, using time as an additional dimension, see for example the work by Hadjieleftheriou et al. (2006). A common approach is to use a multi-dimensional spatial access method like the R-tree. An R-tree would approximate the whole spatio-temporal evolution of an entity with one Minimum Bounding Region (MBR) that tightly encloses all the locations occupied by the entity during its lifetime. An improvement for indexing movement trajectories is to use a multi version index, like the Multi Version R-tree (MVR-tree), also known as a persistent R-tree. This index stores all the past states of the data evolution and allows updates to the most recent state. The MVR-tree divides long-lived entities into smaller intervals by introducing a number of entity copies. A query is directed to the exact state acquired by the structure at the time that the query refers to; hence, the cost of answering the query is proportional to the number of entities that the structure contained at that time.

Algorithms and Data Mining

In the previous section different indexing approaches were discussed. This section will focus on mining trajectories for spatio-temporal patterns. This has mainly been done using algorithmic or data mining approaches.

The most popular tools used in the data mining community for spatio-temporal problems has been association rule mining (ARM) and various types of clustering. Association rule mining seeks to discover associations among transactions within relational databases. An association rule is of the form $X \Rightarrow Y$ where X (antecedents) and Y (consequents) are disjoint conjunctions of attribute-value pairs. ARM uses the concept of *confidence* and *support*. The confidence of the rule is the conditional probability of Y given X, and the support of the rule is the prior probability of X and Y.

The probability is usually the observed frequency in the data set. Now the ARM problem can be stated as follows. Given a database of transactions, a minimal confidence threshold and a minimal support threshold, find all association rules whose confidence and support are above the corresponding thresholds.

Verhein and Chawla (2006) defined spatio-temporal association rules (STARs) that describe how entities move between regions over time. They assume that space is partitioned into regions, which may be of any size and shape. The aim is to find interesting regions and rules that predict how entities will move through the regions. A region is interesting when a large number of entities leaves (sink), a large number of entities enters (source) or a large number of entities enters and leaves (thoroughfare).

A STAR $(r_i, T_1, q) \Rightarrow (r_j, T_2)$ denotes a rule where entities in a region r_i satisfying condition q during time interval T_1 will appear in region r_j during time interval T_2. The support of a rule δ is the number, or ratio, of entities that follow the rule. The *spatial* support takes the size of the involved regions into consideration. That is, a rule with support s involving a small region will have a larger spatial support than a rule with support s involving a larger region. Finally, the confidence of a rule δ is the conditional probability that the consequent is true given that the antecedent is true. By traversing all the trajectories all possible movements between regions can be modeled as a rule, with a spatial support and confidence. The rules are then combined into longer time intervals and more complicated movement patterns.

Some of the most interesting spatio-temporal patterns are periodic patterns, e.g. yearly migration patterns or daily commuting patterns. Mamoulis et al. (2004) considered the special case when the period is given in advance. They partition space into a set of regions which allows them to define a pattern P as a τ-length sequence of the form $r_0, r_1, \ldots, r_{\tau-1}$, where r_i is a spatial region or the special character *, indicating the whole spatial universe. If the entity follows the pattern enough times, the pattern is said to be *frequent*. However, this definition imposes no control over the density of the regions, i.e. if the regions are too large then the pattern may always be frequent. Therefore an additional constraint is added, namely that the points of each subtrajectory should form a cluster inside the spatial region.

Kalnis et al. (2005) define and compute moving clusters where entities might leave and join during the existence of a moving cluster. For each fixed discrete time-step t_i they use standard clustering algorithms to find clusters with a minimum number of entities and a minimum density. Then they compare any cluster c found for t_i with any (moving) cluster c' found for time-step t_{i-1}. If c and c' have enough entities in common, which is formally specified by a threshold value, then c' can be extended by c, which results in a *moving cluster*. They propose several ideas to increase the speed of their method, e.g. by avoiding redundant cluster comparisons, or approximating moving clusters instead of giving exact solutions, and they experimentally analyze their performance.

In 2004 Laube et al. (2004) defined a collection of spatio-temporal patterns based on direction of movement and location, e.g. flock, leadership, convergence and encounter, and they gave algorithms to compute them efficiently. As a result there were several subsequent articles studying the discovery of these patterns. Benkert et al. (2006) modified the original definition of a flock to be a set of entities moving close together during a time interval. Note that in this definition the entities involved in the flock must be the same during the whole time interval, in contrast to the moving cluster definition by Kalnis et al. (2005). Benkert et al. (2006) observed that a flock of m entities moving together during k time steps corresponds to a cluster of size m in $2k$ dimensional space. Thus the problem can be restated as clustering in high dimensional space. To handle high dimensional space one can use well-known dimensionality reduction techniques. There are several decision versions of the problem that have been shown to be NP-hard, for example deciding if there exists a flock of a certain size, or of a certain duration. The special case when the flock is stationary is often called a *meeting* pattern.

Andersson et al. (2007) gave a more generic definition of the pattern *leadership* and discussed how such leadership patterns can be computed from a group of moving entities. The proposed definition is based on behavioral patterns discussed in the behavioral ecology literature. The idea is to define a leader as an entity that (1) does not follow anyone else, (2) is followed by a set of entities and (3) this behavior should continue for a duration of time. Given these rules all leadership patterns can be efficiently computed.

Be it exploratory analysis approaches, indexing techniques or data mining algorithms, all effort put in theory ultimately leads to more advanced ways of inferring high level process knowledge from low level tracking data. The following section will illustrate a wide range of fields where such fundamentals underlie various powerful applications.

Key Applications

Animal Behavior

The observation of behavioral patterns is crucial to animal behavior science. So far, individual and group patterns are rather directly observed than derived from tracking data. However, there are more and more projects that collect animal movement by equipping them with GPS-GSM collars. For instance, since 2003 the positions of 25 elks in Sweden are obtained every 30 min. Other researchers attached small GPS loggers to racing pigeons and tracked their positions every second during a pigeon's journey. It is even possible to track the positions of insects, e.g. butterflies or bees, however most of the times non-GPS based technologies are used that allow for very small and light sensors or transponders. Analyzing movement patterns of animals can help to understand their behavior in many different aspects. Scientists can learn about places that are popular for individual animals, or spots that are frequented by many animals. It is possible to investigate social interactions, ultimately revealing the social structure within a group of animals. A major focus lies on the investigation of leading and following behavior in socially interacting animals, such as in a flock of sheep or a pack of wolves (Dumont et al. 2005). On a larger scale, animal movement data reflects very well the seasonal or permanent migration behavior. In the animation industry, software agents implement movement patterns in order to realistically mimic the behavior of animal groups. Most prominent is the flocking model implemented in NetLogo which mimics the flocking of birds (Wilensky).

Human Movement

Movement data of people can be collected and used in several ways. For instance, using mobile phones that communicate with a base station is one way to gather data about the approximate locations of people. Traffic-monitoring devices such as cameras can deliver data on the movement of vehicles. With the technological advancement of mobile and position aware devices, one could expect that tracking data will be increasingly collectable. Although tracking data of people might be available in principle, ethical and privacy aspects need to be taken into consideration before gathering and using this data (Dobson and Fisher 2003). Nonetheless, if the data is available, it could be used for urban planning, e.g. to plan where to build new roads or where to extend public transport.

The detection of movement patterns can furthermore be used to optimize the design of location-based-services (LBS). The services offered to a moving user could not only be dependent on the actual position, but also on the estimated current activity, which may be derived from a detected movement pattern.

Traffic Management

Movement patterns are used for traffic management in order to detect undesirable or even dangerous constellations of moving entities, such as traffic jams or airplane course conflicts. Traffic management applications may require basic Moving Object Database queries, but also more sophisticated movement patterns involving not just location but also speed, movement direction and other activity parameters.

Surveillance and Security

Surveillance and intelligence services might have access to more detailed data sets capturing the movement of people, e.g. coordinates from mobile phones or credit card usage, video surveillance camera footage or maybe even GPS data. Apart from analyzing the movement data of a suspect to help prevent further crime, it is an important task to analyze the entire data set to identify suspicious behavior in the first place. This leads to define 'normal behavior' and then search the data for any outliers, i.e. entities that do not show normal behavior. Some specific activities and the corresponding movement patterns of the involved moving entities express predefined signatures that can be automatically detected in spatio-temporal or footage data. One example is that fishing boats in the sea around Australia have to report their location in fixed intervals. This is important for the coast guards in case of an emergency, but the data can also be used to identify illegal fishing in certain areas. Another example is that a car thief is expected to move in a very characteristic and hence detectable way across a surveilled car park. Movement patterns have furthermore attracted huge interests in the field of spatial intelligence and disaster management. Batty et al. (2003) investigated local pedestrian movement in the context of disaster evacuation where movement patterns such as congestion or crowding are key safety issues.

Military and Battlefield

The digital battlefield is an important application of moving object databases. Whereas real-time location data of friendly troops is easily accessible, the enemy's location may be obtained from reconnaissance planes with only little time lag. Moving object databases not only allow the dynamic updating of location and status of tanks, airplanes and soldiers, but also answering spatio-temporal queries and detecting complex movement patterns. Digital battlefield applications answer spatio-temporal range queries like 'Report all friendly tanks that are currently in region S.' A more complex movement pattern in a digital battlefield context would be the identification of the convergence area where the enemy is currently concentrating his troops.

Sports Scene Analysis

Advancements in many different areas in technology are also influencing professional sports. For example, some of the major tennis tournaments provide three-dimensional reconstructions of every single point played, tracking the players and the balls. It is furthermore known that, e.g. football coaches routinely analyze match video archives to learn about an opponents behaviors and strategies. Making use of tracking technology, the movement of the players and the ball can be described by 23 trajectories over the length of the match. Researchers were able to develop a model that is based on the interactions between the players and the ball. This model can be used to quantitatively express the performance of players, and more general, it might lead to an improved overall strategy. Finally, real-time tracking systems are developed that keep track of both players and the ball in order to assist the referee with the detection of the well-defined but nevertheless hard to perceive offside pattern.

Movement in Abstract Spaces

In contrast to tracking and analyzing the movement of animals and people on the surface of the earth, it is also possible to obtain and analyze spatio-temporal data in abstract spaces also in higher dimensions. Every scatter plot that constantly updates the changes in the x and y values, produces individual trajectories open for movement analysis. Two stock exchange series plotted against each other could build such a dynamic scatter-plot. As another example, basic ideological conflicts can be used to construct abstract ideological spaces. Performing factor analysis on referendum data, researchers hypothesized a structure of mentality consisting of dimensions such as ‘political left vs. political right’ or ‘liberal vs. conservative’. Whole districts or even individuals such as members of parliament could now be localized and re-localized in such ideological space depending on their voting behavior and its change over time, respectively. Movement in such a space represents the change of opinions and analyzing this can lead to more insight and understanding of human psychology and politics.

Future Directions

For simplicity reasons, theory and application of movement patterns in spatio-temporal data focused so far largely on moving point objects. However, many processes can only be modeled as dynamics in fields or in their discredited counterparts that is dynamic polygons. When monitoring a hurricane threatening urban areas, the tracking of its eye alone may not provide sufficient information, but additional tracking of its changing perimeter will be required. The consideration of both location and change of polygonal objects raises the conceptualization and detection of movement patterns to a higher level of complexity, which has only rarely been addressed so far.

For the many fields interested in movement, the overall challenge lies in relating movement patterns with the underlying geography, in order to understand where, when and ultimately why the entities move the way they do. Grazing sheep, for example, may perform a certain movement pattern only when they are on a certain vegetation type. Homing pigeons may show certain flight patterns only when close to a salient landscape feature such as a river or a highway. And, the movement patterns expressed by a tracked vehicle will obviously be very dependent on the environment the vehicle is moving in, be it in a car park, in a suburb or on a highway. Thus, patterns have to be conceptualized that allow linking of the movement with the embedding environment.

Cross-References

▸ Indexing, Query and Velocity-Constrained
▸ Privacy Threats in Location-Based Services

References

Andersson M, Gudmundsson J, Laube P, Wolle T (2007) Reporting leadership patterns among trajectories. In: Proceedings of the 2007 ACM symposium on applied computing. ACM, New York, pp 3–7

Andrienko NV, Andrienko GL (2003) Interactive maps for visual data exploration. Int J Geogr Inf Sci 13(4): 355–374

Batty M, Desyllas J, Duxbury E (2003) The discrete dynamics of small-scale spatial events: agent-based models of mobility in carnivals and street parades. Int J Geogr Inf Sci 17(7):673–697

Benkert M, Gudmundsson J, Hübner F, Wolle T (2006) Reporting flock patterns. In: Proceedings of the 14th European symposium on algorithms. Lecture notes in computer science, vol 4168. Springer, Berlin/Heidelberg, pp 660–671

Brillinger DR, Preisler HK, Ager AA, Kie JG (2004) An exploratory data analysis (EDA) of the paths of moving animals. J Stat Plan Inf 122(2):43–63

Dobson JE, Fisher PF (2003) Geoslavery. IEEE Technol Soc Mag 22(1):47–52

Dumont B, Boissy A, Achard C, Sibbald AM, Erhard HW (2005) Consistency of animal order in spontaneous group movements allows the measurement of leadership in a group of grazing heifers. Appl Anim Behav Sci 95(1–2):55–66

Dykes JA, Mountain DM (2003) Seeking structure in records of spatiotemporal behaviour: visualization issues, efforts and application. Comput Stat Data Anal 43(4):581–603

Hadjieleftheriou M, Kollios G, Tsotras VJ, Gunopulos D (2006) Indexing spatiotemporal archives. VLDB J 15(2):143–164

Hägerstrand T (1970) What about people in regional science. Pap Region Sci Assoc 24:7–21

Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatiotemporal data. In: Medeiros CB, Egenhofer MJ, Bertino E (eds) Proceedings of the 9th international symposium on advances in spatial and temporal databases. Lecture notes in computer science, vol 3633. Springer, Berlin/Heidelberg, pp 364–381

Kwan MP (2000) Interactive geovisualization of activitytravel patterns using three dimensional geographical information systems: a methodological exploration with a large data set. Transp Res Part C 8(1–6):185–203

Langran G (1999) Time in geographic information systems. PhD thesis, University of Washington

Laube P, van Kreveld M, Imfeld S (2004) Finding REMO – detecting relative motion patterns in geospatial lifelines. In: Fisher PF (ed) Developments in spatial data handling, proceedings of the 11th international symposium on spatial data handling. Springer, Berlin/Heidelberg, pp 201–214

Lorentzos NA (1988) A formal extension of the relational model for the representation and manipulation of generic intervals. PhD thesis, Birbeck College, University of London

Mamoulis N, Cao H, Kollios G, Hadjieleftheriou M, Tao Y, Cheung D (2004) Mining, indexing, and querying historical spatiotemporal data. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining. ACM, New York, pp 236–245

Šaltenis S, Jensen CS, Leutenegger ST, Lopez MA (2000) Indexing the positions of continuously moving objects. In: Proceedings of the ACM SIGMOD international conference on management of data, Dallas, pp 331–342

Verhein F, Chawla S (2006) Mining spatiotemporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases. In: Proceedings of the 11th international conference on database systems for advanced applications. Lecture notes in computer science, vol 3882. Springer, Berlin/Heidelberg, pp 187–201

Wilensky U, Netlogo flocking model. http://ccl.northwestern.edu/netlogo/models/Flocking

Moving Average Regression

► Spatial and Geographically Weighted Regression

Moving Object Constraint Databases

► Constraint Databases and Moving Objects
► Linear Versus Polynomial Constraint Databases
► Polynomial Spatial Constraint Databases

Moving Object Databases

► Moving Object Uncertainty

Moving Object Languages

Ralf Hartmut Güting
Faculty for Mathematics and Computer Science,
Distance University of Hagen, Hagen, Germany

Synonyms

Query languages for moving objects

Definition

The term refers to query languages for *moving objects databases*. Corresponding database systems provide concepts in their data model and data structures in the implementation to represent moving objects, i.e., continuously changing geometries. Two important abstractions are *moving point*, representing an entity for which only the time-dependent position is of interest, and *moving region*, representing an entity for which also the time-dependent shape and extent are relevant. Examples of moving points are cars, trucks, airplanes, ships, mobile phone users, RFID-equipped goods, and polar bears; examples of moving regions are forest fires, deforestation of the Amazon rain forest, oil spills in the sea, armies, epidemic diseases, and hurricanes.

There are two flavors of such databases. The first, which is called the *location management* perspective, represents information about a set of currently moving objects. Basically, one is interested in efficiently maintaining their locations and asking queries about the current and expected near future positions and relationships between objects. In this case, no information about histories of movement is kept. The second is called the *spatio-temporal data* perspective; here the complete histories of movements are represented. The goal in the design of query languages for moving objects is to be able to ask any kind of question about such movements, perform analyses, and derive information in a way that is as simple and elegant as possible. Such queries must be executed efficiently.

Historical Background

The field of moving objects databases, with the related query languages, came into being in the late 1990s mainly by two parallel developments. First, the Wolfson group developed a model in a series of papers Sistla et al. (1997) and Wolfson et al. (1998a, b, 1999) that allows one to keep track in a database of a set of time-dependent locations, e.g., to represent vehicles. They observed that one should store in a database not the locations directly, which would require high update rates, but rather a motion vector, representing an object's expected position over time. An update to the database is needed only when the deviation between the expected position and the real position exceeds some threshold. At the same time this concept introduces an inherent, but bounded uncertainty about an object's real location. The group formalized this model introducing the concept of a *dynamic attribute*. This is an attribute of a normal data type which changes implicitly over time. This implies that results of queries over such attributes also change implicitly over time. They introduced a related query language called future temporal logic (FTL) that allows one to specify time-dependent relationships between expected positions of moving objects. Hence this group established the location-management perspective.

Second, the European project CHOROCHRONOS set out to integrate concepts from spatial and temporal databases and explored the *spatio-temporal data* perspective. This means, one represents in a database time-dependent geometries of various kinds such as points, lines, or regions. Earlier work on spatio-temporal databases had generally admitted only discrete changes. This restriction was dropped and continuously changing geometries were considered. Güting and colleagues developed a model based on the idea of *spatio-temporal data types* to represent histories of continuously changing geometries (Erwig et al. 1999; Güting et al. 2000; Forlizzi et al. 2000; Cotelo Lema et al. 2003). The model offers data types such as *moving point* or *moving region* together with a comprehensive set of operations. For example, there are operations to compute the projection of a moving point into the plane, yielding a *line* value, or to compute the distance between a moving point and a moving region, returning a time dependent real number, or *moving real*, for short. Such data types can be embedded into a DBMS data model as attribute types and can be implemented as an extension package.

A second approach to data modeling was pursued in CHOROCHRONOS by Grumbach and colleagues who applied the constraint model to the representation of moving objects (Grumbach et al. 1998; Rigaux et al. 2003) and implemented a prototype called Dedale. Constraint databases can represent geometries in n-dimensional spaces; since moving objects exist in 3D (2D + time) or 4D (3D + time) spaces, they can be handled by this approach. Several researchers outside CHOROCHRONOS also contributed to the development of constraint-based models for moving objects.

Scientific Fundamentals

The following two subsections describe two major representations for the location management and the spatio-temporal data flavor of moving objects databases in some detail, namely the MOST model and FTL language, and the approach of spatio-temporal data types. In a short closing subsection we mention some further work related to languages for moving objects.

Modeling and Querying Current Movement: The MOST Model and FTL Language

In this section we discuss moving objects databases based on the location management perspective and a related query language. That is, the database keeps track of a collection of objects moving around currently and we wish to be able to answer queries about the current and expected near-future positions. Such sets of entities might be taxi cabs in a city, trucks of a logistics company, or military vehicles in a military application. Possible queries might be:

- Retrieve the three free cabs closest to Cottle Road 52 (a passenger request position).
- Which trucks are within 10 km of truck T70 (which needs assistance)?
- Retrieve the friendly helicopters that will arrive in the valley within the next 15 min and then stay in the valley for at least 10 min.

Statically, the positions of a fleet of taxi cabs, for example, could be easily represented in a relation

```
taxi cabs(id: int, pos: point).
```

Unfortunately this representation needs frequent updates to keep the deviation between real position and position in the database small. This is not feasible for large sets of moving objects.

The moving objects spatio-temporal (MOST) data model (Sistla et al. 1997; Wolfson et al. 1999) discussed in this section stores, instead of absolute positions, a motion vector which represents a position as a linear function of time. This defines an expected position for a moving object. The distance between the expected position and the real position is called the *deviation*. Furthermore, a *distance threshold* is introduced and a kind of contract between a moving object and the database server managing its position is assumed. The contract requires that the moving object observes the deviation and sends an update to the server when it exceeds the threshold. Hence the threshold establishes a bound on the *uncertainty* about an object's real position.

The MOST model relies on a few basic assumptions: A database is a set of object classes. Each object class is given by its set of attributes. Some spatial data types like *point*, *line*, or *polygon* with suitable operations are available. Object classes may be designated as spatial which means they have a single spatial attribute. Spatial operations can then be directly applied to objects, e.g., *distance* (o_1, o_2) for two objects o_1 and o_2. Besides object classes, the database contains an object called *Time* which yields the current time at every instant. Time is assumed to be discrete and can be represented by *integer* values. The value of the *Time* object increases by one at each clock tick (e.g., every second).

Dynamic Attributes

A fundamental new concept in the MOST model is that of a *dynamic attribute*. Each attribute of an object class is classified to be either static or dynamic. A dynamic attribute is of a standard data type (e.g., *int*, *real*) within the DBMS conceptual model, but changes its value automatically over time. This means that queries involving such attributes also have time-dependent results, even if time is not mentioned in the query and no updates to the database occur.

For a data type to be eligible for use in a dynamic attribute, it is necessary that the type has a value 0 and an addition operation. This holds for numeric types but can be extended to types like *point*. A dynamic attribute A of type T is then represented by three subattributes *A.value*, *A.updatetime*, and *A.function*, where *A.value* is of type T, *A.updatetime* is a time value, and *A.function* is a function f: *int*→*T* such that at time $t = 0$, $f(t) = 0$. The semantics of this representation is called the value of A at time t and defined as

$$\begin{aligned} value(A,t) =& A.value + A.function \\ & (t - A.updatetime). \\ & \text{for} \quad t \geq A.updatetime \end{aligned}$$

When attribute A is mentioned in a query, its dynamic value *value*(A, t) is meant.

Representing Object Positions

A simple way of modeling objects moving freely in the *xy*-plane would be to introduce an attribute *pos* with two dynamic subattributes *pos.x* and *pos.y*. For example, we might define an object class for cars:

```
cars (license_plate: string, pos:
      (x: dynamic real,
       y: dynamic real)).
```

For vehicles, a more realistic assumption is that they move along road networks. A more sophisticated modeling of time dependent positions in MOST uses a *loc* attribute with six subattributes *loc.route*, *loc.startlocation*, *loc.starttime*, *loc.direction*, *loc.velocity*, and *loc.uncertainty*. Here *loc.route* is a (pointer to) a *line* value (a polyline) describing the geometry of the road on which the vehicle is moving. Say that on *loc.route*, one chooses a point as origin and a particular direction as positive direction. Then, the initial location *loc.startlocation* is given by its distance from the origin; this distance is positive if the direction from the origin to the initial location is in the positive direction, otherwise it is negative. The velocity also is negative or positive accordingly. *Startlocation*, *starttime*, and *velocity* correspond to the components of a dynamic attribute explained above. The value of *loc* at time t, *value*(*loc*, t) is now a position on the *route* polyline defined in the obvious way. Query evaluation may take the *uncertainty* into account.

Semantics of Queries, Query Types

In traditional databases, the semantics of a query are defined with respect to the current state of a database. This is not sufficient for the MOST model, as queries may refer to future states of a database. A *database state* is a mapping that associates each object class in the database with a set of objects of appropriate types, and the *Time* object with a time value. Let $o.A$ and $o.A.B$ denote attribute A of object o and subattribute B of attribute A of object o, respectively. In database state s, the value of $o.A$ is denoted $s(o.A)$ and the value of the *Time* object as s(*Time*). For each dynamic attribute A, its value in state s is *value(A, s(Time))*.

The semantics of queries are now defined relative to a database history. A *database history* is an infinite sequence of database states, one for each clock tick, beginning at some time u, hence is s_u , $s_{u+1}, s_{u+2}, \ldots$ An update at some time $t > u$ will affect all database states from t on. Hence with each clock tick there is a new database state, and with each update a new database history. Let $Q(H, t)$ denote a query Q evaluated on database history H assuming a current time t.

There are now two types of queries, namely *instantaneous* and *continuous* query (There exists a further query type, persistent query, which is omitted here.). An instantaneous query issued at time t is evaluated once on the history starting at time t, hence:

$$Q(H_t, t)\ .(instantaneous\ query)$$

In contrast, a continuous query is (conceptually) reevaluated once for each clock tick, hence as a sequence of instantaneous queries:

$$Q(H_t, t), Q(H_{t+1}, t+1), Q(H_{t+2}, t+2), \ldots \quad (continuous\ query)$$

The result of a continuous query changes over time; at time u the result of $Q(H_u, u)$ is valid. Of course, reevaluating the query on each clock tick is not feasible, instead, the evaluation algorithm for such queries is executed only once and produces a time dependent result in the form of a set of tuples with associated time stamps. Reevaluation is necessary only for explicit updates.

The Language FTL

The query language associated with the MOST model is called FTL. The following are example queries formulated in FTL.

1. Which trucks are within 10 km of truck T70?

   ```
   RETRIEVE t
   FROM trucks t, trucks s
   WHERE s.id = 'T70' ^ dist(s, t)
            <= 10
   ```

   Here nothing special happens, yet, the result is time dependent.
2. Retrieve the helicopters that will arrive in the valley within the next 15 min and then stay in the valley for at least 10 min.

   ```
   RETRIEVE h
   FROM helicopters h
   WHERE eventually_within_15
     (inside(h, Valley) ^ always_
     for_10 (inside(h, Valley))
   ```

   Here *Valley* is a polygon object.

The general form of a query in FTL is (In the original literature about FTL, a single class of moving objects is assumed and the FROM clause omitted.)

```
        RETRIEVE < target-list >
          FROM < object classes >
          WHERE <FTL-formula>.
```

The interesting part is the FTL formula. FTL formulas are similar to first-order logic, hence they are built from constants, function symbols, predicate symbols, variables and so forth. Some special constructs in the definition of formulas are the following:

- If f and g are formulas, then f **until** g and **nexttime** f are formulas

The semantics of a formula are defined with respect to:

- A variable assignment μ which associates with each variable in the formula a corresponding database object (eg., for$s$, t in the first example query $\mu = [s, T_{10}), (t, T_{20})]$ where T_i are truck objects in the database)
- A database state s on history h

Next, it is necessary to define what it means for a formula to be satisfied at state s on history H with respect to variable assignment μ (satisfied at (s, μ) for short). The semantics of the special constructs are defined as follows:

- f **until** g is satisfied at (s, μ):$\Leftrightarrow$ either g is satisfied at (s, μ), or there exists a future state s' on history H such that (g is satisfied at (s', μ) $\quad\wedge$ for all states s_i on history H before state s', f is satisfied at (s_i, μ))
- **nexttime** f is satisfied at (s, μ):$\Leftrightarrow$ f is satisfied at (s', μ) where s' is the state immediately following s in history H.

Based on these temporal operators with well-defined semantics, some derived notations can be defined:

- **eventually** $g \equiv$ *true* **until** g
- **always** $g \equiv$ ($\neg$**eventually** $(\neg g)$)

In addition, it is useful to have bounded temporal operators:

- f **until_within_c** g asserts that there exists a future time within c units of time from now such that g holds and until that time f will hold continuously
- f **until_after_c** g asserts that there exists a future time after at least c units of time from now such that g holds and until that time f will hold continuously

Based on these, one can again define further bounded temporal operators:

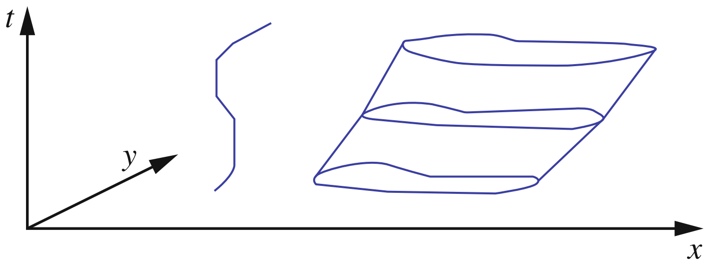
- **eventually_within_c** $g \equiv$ *true* **until_within_c** g
- **eventually_after_c** $g \equiv$ *true* **until_after_c** g
- **always_for_c** $g \equiv g$ **until_after_c** *true*

That concludes the explanation of the semantics of the constructs used in example query 2 above.

Evaluation

The algorithm for evaluating FTL queries can only be briefly sketched here. The basic idea is to compute a relation for every subformula of the given FTL formula bottom up, starting from atomic formulas like `dist(s, t)<= 10`. The relation R_f for subformula f has an attribute for each free variable occurring in f, and two attributes for time stamps t_{start}, t_{end}. Hence the relation for formula `dist(s, t)<= 10` has schema $(s, t, t_{\text{start}}, t_{\text{end}})$. It has a tuple (o, o', T, T') for every pair of objects O, O' and for every maximal time interval $[T, T']$ such that objects O, O' are within distance 10 throughout the interval $[T, T']$. For operators combining two subformulas such as $f \wedge g$ or f **until** g it is then possible to compute their relations essentially by joins over common object identifier attributes (equal variables in both subformulas), manipulating the time stamps in an appropriate way.

The result relation for the complete formula is the result for a continuous query. As time progresses, tuples of this relation can be added to



Moving Object Languages, Fig. 1 A moving point and a moving region

or removed from the current result. It can also be used to answer an instantaneous query selecting just the tuples valid at the time of issuing this query.

Modeling and Querying History of Movement: Spatio-Temporal Data Types

This section discusses the spatio-temporal data perspective for moving objects databases. That is, we consider the geometries stored in spatial databases and allow them to change continuously over time. Some of the most important abstractions used in spatial databases are

- Point – an object for which only the position in space is relevant
- Line – a curve often representing connections, such as roads and rivers
- Region – representing objects for which the extent is relevant
- Partition – subdivisions of the plane, e.g. of a country into states
- Network – graph or network structures over roads, rivers, power lines, etc.

Such abstractions are usually captured in *spatial data types*, consisting of the type together with operations.

Spatio-Temporal Data Types

The idea of the approach (Erwig et al. 1999) presented in the following is to introduce spatio-temporal data types that encapsulate time dependent geometries with suitable operations. For moving objects, point and region appear to be most relevant, leading to data types *moving point* and *moving region*, respectively. The *moving point* type can represent entities such as vehicles, people, or animals moving around whereas the *moving region* type can represent hurricanes, forest fires, armies, or flocks of animals, for example. Geometrically, values of spatio-temporal data types are embedded into a 3D space (2D + time) if objects move in the 2D plane, or in a 4D space if movement in the 3D space is modeled. Hence, a moving point and a moving region can be visualized as shown in Fig. 1.

In the following, we first motivate the approach by introducing an example database with spatio-temporal data types, providing a number of operations on these data types, and formulating queries using these operations. We then discuss the underlying design principles for types and operations, the distinction between abstract model and discrete model in defining the semantics of types, and the structure of type system and operations in more detail. Finally, the implementation strategy is briefly outlined.

Example Operations and Queries

As a simple example, suppose we have two relations representing cars and weather conditions, whose movements have been recorded.

```
cars(license_plate: string,
     trip: mpoint)

weather(id: string, area:
        mregion)
```

Here *mpoint* and *mregion* are abbreviations for the *moving point* and *moving region* data types, respectively. Assume further, some available operations on these types, used in queries below, are the following:

Signature		Operation
moving(*point*)	→ *line*	**trajectory**
moving(*region*)	→ *region*	**traversed**
moving(α)	→ *periods*	**deftime**
moving(*point*) × *moving*(*region*)	→ *moving*(*point*)	**intersection**
moving(α)× *instant*	→ *intime*(α)	**atinstant**
intime(α)	→ *instant*	**inst**
intime(α)	→ α	**val**
periods	→ *int*	**duration**
int × *int* × *int* × *int*	→ *instant*	**theinstant**

In these signatures, *moving* is viewed as a type constructor that transforms a type α into a time dependent version of that type, *moving*(α). These operations use further data types:

- *instant*, representing an instant of time
- *periods*, representing a set of disjoint time intervals
- *intime* α), where *intime* is a type constructor building pairs of an *instant* and a value of another type α

The operations have the following meaning: **Trajectory** and **traversed** compute the projection of a moving point or moving region into the 2D plane; **deftime** computes the projection on the time axis. The intersection of a moving point and a moving region is a moving point again containing the parts of the moving point inside the moving region. **Atinstant** evaluates the moving object at a particular instant of time, returning an (*instant*, α) pair. **Inst** and **val** allow access to the components of *intime*(α) pairs. **Duration** returns the total length of time intervals in a periods value (say, in seconds). Operator **theinstant** constructs instants of time for a variable number of integer arguments (here four) in the order year, month, day, hour, minute, and second returning the first instant of time of such a time interval.

We can then formulate the following queries:

1. What was the route taken by the car "DO-GL 871"?

```
SELECT trajectory(trip) AS route
FROM cars WHERE license_plate
   = "DO-GL 871"
```

2. What was the total area swept by the fog region with identifier “F276”?

```
SELECT traversed(area) AS fogarea
FROM weather WHERE id = "F276"
```

3. How many cars stayed in the fog area for more than 30 min?

```
SELECT count(*)
FROM cars AS c, weather AS w
WHERE duration(deftime(inter-
section (c.trip,
   w.area))) > 1800
```

4. Where was the fog area at 5 p.m. (on the respective day January 8, 2007)?

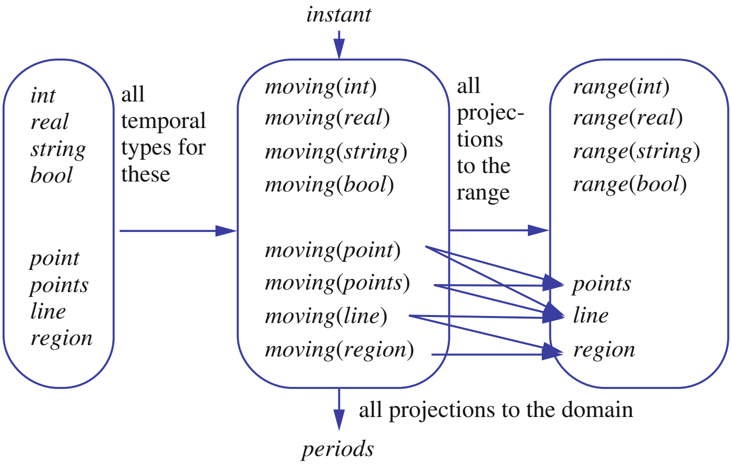
```
SELECT val(atinstant(area,
theinstant (2007, 1, 8, 17)))
FROM weather WHERE id = "F276"
```

Goals in the Design of Types and Operations

The examples have illustrated the basic approach of using spatio-temporal data types. Starting from this idea, the question is how exactly data types and operations should be chosen. Such a systematic design was given in Güting et al. (2000) and the types and operations above are already part of that design. The design pursues the following goals:

- *Closure of type system*. Type constructors should be applied systematically and consistently. This means in particular:
  - For all base types of interest, we have related temporal (“moving”) types.
  - For all temporal types (whose values are functions from time into some domain), there exist types to represent their projection into domain and range.
- *Genericity*. There will be a large set of data types. Operations should be designed in a generic way to cover as many types as possible.
- *Consistency between nontemporal and temporal types*. The structure of a temporal type, taken at a particular instant of time, should agree with the structure of the corresponding static type.

Moving Object Languages, Fig. 2 Structure of the type system



- *Consistency between nontemporal and temporal operations*. For example,

$$\textbf{val}(\textbf{atinstant}(\textbf{intersection}(mp, mr), t)) =$$
$$\textbf{intersection}(\textbf{val}(\textbf{atinstant}(mp, t)),$$
$$\textbf{val}(\textbf{atinstant}(mr, t))$$

Abstract and Discrete Model

Before considering the type system in more detail, one should understand the meaning of data types. There exists a choice at what level we define the semantics of types. For example, a moving point could be viewed in two ways:

- A continuous function from time (viewed as isomorphic to the real numbers) into the Euclidean plane, i.e., a function $f : \mathbb{R} \to \mathbb{R}^2$.
- A polyline in the 3D space representing such a function.

The essential difference is that in the first case, we define the semantics of the type in terms of infinite sets without fixing a finite representation. In the second case we choose a finite representation. We call the first an *abstract model* and the second a *discrete model*. Note that there are many discrete models for a given abstract model. For example, a moving point might also be represented as a sequence of splines. The properties of such models can be summarized as follows:

- Abstract models are mathematically simple, elegant, and uniform, but not directly implementable.
- Discrete models are more complex and heterogeneous, but can be implemented.

As a consequence, the design of spatio-temporal data types proceeds in two steps: First, an abstract model of types and operations is designed. Second, a discrete model to represent (a large part of) the abstract model is constructed.

Type System

The structure of the type system is illustrated in Fig. 2. It reflects the design goals stated above. Projections of standard types are represented as sets of disjoint intervals over the respective base type; the *range* type constructor yields such types *range*(α) for base type α. Projections of time dependent geometries can generally be of different data types. For example, a moving point can "jump around", i.e., change position in discrete steps, yielding a *points* value (set of points) as a projection. Or it can move continuously, yielding a *line* value (a curve in the plane). Note that *line* and *region* values can have multiple components, so that the respective projection operations are closed. The *intime*(α) types have been omitted in this figure.

Operations

The design of operations proceeds in three steps:

1. Carefully design operations for nontemporal types, using generic definition techniques.
2. By a technique called lifting make them all time dependent in a way consistent with the static definition.
3. Add specialized operations for the temporal types.

There is a comprehensive set of operations first on the nontemporal types having classes of operations such as predicates, set operations, aggregate, numeric, distance and direction operations. Second, these are all lifted, which means each of their arguments may become time dependent which makes the result time dependent as well. Third, specialized operations on temporal types have classes of operations addressing projection to domain and range, interaction with values in domain and range, and operations to deal with rate of change (e.g., derivative).

Implementation

Implementation is based on the discrete model proposed in Forlizzi et al. (2000) using algorithms for the operations studied in Cotelo Lema et al. (2003). The discrete model uses the so-called *sliced representation* as illustrated in Fig. 3.

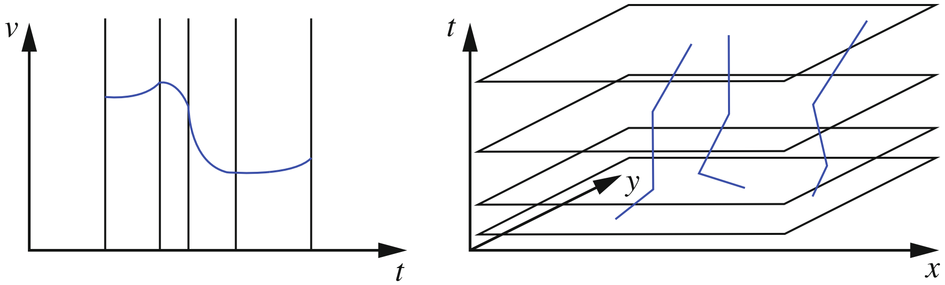
A temporal function value is represented as a time-ordered sequence of *units* where each unit has an associated time interval and time intervals of different units are disjoint. Each unit is capable of representing a piece of the moving object by a "simple" function. Simple functions are linear functions for moving points or regions, and quadratic polynomials (or square roots of such) for moving reals, for example.

Within a database system, an extension module (data blade, cartridge, extender, etc.) can be provided offering implementations of such types and operations. The sliced representation is basically stored in an array of units (It is a bit more complicated in the case of variable size units as for a moving region, for example.). Because values of moving object types can be large and complex, the DBMS must provide suitable storage techniques for managing large objects. A large part of this design has been implemented prototypically in the SECONDO extensible DBMS (Almeida et al. 2006) which is available for download (Secondo System).

Some Further Work on Moving Object Languages

Moving Objects in Networks

The model of spatio-temporal data types presented in the previous section has been extended to model movement in networks (Güting et al. 2006). A network is modeled as a set of routes and junctions between routes. Four data types *gpoint*, *gline*, *moving*(*gpoint*) and *moving*(*gline*) are introduced to represent static and moving network positions and regions, respectively. Rep-



Moving Object Languages, Fig. 3 Sliced representation of a *moving*(*real*) and a *moving*(*points*) value resentative entities on a highway network would be gas stations, construction areas, vehicles, or traffic jams, for example. Some advantages over a model with free movement are that descriptions of moving objects become much more compact (as they do not contain geometries any more) such that relationships between moving objects and the underlying network can be easily used in queries, and that connectivity of the network is considered, e.g., for network distance or shortest path computations.

Spatio-Temporal Predicates and Developments

Erwig and Schneider (1999, 2002) extend the spatiotemporal data type approach by considering developments of topological relationships over time. They develop a language to describe such developments in predicates that can then be used for filter and join conditions on moving objects. For example, consider an airplane traversing a storm area. The topological relationship between the moving point and the moving region will be *disjoint* for a period of time, then *meet* for an instant, then *inside* for a time interval, then *meet* again and finally *disjoint* again. The framework first allows one to obtain basic spatiotemporal predicates by aggregating a static topological relationship over all instants of a time interval. This is essentially done by lifting (as explained above) and existential or universal quantification. Existing predicates can then be sequentially composed to derive new predicates. For example, we may define a predicate **Cross** to describe a development like the passing of the air plane through the storm as:

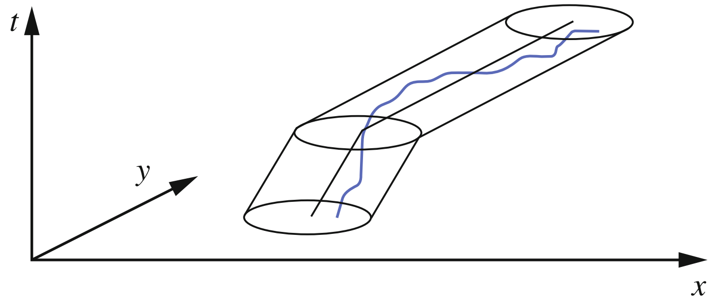
$$\textbf{Cross}:=\textbf{Disjoint}\triangleright\textbf{meet}\triangleright\textbf{Inside}\triangleright\textbf{meet}\triangleright\textbf{Disjoint}$$

Uncertain Trajectories

A *moving point* (Fig. 1) in the 2D + time space is in the literature often called a *trajectory*. If one represents it discretely as a polyline and takes an uncertainty threshold into account, geometrically it will obtain the shape of a kind of slanted cylinder (Fig. 4).

It is only known that the real position is somewhere inside this volume. Based on this model, Trajcevski et al. (2004) have defined a set of predicates between a trajectory and a region in space taking uncertainty and aggregation over time into account, namely

PossiblySometimeInside	**SometimePossiblyInside**
PossiblyAlwaysInside	**AlwaysPossiblyInside**
DefinitelySometimeInside	**SometimeDefinitelyInside**
DefinitelyAlwaysInside	**AlwaysDefinitelyInside**



Moving Object Languages, Fig. 4 Geometry of an uncertain trajectory

They also give algorithms for evaluating such predicates.

A text book covering the topics presented in this article in more detail is available (Güting and Schneider 2005).

Key Applications

Query languages of the first kind, that is, for querying current and near future movement like the MOST model described, are the foundation for location-based services. Service providers can keep track of the positions of mobile users and notify them of upcoming service offers even some time ahead. For example, gas stations, hotels, shopping centres, sightseeing spots, or hospitals in case of an emergency might be interesting services for car travelers.

Several applications need to keep track of the current positions of a large collection of moving objects, for example, logistics companies, parcel delivery services, taxi fleet management, public transport systems, air traffic control. Marine mammals or other animals are traced in biological applications. Obviously, the military is also interested in keeping track of fighting units in battlefield management.

Query languages of the second kind – for querying history of movement – are needed for more complex analyses of recorded movements. For example, in air traffic control one may go back in time to any particular instant or period to analyse dangerous situations or even accidents. Logistics companies may analyze the paths taken by their delivery vehicles to determine whether optimizations are possible. Public transport systems in a city may be analyzed to understand reachability of any place in the city at different periods of the day. Movements of animals may be analyzed in biological studies. Historical modeling may represent movements of people or tribes and actually animate and query such movements over the centuries.

Query languages of the second kind not only support moving point entities but also moving regions. Hence also developments of areas on the surface of the earth may be modeled and analyzed like the deforestation of the Amazone rain forest, the Ozone hole, development of forest fires or oil spills over time, and so forth.

Future Directions

Recent research in databases has often addressed specific query types like continuous range queries or nearest neighbour queries, and then focused on designing efficient algorithms for them. An integration of the many specific query types into complete language designs as presented in this entry is still lacking. Uncertainty may be treated more completely also in the approaches for querying history of movement. A seemless query language for querying past, present, and near future would also be desirable.

Cross-References

▸ Spatiotemporal Data Types
▸ Trajectory

References

Almeida VT, Güting RH, Behr T (2006) Querying moving objects in SECONDO. In: Proceedings of the mobile data management conference, Nara, pp 47–51

Cotelo Lema JA, Forlizzi L, Güting RH, Nardelli E, Schneider M (2003) Algorithms for moving object databases. Comput J 46(6):680–712

Erwig M, Schneider M (1999) Developments in spatio-temporal query languages. In: IEEE international workshop on spatio-temporal data models and languages (STDML), Florence, pp 441–449

Erwig M, Schneider M (2002) Spatiotemporal predicates. IEEE Trans Knowl Data Eng (TKDE) 14(4):881–901

Erwig M, Güting RH, Schneider M, Vazirgiannis M (1999) Spatio-temporal data types: an approach to modeling and querying moving objects in databases. GeoInformatica 3:265–291

Forlizzi L, Güting RH, Nardelli E, Schneider M (2000) A data model and data structures for moving objects databases. In: Proceedings of ACM SIGMOD conference, Dallas, pp 319–330

Grumbach S, Rigaux P, Segoufin L (1998) The DEDALE system for complex spatial queries. In: Proceedings of ACM SIGMOD conference, Seattle, pp 213–224

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

Güting RH, Böhlen MH, Erwig M, Jensen CS, Lorentzos NA, Schneider M, Vazirgiannis M (2000) A foundation for representing and querying moving objects in databases. ACM Trans Database Syst 25:1–42

Güting RH, de Almeida VT, Ding Z (2006) Modeling and querying moving objects in networks. VLDB J 15(2):165–190

Rigaux P, Scholl M, Segoufin L, Grumbach S (2003) Building a constraint-based spatial database system: model, languages, and implementation. Inf Syst 28(6):563–595

SECONDO System. Available for download at http://www.informatik.fernuni-hagen.de/import/pi4/Secondo.html/

Sistla AP, Wolfson O, Chamberlain S, Dao S (1997) Modeling and querying moving objects. In: Proceedings of 13th international conference on data engineering, Birmingham, pp 422–432

Trajcevski G, Wolfson O, Hinrichs K, Chamberlain S (2004) Managing uncertainty in moving objects databases. ACM Trans Database Syst (TODS) 29(3):463–507

Wolfson O, Chamberlain S, Dao S, Jiang L, Mendez G (1998) Cost and imprecision in modeling the position of moving objects. In: Proceedings of 14th international conference on data engineering, Orlando, pp 588–596

Wolfson O, Xu B, Chamberlain S, Jiang L (1998) Moving objects databases: issues and solutions. In: Proceedings of 10th international conference on scientific and statistical database management, Capri, pp 111–122

Wolfson O, Sistla AP, Chamberlain S, Yesha Y (1999) Updating and querying databases that track mobile units. Distrib Parallel Database 7:257–387

Moving Object Uncertainty

Goce Trajcevski and Peter Scheuermann
Department of Electrical Engineering and Computer Science, Northwestern University,
Evanston, IL, USA

Synonyms

Dead-Reckoning; GPS; Location-Based Services; Motion Tracking; Moving Object Databases; Spatiotemporal Uncertainty

Definition

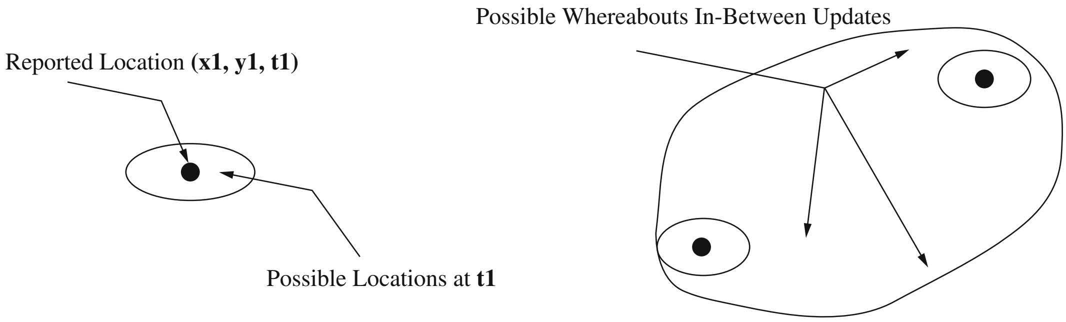
Uncertainty is an inherent component of any system that manages the data pertaining to the location-in-time information of mobile entities. Typically, such systems receive some form of an on-line *(location, time)* updates for individual moving objects, which may be obtained either from a GPS device on-board each moving object or by using some other tracking technology (e.g., PCS triangulation networks and motion tracking sensor networks). This information is transmitted to a repository which stores the data and can be used for providing answers to various queries of interest. However, there are several constraints which limit the accuracy of these data:

1. Due to bandwidth limitations, networks' connectivity and various clocksynchronization issues, the *actual* time that a given object was at a particular location may not be equal to the time that its presence at that location is recorded in the database. This, in a sense, resembles the traditional distinction between *valid time* and *transaction time* as studied in temporal databases (Böhlen and Jensen 2003).
2. The devices that are used to determine the location of a given object are prone to measurementerrors themselves. For example, depending on the number of satellites whose signals are available in a given region, the GPS-based error may range from few decimeters to a few meters (Pfoser et al. 2005); the sensor nodes' coverage may not be sufficient to guarantee exact location (Cheng et al. 2004).
3. Furthermore, one cannot make any exact claims about the location of a given object in-between consecutive updates (Pfoser and Jensen 1999; Trajcevski et al. 2004).

These basic concepts are illustrated in Fig. 1, where the left portion indicates the uncertainty with respect to the object's location for a particular update, and the right portion indicates the boundaries of the region of possible whereabouts of the object between the updates.

Historical Background

Miniaturization of computing devices and the advances of wireless communications and sensor technologies have spurred many research and implementation efforts into several classes of


Reported Location (x1, y1, t1)
Possible Locations at t1
Possible Whereabouts In-Between Updates

Moving Object Uncertainty, Fig. 1 Example of moving object uncertainty

applications that can be grouped as Location-Based Services (LBS) (Schiller and Voisard 2004). An important aspect of almost any LBS system is the management (i.e., modelling, storing/retrieving and querying) of the transient location-in-time data pertaining to the mobile entities involved. As a result, in recent years, a large body of research works have emerged, which are collectively forming the field of Moving Objects Databases (MOD) (Güting 2005).

Historically, researchers have independently pursued two complementary tracks that have extended the traditional database research. On one hand, the transactional aspects of database management systems were extended with temporal awareness and the field of temporal databases investigated various impacts that the semantics of time has on the quality of the stored data with respect to the actual world being modelled (Böhlen and Jensen 2003). On the other hand, the management (i.e., representation, indexing, querying) of the spatial data for applications such as GIS, in which the entities of interest are objects with particular geographic and dimensionality attributes (e.g., location, shape, extent) was investigated (Samet 2006). One of the first commercially-important applications in which the efficient management of the *(location, time)* information for a large number of mobile users is a paramount, was the cellular telephony. Various data management architectures were proposed for the efficient tracking, call-forwarding and billing of users (Pitoura and Samaras 2001). Around the same time, motivated by various LBS applications, researchers recognized the need for more thorough treatment of spatio-temporal data, i.e., data whose spatial attributes (e.g., location) change over time. In particular (Sistla et al. 1997), introduced the concept of dynamic attributes and spurred the development of the field of moving objects databases (Güting 2005). Due to the dynamic nature of the entities involved, data management in MOD settings has a number of distinguishing features, with respect to the traditional data management aspects of modelling/representing the data (Güting et al. 2003), indexing structures for accessing the data items (Pelanis et al. 2006; Tao et al. 2003), and algorithms and methodologies for processing the spatio-temporal queries (Ding et al. 2007; Gedik and Liu 2006; Güting 2005). In particular, the most widely used queries (range, (k)nearest-neighbor, join) become *continuous*. In other words, their answers change over time due to the changes in the locations of the moving objects, and efficient techniques are needed for maintaining the correctness of the answers.

Different works have used different models to represent the motion plans of the moving objects and, based on the model chosen, a plethora of indexing structures and algorithms for processing the popular categories of spatio-temporal queries have been proposed (Güting 2005; Tao et al. 2003). However, irrespective of the chosen representation, the *uncertainty* of the moving objects remains an inherent component. As has been pointed out in many works (e.g., Trajcevski et al.

2004; Wolfson et al. 1999), unless the uncertainty is captured in the model itself, the burden of factoring it out from the meaning/answers of the various spatio-temporal queries will fall on the end user. Hence, we need the following: (1) Models of the uncertainty as part of the objects' motion model. (2) Linguistic constructs that will enable the users to specify queries in the presence of uncertainty. (3) Techniques for efficient query processing for uncertain moving objects' data.

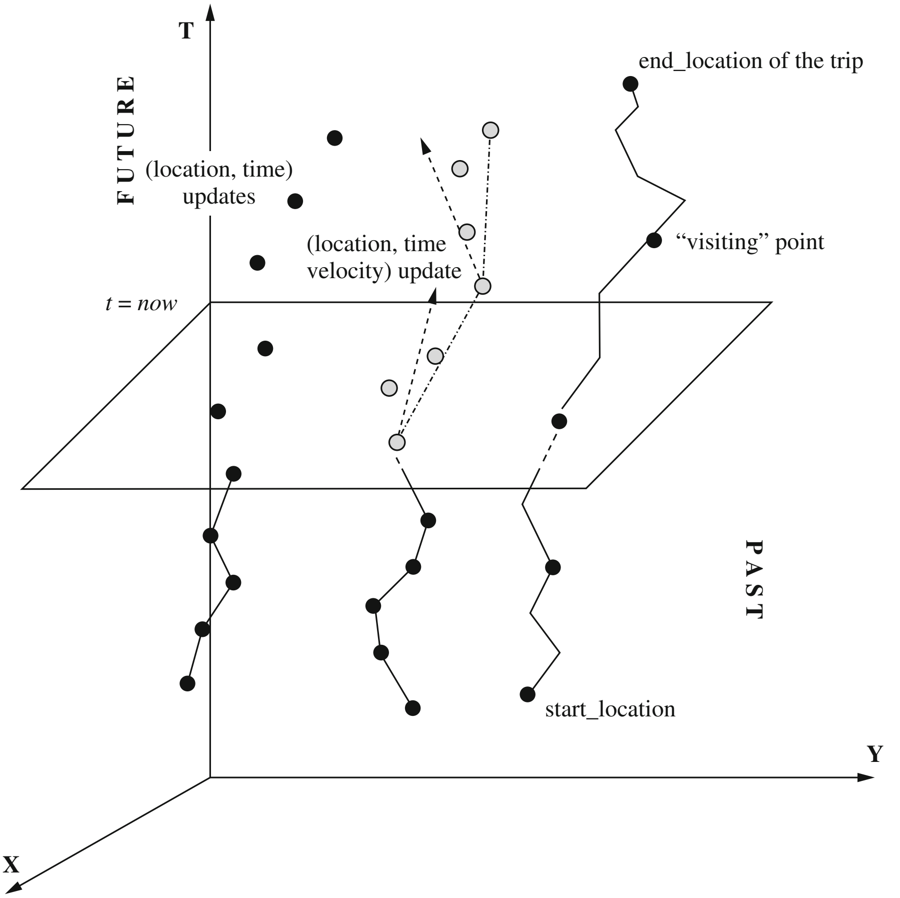
Scientific Fundamentals

Traditionally, there are three main models for representing the future motion plans of moving point objects. As a consequence of the chosen model, the researchers have developed corresponding algorithms for the processing of continuous spatio-temporal queries.

- At one extreme is the model in which the objects periodically send their *(location, time)* updates to the MOD server (left-most part in Fig. 2). Due to the frequency of the updates, intelligent methodologies are needed that will avoid constant reevaluation of the pending continuous queries, while still ensuring the correctness of their answers (Mokbel et al. 2004). In order to balance the efficiency of query processing with keeping the MOD up-to-date, recent works have also addressed "lazy" updating mechanisms (Xing and Aref 2006).
- In the "middle-land" is the model in which the moving objects are assumed to periodically send *(location, time, velocity)* updates to the MOD server (Sistla et al. 1997), as illustrated by the middle portion of Fig. 2. Efficient algorithms for processing continuous queries in MOD under this model were presented in Iwerks et al. (2006), and Gedik and Liu (2006) addressed the distributed processing of such queries, by delegating some of the responsibilities to the moving objects themselves. One peculiar feature of this model is that in-between two consecutive updates, the objects are allowed to deviate from the expected route which is calculated using the *velocity* parameter of the most recent update, for as long as the deviation is within certain tolerance bounds (Wolfson et al. 1999). This is illustrated by the shaded circles in the middle portion of Fig. 2, which shows the actual locations-in-time, as opposed to the expected ones that would be along the dotted arrowed line.
- The other extreme model is the one in which the entire future motion plan of a given object is represented as a *trajectory* (right-most portion in Fig. 2). Under this model, each object is assumed to initially transmit to the MOD server the information about its *start_location*, *end_location*, and *start_time* of the trip, plus (possibly) a set of "to-be-visited" points. Using the information available from the electronic maps, plus the knowledge about the distribution of the traffic patterns in a given time-period, the server will apply an A*like variant of the time-aware Dijkstra's algorithm to generate the optimal travel plan (Trajcevski et al. 2004). A trajectory is essentially a sequence of 3D points (2D geography + time) of the form $(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)$, where $t_1 < t_2 < \ldots t_n$ and in-between two points the object is assumed to move along a straight line and with a constant speed. The peculiarity of this model is that it enables answering continuous queries pertaining to the further-future; however, the consequence is that a disturbance of the traffic patterns in a small geographic region may affect the correctness of the queries in widely-dispersed areas and at different time-intervals (Ding et al. 2007).

An important observation is that when it comes to the *past* portion of the objects' motion, all three models, in a sense, converge, and represent it as a trajectory (bottom part of Fig. 2).

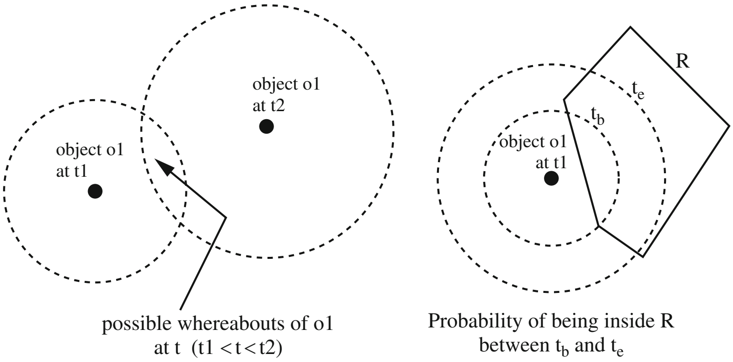
Any model of the uncertainty of the moving objects, as well as its implications on the query processing algorithms, is closely associated with the adopted model for representing the objects' motion plans.



Moving Object Uncertainty, Fig. 2 Modelling the motion plans of moving objects

As we already have illustrated in Fig. 1, even a single location update at a given time-instance is associated with its own uncertainty. Furthermore, if the motion plan of the moving objects is represented as a sequence of *(location, time)* updates, then there are other consequences of the uncertainty. As an illustration, observe the left portion of Fig. 3. Assume that it represents a scenario in which an *exact* value of the locations of a given object is known at times t_1 and t_2 (also assuming that the consecutive updates were sent at those times). However, even under this assumption (and, again, ignoring the fact that these very values already have uncertainty associated with them, cf., Fig. 1.), there is still some imprecision regarding the object's motion. Namely, given only the velocity bounds of the object o_1, its location at a time t ($t \in (t_1, t_2)$) cannot be exactly determined. As illustrated in the left portion of Fig. 3 (assuming that $(t - t_1) < (t_2 - t)$), at t, o_1 can be anywhere inside the lens obtained as the intersection of the of the two circles bounding the object's possible locations with respect to its maximum speed limit. It can be demonstrated that all the possible whereabouts of the object for the time-interval (t_1, t_2) are bound by an ellipse with foci at the locations reported at the respective times (Pfoser et al. 2005). If one assumes a certain probability distribution corresponding to the objects uncertainty (Cheng et al. 2004), then algorithms can be developed for answering the spatio-temporal queries with uncertainty. As a particular example, illustrated by the right portion of Fig. 3, one can pose the range query: **QR1:** *"What is the probability that the object o_1 will be inside the region R between*

Moving Object Uncertainty, Fig. 3 Uncertainty for the (location, time) model

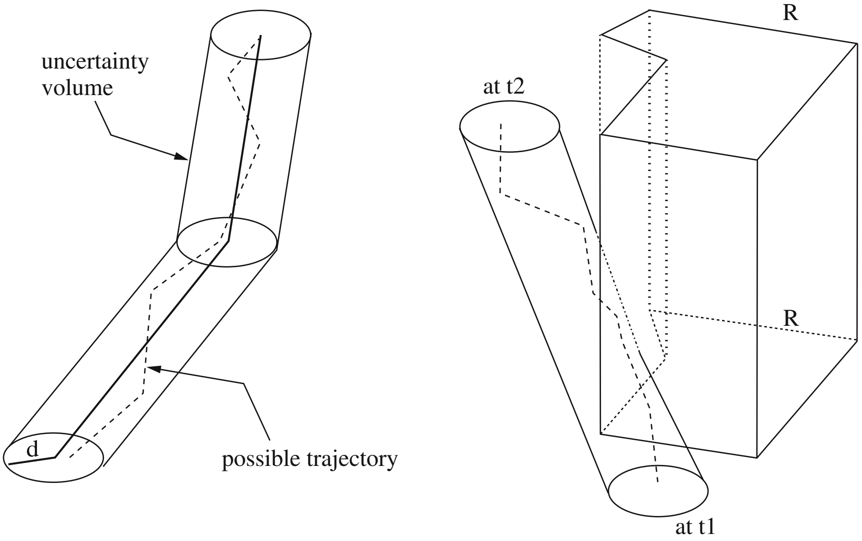


t_b *and* t_e", for which the value is obtained by calculating the area of the intersection of the circular ring bounding the location of the object between t_b and t_e, with the polygon bounding the region R, and dividing the result with the area of the circular ring. Another variant of the continuous range query with uncertainty is: **QR2:** *"Retrieve all the objects that have more than 75 % chance of being inside the region R between* t_b *and* t_e". One can easily envision the impact of the uncertainty on the other types of continuous spatio-temporal queries. For example, a variant of the nearest-neighbor query would be: **Q-NN:** *"Retrieve all the objects that have more then 90 % chance of being nearest neighbors to the object* o_1 *between* t_b *and* t_e".

When the motion plan of the moving object is modelled as a sequence of *(location, time, velocity)* updates (cf., the middle portion of the Fig. 2), the uncertainty is already incorporated as a part of the *contract/agreement* between the MOD server and the individual mobile objects. Namely, in order to minimize the communication overhead, after each update, the particular mobile object knows what is the *expected* location that the server calculates based on the *velocity* parameter. For as long as the expected location does not deviate by more than a certain predefined threshold, say, ε, from the actual location of the object obtained by some measurement (e.g., an on-board GPS device), the object will not send new updates to the MOD server. This kind of update policy is known as a *dead-reckoning*, and different variants of it have been explored in detail in Wolfson et al. (1999), along with the analytic expressions for deriving the overall *information cost* of keeping the (im)precision of the MOD data within desired bounds. Clearly, every spatio-temporal query under this model has an explicit uncertainty in its answer, due to the location error-bound of ε. Recent algorithms for efficient processing of the spatio-temporal queries under these settings were presented in Gedik and Liu (2006).

For the case when the future motion plan of a moving object is represented as a *trajectory*, since one of the parameters used in its construction is the distribution of the speed-patterns on the road-segments, the uncertainty represents the acceptable deviation of the objects due to traffic fluctuations.

The implications of this assumption are illustrated in Fig. 4. Namely, if the object is expected to be at some location, say, (x_1, y_1) at a given time t_1, the *uncertainty area* of its whereabouts is a disk with a radius d, centered at the expected location (x_1, y_1). Extending this over a time-interval, the set of all the *possible trajectories* of a given moving object defines an *uncertainty volume*, which in 3D settings (2D geography + time) is represented as a sequence of sheared cylinders, one for each straight-line segment of the object's route. Detailed algorithms for processing continuous spatio-temporal queries under these settings were presented in Trajcevski et al. (2004), where the *qualitative* uncertainty of the queries was discussed. Namely, to test for the satisfiability of the spatio-temporal predicates, the quantifiers *possibly* and *definitely* were considered in the spatial dimension, and the quantifiers considered in the temporal dimension were *sometimes* and *always*. The query operators corresponded to the



Moving Object Uncertainty, Fig. 4 Uncertainty for the (location, time) model

various predicates that can be composed by interleaving the order of the quantification. Thus, the right portion of Fig. 4 presents an example of an uncertain trajectory which satisfies the predicate: *"Possibly inside region R, sometimes between t_1 and t_2"*. An approach assuming quantitative probabilistic values for the answer to such queries, under the assumption of uniform distribution for the probability of the object being inside the disk with radius d was presented in Trajcevski (2003).

Clearly, one cannot dwell on query processing for any large data sets unless there are proper indexing techniques for retrieving that data, especially during the filtering stage in which disk-accesses should be minimized, while ensuring as few false-positives as possible. There has been a plethora of indexing structures proposed for processing continuous spatio-temporal queries (e.g., see the references in Güting 2005). Furthermore, recent research works have specifically focused on developing efficient indexes which explicitly take into consideration the uncertainty of the mobile entities (de Almeida and Güting 2005). One of the main benefits of incorporating the uncertainty into the index is that certain objects can be pruned from the search earlier during the filtering part of the processing of probabilistic queries.

Some of the recent works targeting efficient management of continuous spatio-temporal queries for moving objects have explicitly focused on deeper exploitation of the fact that in many applications of interest, the objects are moving on road networks (Ding and Güting 2004b). This additional semantic knowledge can be exploited to obtain further gains in the efficiency of objects' tracking and query processing. Recently, techniques have been developed for efficient tracking and indexing of moving objects on road networks (de Almeida and Güting 2005) and, in particular, the impact of uncertainty under such settings has been investigated in Ding and Güting (2004a).

Key Applications

Moving object uncertainty is of interest in several scientific and application domains.

GIS

Typically, in MOD research, the moving objects are approximated as points whose dimensions (e.g., size) can be neglected with respect to the overall area/volume of the universe of discourse. However, many of the objects of interest in GIS can not be approximated as points – namely, rivers have their shapes, forests cover areas typically represented as polygon-bounded regions, fires are spreading in time and are represented as moving polygons (Güting et al. 2003). However, the boundaries between regions are seldom precise (e.g., a boundary between a prairie and a desert). Hence, one must account for the uncertainty in the representation and use the corresponding mathematical tools to model it and develop query algorithms (e.g., fuzzy-set theory Pfoser et al. 2005).

LBS

A variety of applications in LBS can use the uncertainty inherent to mobile objects. As a typical example, in tourist-information systems, the main concern is how to provide a context-aware delivery of the data which matches the preferences of a given user based on its location (Schiller and Voisard 2004). However, if revealing the exact location of the mobile user can be adversely used for violating some of the privacy issues, uncertainty can be used to provide some forms of location-based privacy. As another example, depending on the trade-offs between the desired uncertainty of the information (e.g., as a function of the distance of a given moving object from a given target) and the size of data kept in the cache, methodologies have been devised which implement various policies for keeping/purging data items in/out a given cache (Cherniack et al. 2003).

Meteorology and Seismology

Many of the phenomena of interest in meteorology are entities which move and even change their shape over time (e.g., clouds, flood regions). Clearly, one cannot exactly model their shapes and bounding regions/volumes and for the purpose of query processing and any kind of reasoning involved, the uncertainty must be incorporated as part of the representation and the evolution of the objects (Reiners 2003). Furthermore, the probabilistic values due to uncertainty must be properly taken into consideration if the system, which monitors the phenomena of interest, is expected to exhibit some form of reactive behavior (Pfoser et al. 2005; Trajcevski et al. 2006). These issues are extremely important when, based on the changes of the monitored values (e.g., the coastal erosion co-related with seismic measurements in tsunami prediction; the CO-concentration co-related sulphur concentration and the temperature increases for predicting volcano's eruption) disaster-preventing alarms need to be issued with certain probabilistic guarantees (Stora et al. 1999).

Bio-chemistry

The efficient management of a vast body of observational data generated by expensive experiments is a paramount for researchers in biology and chemistry. As a particular example, the modelling of the information representing large sequences of metabolic pathways requires special tools to store, visualize and query such data. Due to the inherent properties of the micro-world, uncertainty is a natural parameter in such data sets. However, an important part of the research in biology and chemistry is related to modelling and reasoning about the reactions that may occur in various experiments. In such settings, the micro-mobility of the compounds which bind themselves in larger structures during the process of a given experiment cannot be specified in a crisp manner. Consequently, some researchers have recently incorporated the uncertainty when modelling the dynamics of the metabolic control (Wang et al. 2004).

Sensor Networks

The management of spatio-temporal data in sensor networks is a relatively new field with many open challenges and its natural settings simply cannot avoid the uncertainty. Namely, regardless of what kind of sensors are used for tracking mobile objects, the precision of determining their location is limited. Furthermore, due to the discrete coverage of a given geographic region of their deployment, determining the set of sensors that should process a particular query (e.g., the boundary of a given region for a range query) introduces yet another source of uncertainty (Buragohain et al. 2006). Yet another domain-specific problem that arises in these settings, and is a natural source for the imprecision of the data, is due to the fact that the quality of the readings of the sensor nodes are (an inverse) function of the mobile object's distance and, moreover, maintaining the identities of the individual objects (for the purpose of correct answer to the queries) is a challenge of its own (Zhao and Guibas 2004).

Spatio-temporal Data Reduction and Data Mining

The goal of any data reduction method is to decrease the size of the data-set of interest for a particular application. Clearly, there is a trade-off between the amount of data reduced and the level of the uncertainty introduced (with respect to the original data set). As demonstrated in Cao et al. (2006), unless proper caution is exercised when selecting the distance-function used in the reduction process, the errors obtained for the widely used class of spatio-temporal queries may become unbounded. Similar trade-offs, involving the uncertainty as part of the model, are present when, for various data mining purposes, one needs to cluster a set of trajectories representing the typical motions of moving objects along given routes and within given time-intervals and perform some similarity-based reasoning.

Future Directions

In general, any system which is targeted towards managing the *(location, time)* information pertaining to large amounts of mobile entities, is bound to incorporate uncertainty into its model and, as a consequence, in the processing algorithms for users' queries. One of the applications that poses a very intriguing challenge is the field of mobile data management in sensor network settings. In particular, the uncertainty present in these settings has a variety of sources, e.g., imprecision of the devices used (as a function of the object's distance), impossibility of the perfect coverage of the regions of interest, etc. However, this uncertainty can be exploited for the purpose of optimizing the in-network processing of various queries, in the sense that one can minimize the transmission of the individual sampling results when they are within the predefined bounds. This way, one can reduce the consumption of the most expensive resource - the energy of the individual nodes, which is mostly consumed when data transmission is required.

As pointed out in Pfoser et al. (2005), a thorough treatment of uncertainty, besides the adopted model used for its representation, needs a solid apparatus for executing the operations over the data. One of the challenges of managing the mobile object uncertainty is finding a proper fusion of probability theory, fuzzy-sets theory and other existing fields (e.g., computational geometry Buragohain et al. 2006) that will best serve the needs of a given application domain.

Cross-References

- ▶ Indexing Schemes for Multidimensional Moving Objects
- ▶ Privacy Threats in Location-Based Services
- ▶ Spatiotemporal Query Languages
- ▶ Uncertainty, Modeling with Spatial and Temporal

Acknowledgements Research Supported by the NorthropGrumman Corp. (P.O.8200082518) and the NSF (IIS-0325144).

References

Böhlen MH, Jensen CS (2003) Temporal data model and query language concepts. In: Bidgoli H (ed) Encyclopedia of information systems, vol 4. Academic, Amsterdam/Boston

Buragohain C, Gandhi S, Hershberger J, Suri S (2006) Contour approximation in sensor networks. In: DCOSS, San Francisco

Cao H, Wolfson O, Trajcevski G (2006) Spatio-temporal data reduction with deterministic error bounds. J Very Large Databases 15(3):211–228

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

Cherniack M, Galvez EF, Franklin M, Zdonik S (2003) Profiledriven cache management. In: ICDE, Bangalore

de Almeida VT, Güting RH (2005) Indexing the trajectories of moving objects in networks. GeoInformatica 9(1):33–60

Ding Z, Güting RH (2004a) Uncertainty management for network constrained moving objects. In: DEXA, Zaragoza

Ding Z, Güting RH (2004b) Managing moving objects on dynamic transportation networks. In: International conference on scientific and statistical database management (SSDBM), Santorini Island

Ding H, Trajcevski G, Scheuermann P (2007) Efficient maintenance of continuous queries for trajectories. GeoInformatica. doi:10.1007/s10707-007-0029-9

Gedik B, Liu L (2006) Mobieyes: a distributed location monitoring service using moving location queries. IEEE Trans Mobile Comput 5(10):1384–1402

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

Güting RH, Bohlen MH, Erwig M, Jensen CS, Lorentzos N, Nardeli E, Schneider M, Viqueira JRR (2003) Spatio-temporal models and languages: an approach based on data types. In: Spatio-temporal databases – the chorochronos approach

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

Mokbel MF, Xiong X, Aref WG (2004) Sina: scalable incremental processing of continuous queries in spatiotemporal databases. In: ACM SIGMOD International conference on management of data, Paris

Pelanis M, Saltenis S, Jensen CS (2006) Indexing the past, present, and anticipated future positions of moving objects. ACM Trans Database Syst 31(1):255–298

Pfoser D, Jensen C (1999) Capturing the uncertainty of moving objects representation. In: SSDB

Pfoser D, Tyfona N, Jensen C (2005) Indeterminacy and spatiotemporal data: basic definitions and case study. Geoinformatica 9(3):211–236

Pitoura E, Samaras G (2001) Locating objects in mobile computing. IEEE Trans Knowl Data Eng 13(4):571–592

Reiners WA (2003) Transport of energy, information and material through the biosphere. Annu Rev Environ Resour 28(1):107–136

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

Schiller J, Voisard A (2004) Location-based Services. Morgan Kaufmann, San Francisco

Sistla AP, Wolfson O, Chamberlain S, Dao S (1997) Modeling and querying moving objects. In: 13th international conference on data engineering (ICDE), Birmingham

Stora D, Agliati P, Cani M, Neyret R, Gascuel J (1999) Animating lava flows. In: Graphics interfaces, Ontario

Tao Y, Papadias D, Sun J (2003) The tpr*-tree: an optimized spatiotemporal access method for predictive queries. In: VLDB, Berlin

Trajcevski G (2003) Probabilistic range queries in moving objects databases with uncertainty. In: MobiDE, San Diego

Trajcevski G, Wolfson O, Hinrichs K, Chamberlain S (2004) Managing uncertainty in moving objects databases. ACM Trans Database Syst 29(3):463–507

Trajcevski G, Scheuermann P, Ghica O, Hinze A, Voisard A (2006) Evolving triggers for dynamic environments. In: Extending database technology (EDBT), Munich

Wang L, Birol I, Hatzimanikatis V (2004) Metabolic control analysis under uncertainty: framework development and case studies. Biophys J 87(6):3750–3763

Wolfson O, Sistla AP, Chamberlain S, Yesha Y: Updating and querying databases that track mobile units. Distrib Parallel Databases 7:257–387 (1999)

Xing X, Aref W (2006) R-trees with update memos. In: ICDE, Atlanta

Zhao F, Guibas L (2004) Wireless sensor networks: an information processing approach. Morgan Kaufmann, San Francisco

Moving Objects

- ▶ Epidemics Modeling by Spatiotemporal Constraint Data
- ▶ Indexing of Moving Objects, B^x-Tree
- ▶ Indexing Schemes for Multidimensional Moving Objects
- ▶ Indexing Spatiotemporal Archives
- ▶ Spatial Predictive Query Processing on Road Networks

Moving Objects Database

- ▶ Mobile Objects Databases

Moving Points

- ▶ Constraint Databases and Moving Objects
- ▶ Trajectory

Moving Queries

▶ Continuous Queries in Spatio-Temporal Databases

Moving Regions

▶ Constraint Databases and Moving Objects

MPI in GIS

Eric Shook
Department of Geography, Environment, and
Society, University of Minnesota, MN, USA

Synonyms

Message passing interface

Definition

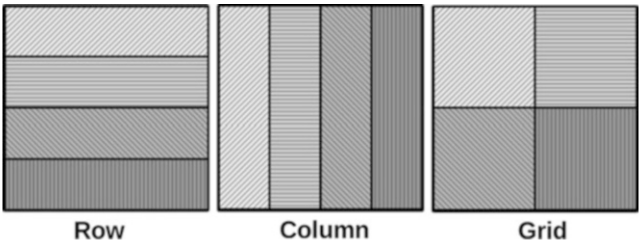
Message passing is an inter-process communication approach that enables processes executing on one or more computing nodes to exchange data. A message-passing interface now commonly referred to as MPI is the defacto standard for implementing message passing in parallel and high-performance computing (HPC) (Snir et al. 1998). MPI has been used to help parallelize multiple applications in the areas of geographic information systems (GIS), spatial analysis, and spatial modeling, which often decompose the spatial domain of a problem into subdomains and assign each processing core one or more subdomains to process simultaneously. Many parallel applications can be developed using only six MPI operations, which can be further enhanced by using the hundreds of more advanced operations. Geospatial libraries are beginning to exploit advanced functionality available in MPI to create parallel applications that scale to a large number of processing cores and are capable of processing massive amounts of spatial big data.

Historical Background

MPI was established in 1994 following years of research in the field of parallel and high-performance computing (HPC). It is a collaborative standard that was designed by many HPC experts and scholars in industry, national laboratories, and universities. MPI has hundreds of operations, but many parallel application can be written using six to ten operations. Since MPI is not a single software product but rather a standardized interface multiple implementations of MPI exist. In fact, many supercomputers and high-performance computing clusters install multiple implementations, because each implementation may offer different performance characteristics for parallel applications. Geospatial applications increasingly use MPI for parallel spatial data processing, because it is widely deployed helping to make these applications portable across HPC systems. The importance of parallel processing for geospatial applications has been acknowledged since at least the mid-1980s with relatively slow adoption in the early 1990s (Armstrong 2000) but is now a rapidly growing research area with including the establishment of a next-generation GIS based on cyberinfrastructure known as cyberGIS (Wang 2010).

Scientific Fundamentals

Parallel geospatial applications have been discussed for over almost 30 years (Armstrong 2000) providing a rich suite of parallel algorithms and approaches that can be exploited to improve computational performance for geographic information systems (GIS), spatial analysis, and spatial models (Healey et al. 1996). Although not all spatial problems are computationally tractable (or solvable) (Clarke 2003), parallel computation enables applications to leverage computational capabilities of HPC resources by decomposing a spatial problem into multiple subproblems that can be solved simultaneously. Oftentimes these problems are decomposed spatially known as spatial domain decomposition (Ding and Densham 1996). Row, column, and grid decomposition are common examples of spatial domain decomposition (Fig. 1). A spatial domain could be decomposed into equal or differently sized subdomains based on the computational intensity or amount of



MPI in GIS, Fig. 1 Illustration of three common spatial domain decompositions, namely, row, column, and grid. The spatial domain is decomposed into four equal subdomains, which could be distributed to four processes

computation associated with each subdomain to balance the computational workload among processes (Ding and Densham 1996; Wang and Armstrong 2009). MPI can be used to coordinate the parallel processing of subdomains among multiple processes by enabling processes to exchange data and information.

MPI uses a distributed memory approach where each process has a private memory that cannot be accessed by any other processes and uses messages that are passed between processes to exchange data. An alternate interprocess communication approach is called shared memory. Shared memory allows simultaneous access to a single memory that is shared among multiple processes. Shared memory applications can only leverage processing cores that have a shared memory space, which can often limit them to a single computing node of 4–32 cores. Since MPI applications do not depend on shared memory, MPI applications can execute on multiple computing nodes that each have their own distributed memory increasing both the number of cores that can be exploited and the total amount of memory that is available to the application. Parallel geospatial applications that use MPI can scale to thousands of processing cores and have access to terabytes of memory (Shook et al. 2013).

MPI specifies a programming interface for a message-passing model. It is not a specific implementation or software product, and many implementations exist in multiple languages so this entry will not focus any particular language or implementation of MPI. While MPI applications have scaled to hundreds of thousands of processing cores on some of the most powerful supercomputers in the world, MPI is not appropriate for all GIS applications. If an application does not need parallelism or parallel libraries already exist that the application can readily use, then MPI is likely not appropriate. If, however, an application has a need for: (1) Parallelism; (2) Portability so the application that can be easily ported between computational systems; (3) High-performance and scalability such that the application can execute on potentially thousands of processing cores, then MPI may be appropriate. This entry covers basic MPI operations and discusses advanced operations that can be used for geospatial applications.

Overview of Basic Operations

MPI provides a standard interface for passing messages between processes. A message is simply data often either (1) an integer value (e.g., 13, 0, or −5), (2) a floating point value (e.g., 7.8, 3.1415, −1.234), or (3) an array of integer or floating point values, although many data types are supported and developers can create custom MPI data types. MPI operations can be classified as point to point or collective based on how many processes are sending messages and how many processes are receiving messages. This entry will briefly overview basic operations of MPI that are commonly used and serve as a foundation for more advanced operations. These operations represent two-way communication, meaning that for each sender there must be a receiver. For brevity, each MPI operation is discussed in the context of the simplest and most common use cases. Due to the flexibility of MPI, these operations can also be used in many complex contexts, which overly complicate the discussion and detract from the purpose of introducing these operations. The reader is referred to Gropp et al. (1999), Snir et al. (1998), and Wilkinson and Allen (1999) for a comprehensive discussion of these operations and how they can be applied in more complex contexts.

Initialization and Finalization

The following four operations setup and tear down the MPI environment enabling each process to communicate with all other processes. Combining these four operations with MPI_Send and MPI_Recv operations (see next section) allows many parallel applications to be written.

MPI_Init

MPI_Init will setup the MPI environment. This is the first MPI *operation* called in an MPI *application*, because MPI operations will not function properly before the environment has been setup. Different implementations will setup different environments for different parallel systems to support message passing, but these technical details are abstracted from developers through the use of MPI.

MPI_Comm_rank

MPI_Comm_rank returns a unique identifying number referred to as a rank for each process, which is used to route messages from process to process. All processes in an MPI application start simultaneously, so processes use this operation to identify themselves (i.e., get their rank).

MPI_Comm_size

MPI_Comm_size returns the number of processes executing an MPI application. This number is identical for all processes. The "size" of an executing MPI application may influence how a spatial problem is decomposed and distributed among all processes for parallel processing.

MPI_Finalize

MPI_Finalize will tear down the MPI environment. This is the last MPI operation called in an MPI application, because MPI operations will not function properly after the environment has been torn down. It is critical to ensure that *all* processes are finished communicating before calling MPI_Finalize. A common bug in MPI applications occurs when one process – that finishes early – calls MPI_Finalize that tears down the MPI environment, but other processes have not finished communicating causing an application to crash or produce unexpected results. It is common to use an MPI_Barrier (see description below) to ensure all processes are finished communicating before calling MPI_Finalize.

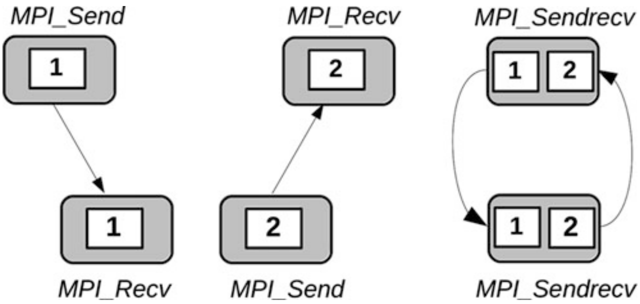
Point-to-Point Operations

MPI_Send

MPI_Send is a point-to-point operation that sends a message from one process to another process (Fig. 2). MPI defines multiple modes for sending messages: *Buffered mode* may buffer a message in memory before sending. *Ready mode* assumes a matching receive has already been called and the send can begin immediately. *Synchronous mode* will not send a message until a matching receive is called. MPI_Send can use any of these modes, and it is up to the specific implementation to select the best mode. Alternatively, developers can explicitly use one of the send modes by calling MPI_Bsend, MIP_Rsend, or MPI_Ssend, respectively.

MPI_Recv

MPI_Recv is a point-to-point operation that receives a message sent by another process. MPI_Recv can receive a message sent using any send mode including synchronous mode, buffered mode, and ready mode. Every MPI_Send is paired with an MPI_Recv. Specifically, if process N uses MPI_Send to send message to process M, then process M must call MPI_Recv to receive the message. If a message is sent and no process receives the message, then it can cause deadlock, which occurs when two or more processes are waiting for the other to finish a task resulting in none of them finishing a task. This operation can receive a message from (1) any process or (2) a specific process identified by MPI rank. While receiving a message from



MPI in GIS, Fig. 2 Illustration of point-to-point operations including send and receive operations (*left*) and a combined send and receive operation (*right*)

any process provides the most flexibility, it can also be difficult to troubleshoot bugs in a parallel code, because a process may be receiving the wrong message from the wrong process leading to unexpected results or deadlock.

MPI_Sendrecv

MPI_Sendrecv is a point-to-point operation that combines sending and receiving a message into a single operation. This operation helps to make MPI application code more explicit by stating that a process will send *and* receive a message (Fig. 2). Further, this operation may help avoid deadlock. Take, for instance, the case where N processes are organized in a ring and each process sends a message to the process to their "right" (e.g., process 0 sends to process 1, process 1 sends to process 2, …, process $(N - 1)$ sends to process 0). This case will result in deadlock, because all processes are sending a message and no processes are receiving any messages. While there are multiple solutions to eliminate deadlock in this case, using MPI_Sendrecv is one of the easiest by combining send and receive.

Collective Operations

Collective operations facilitate exchanging messages between more than two processes. Figure 3 illustrates four of the most commonly used collective operations, which are described below.

MPI_Bcast

MPI_Bcast is a collective operation that broadcasts a message from one process to all other processes. Recall that processes do not share memory so MPI_Bcast serves as a means to share data with all processes. This operation is much more efficient than sending $N - 1$ messages using MPI_Send even though they are functionally equivalent, because this operation uses algorithmic optimizations that coordinate multiple processes to help broadcast the message to the rest of the processes.

MPI_Scatter

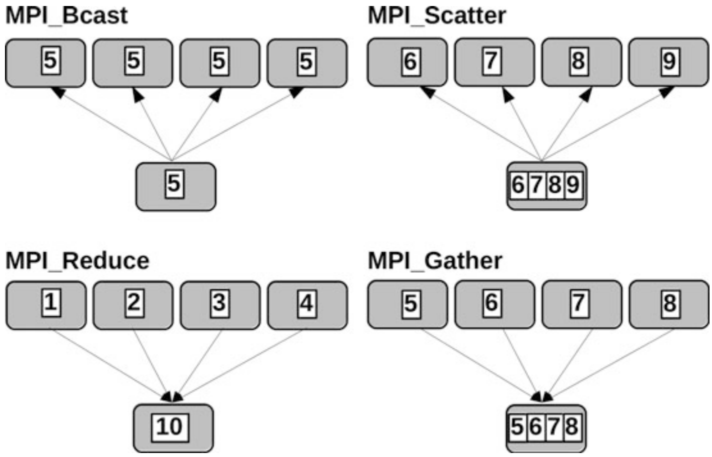
MPI_Scatter is a collective operation in which one process sends a *different* message to each other process. This operation offers more flexibility compared to broadcast, which sends only one message to all processes.

MPI_Gather

MPI_Gather is a collective operation in which one process gathers messages sent from each other process. This operation is the reverse of MPI_Scatter. A gather operation may be used to collect the number of spatial features (e.g., points, polygons, or raster cells) that is associated with each process following a spatial domain decomposition, for example.

MPI_Reduce

MPI_Reduce is a collective operation that gathers a value from all processes and reduces them to a single value. Unlike other operations, this operation combines basic computation to reduce multiple values to a single value. Built-in MPI operations include sum (MPI_SUM), product (MPI_PROD), minimum (MPI_MIN), and maximum (MPI_MAX) of the values, and users can create their own custom operation to reduce values. To calculate the total number of spatial features being processed in a parallel application, for example, first each process counts



MPI in GIS, Fig. 3 Illustration of collective operations including broadcast (*upper left*), reduce (*lower left*), scatter (*upper right*), gather (*lower right*)

the number of local spatial features and saves it to a localsum value. The localsum value is then passed to MPI_Reduce with the MPI_SUM operation that adds the localsum value from each process to a global sum of all spatial features.

MPI_Barrier

MPI_Barrier is a collective operation that does not explicitly share data. Instead, it will cause each process that calls MPI_Barrier to wait until all other processes have called MPI_Barrier before continuing. This is commonly referred to as synchronization, because after a barrier all processes are synchronized to a single point in an application. Although barriers are rarely required for well-designed parallel applications, this operation is commonly called at the end of a time step or iteration in modeling applications or the end of a complex parallel calculation in GIS and spatial analysis applications. It is also commonly called before an MPI_Finalize to ensure all processes are finished communicating before tearing down the MPI environment.

Advanced Operations

Non-blocking Operations

MPI also offers multiple "non-blocking" operations that do not wait for the operation to finish before returning control back to the application. Non-blocking operations are advantageous, because applications are able to continue processing while the communication operation finishes in the background. However, the application is required to check that the operation was successfully completed later. Generally, non-blocking operations improve performance by improving communication-computation overlap (e.g., computing while communicating) but result in more complex parallel applications.

Advanced Collective Operations

Many collective operations are available (Snir et al. 1998). Many of these operations essentially combine multiple collective operations into a single operation or handle special cases with respect to data. MPI_Allgather, for example, is a collective operation that in essence combines a gather and broadcast. A brute-force implementation of Allgather would first gather values from all processes to a single process, which would then broadcast the values to all other processes. However, algorithmic optimizations can improve performance when these operations are combined compared to when they are separate. MPI_Alltoall is collective operation that in essence combines $N - 1$ send operations and $N - 1$ receive operations. This operation enables each process to send a message to and receive a message from each other process.

Parallel I/O Operations

MPI version 2 defined parallel input and output (MPI-IO) in which multiple processes may read or write to a file in parallel. MPI-IO can significantly enhance IO performance for parallel applications. Using MPI-IO operations, an application can manipulate files in parallel (e.g., MPI_File_open, MPI_File_read, MPI_File_write, and MPI_File_close). Importantly, applications can establish what are known as file views that control which portions of a file each process is able to read from and write to. File views can be defined based on a spatial domain decomposition (e.g., row, column, or grid decomposition) providing a powerful way for parallel processes to manipulate spatial data. Recent research has also looked to incorporate MPI and parallel IO into widely used libraries such as the Geospatial Data Abstraction Library (GDAL) (Qin et al. 2013; Guan et al. 2014).

Key Applications

GIS and Spatial Analysis

GIS and spatial analysis operations are often computationally intensive especially with respect to growing spatial data sizes. MPI provides a standard interface to parallelize GIS and spatial analysis operations, which can then be ported from a desktop PC to a cluster or grid-computing infrastructure. Multiple parallel algorithms and approaches have been established in the literature, which can readily make use of MPI (Healey et al. 1996).

CyberGIS

CyberGIS integrates cyberinfrastructure, GIS, and spatial analysis for geographic problem-solving (Wang 2010). Applications and tools written for cyberGIS can leverage MPI to improve scalability and speed up processing of spatial big data.

Spatial Modeling

Spatial models including cellular automata and agent-based models (ABMs) use MPI to facilitate parallelization. Frameworks and computational systems in this area often aim to simplify development and hide many of the complexities of communication while supporting advanced functionality. Communication frameworks for ABMs have scaled to thousands of processing cores using MPI (Shook et al. 2013).

Future Directions

Just as MPI is an interface and not an implementation, GIS research would benefit from establishing a standard parallel GIS interface. An MPI-based GIS interface that supports common GIS functionality will help reduce development time for parallel GIS applications and serve as a platform to drive future parallel GIS research. Just as MPI was built on the collective research and toolkits at the time, so too a parallel GIS interface could be built upon the many libraries, frameworks, and systems that exist for parallel GIS, cyberGIS, parallel ABMs, and parallel cellular automata (Wang 2010; Yang et al. 2010; Shook et al. 2013; Guan et al. 2014; Qin et al. 2014).

Parallel IO is a growing computational bottleneck for parallel geospatial applications. Many geospatial applications are data intensive, and without sufficient ability to read and write large amounts of spatial data, parallel geospatial applications will be limited in fully exploiting HPC resources. While current research is beginning to address this issue (Qin et al. 2013), there is still insufficient research in this area to handle the diversity of spatial data that is used in GIS.

References

Armstrong MP (2000) Geography and computational science. Ann Assoc Am Geogr 90(1):146–156

Clarke KC (2003) Geocomputations future at the extremes: high performance computing and nanoclients. Parallel Comput 29(10):1281–1295

Ding Y, Densham PJ (1996) Spatial strategies for parallel spatial modelling. Int J Geogr Inf Syst 10(6):669–698

Gropp W, Lusk E, Skjellum A (1999) Using MPI: portable parallel programming with the message-passing interface. MIT Press, Cambridge

Guan Q, Zeng W, Gong J, Yun S (2014) pRPL 2.0: improving the parallel raster processing library. Trans GIS 18(S1):25–52

Healey RG, Dowers S, Minetar MJ (eds) (1996) Parallel processing and GIS. Taylor & Francis, Inc., Bristol

Qin C, Zhan L, Zhu A et al (2013) How to apply the geospatial data abstraction library (GDAL) properly to parallel geospatial raster I/O? Trans GIS 18(6):950-957

Qin C, Zhan L, Zhu A, Zhou C (2014) A strategy for raster-based geocomputation under different parallel computing platforms. Int J Geogr Inf Sci 28(11): 2127–2144

Shook E, Wang S, Tang W (2013) A communication-aware framework for parallel spatially explicit agent-based models. Int J Geogr Inf Sci 27(11):2160–2181

Snir M, Otto SW, Walker DW, Dongarra J, Huss-Lederman S (1998) MPI: the complete reference. MIT Press, Cambridge

Wang S (2010) A cyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. Ann Assoc Am Geogr 100(3):535–557

Wang S, Armstrong MP (2009) A theoretical approach to the use of cyberinfrastructure in geographical analysis. Int J Geogr Inf Sci 23(2):169–193

Wilkinson B, Allen M (1999) Parallel programming. Prentice Hall, Upper Saddle River, New Jersey

Yang C, Raskin R, Goodchild M, Gahegan M (2010) Geospatial cyberinfrastructure: past, present and future. Comput Environ Urban Syst 34(4):264–277

MRA-Tree

▸ Multi-resolution Aggregate Tree

MRV

▸ Climate Risk Analysis for Financial Institutions

M-Tree

▸ Indexing, High Dimensional

Multi Agent Systems

▸ Wayfinding: Affordances and Agent Simulation

Multicriteria Decision-Making, Spatial

Salem Chakhar and Vincent Mousseau
LAMSADE, University of Paris Dauphine,
Paris, France

Synonyms

Analysis, robustness; Analysis, sensitivity; Decision-making, multi-attribute; Decision-making, multi-criteria; Decision-making, multi-objective; Decision rules; GIS-based multicriteria decision analysis; Mathematical programming; Preference structure; Spatial multicriteria decision aid

Definition

Multicriteria analysis is generally defined as "*a decision-aid and a mathematical tool allowing the comparison of different alternatives or scenarios according to many criteria, often conflicting, in order to guide the decision maker toward a judicious choice*" (Roy 1996). The set of decision alternatives considered in a given problem is often denoted by A and called the *set of potential alternatives*. A *criterion* is a function g, defined on A, taking its values in an ordered set and representing the decision maker's preferences according to some points of view. The evaluation of an alternative a according to criterion g is written $g(a)$.

Spatial multicriteria decision making refers to the application of multicriteria analysis in spatial contexts where alternatives, criteria and other elements of the decision problem have explicit spatial dimensions. Since the late 1980s, multicriteria analysis has been coupled with *geographical information systems* (GIS) to enhance spatial multicriteria decision making.

Historical Background

It is generally assumed that multicriteria analysis was born and took its actual vocabulary and form at the beginning of 1960s. In fact, most multicriteria analysis practitioners consider that their field stems largely from the research of Simon on satisficing and the early works on goal programming. Closely related to decision-making in general and to multicriteria analysis in particular is utility theory. Although utility theory was first used to model simple individual preferences, it has been extended to multicriteria preferences and led to the *multiattribute utility theory* (Keeney and Raffia 1976).

The first methods in multicriteria analysis were developed during the 1960s. Goal programming, for example, uses linear programming method to resolve a multicriteria problem. In 1968, Roy conceived the initial version of the ELECTRE method (see Figueira et al. (2005)).

Throughout the 1970s, the widely dispersed scientific field of multicriteria analysis started to take form. First, in 1971 Roy organized the first independent session specifically devoted to multicriteria research within the 7th Mathematical Programming Symposium, held in The Hague. Second, in 1972 Cochrane and Zeleny organized the First International Conference on multicriteria decision making at the University of South Carolina. Then in 1975, Roy organized the first meeting of the EURO Working Group on Multi-Criteria Decision Aid in Brussels. Also in 1975, Thiriez and Zionts organized the First Conference of the International Society on multicriteria analysis. In addition to these first scientific meetings, multicriteria analysis research focused in the 1970s on the theoretical foundations of *multiobjective decision making*.

The 1980s and 1990s witnessed the consolidation and development of a great number of interactive methods. Most of these methods are oriented toward negotiation or multiple decision makers and multicriteria decision support systems.

Multicriteria analysis has been used since its emergence to deal with spatial decision problems. The first works involving GIS-based multicriteria analysis where published in the late 1980s and the early 1990s. Currently, there are a number of relatively important articles devoted to GIS-based multicriteria analysis that have been published (Malczewski 2006).

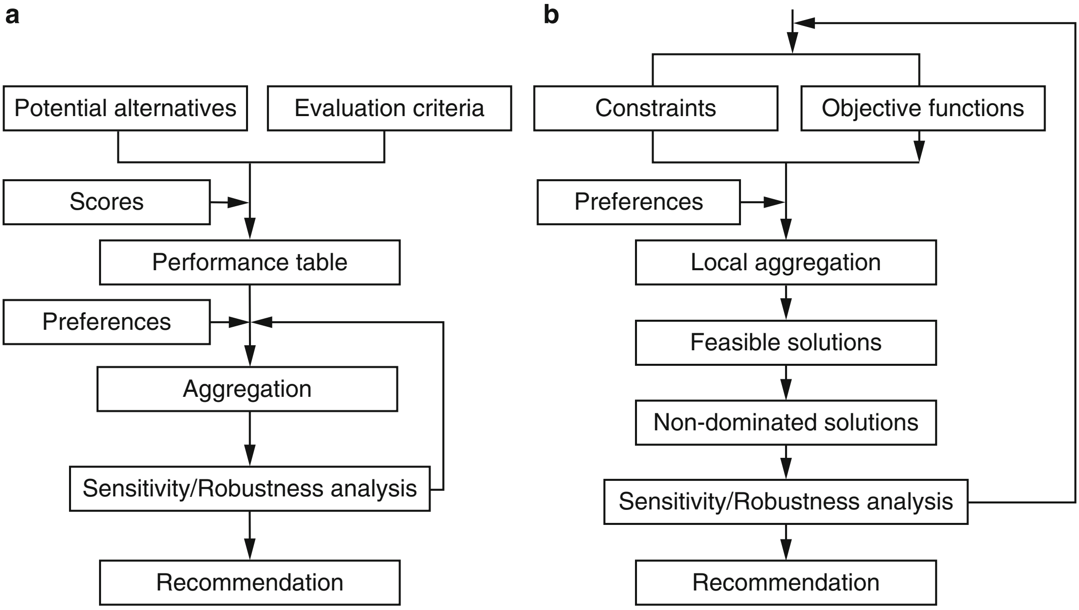
Scientific Fundamentals

General Schema of Multicriteria Analysis Methods

Different multicriteria analysis methods are available in the literature (Figueira et al. 2005). An excellent online bibliography of multicriteria analysis and its applications is available at http://www.lamsade.dauphine.fr/mcda/biblio/. Multicriteria methods are commonly categorized as *discrete* or *continuous*, depending on the domain of alternatives. The former deals with a discrete, usually limited, number of prespecified alternatives. The latter deals with variable decision values to be determined in a continuous or integer domain of infinite or large number of choices. Several authors classify them as (i) *multiple attribute decision-making* (MADM), and (ii) *multiple objective decision-making* (MODM). In this presentation, the discrete/continuous classification is chosen since it is in accordance with the conventional representation of data in GIS (vector vs. raster) and it is more general than the MADM/MODM classification. Figure 1 gives the general schema of discrete and continuous multicriteria methods that will be briefly described in the following two paragraphs.

Discrete Methods

The first requirement of nearly all discrete techniques is a *performance table* containing the evaluations or *criteria scores* of a set of alternatives on the basis of a set of criteria. The next step consists of the aggregation of the different criteria scores using a specific *decision rule* (or *aggregation procedure*). It takes into account the *decision maker's preferences*, generally represented in terms of *weights* that are assigned to different criteria. The aggregation of criteria scores permits the decision maker to make a comparison between the different alternatives on the basis of these scores. The aggregation procedures represent the identities of the multicriteria analysis techniques. The discrete methods are usually categorized based on their aggregation procedures into two different families:



Multicriteria Decision-Making, Spatial, Fig. 1 General schema of discrete (**a**) and continuous (**b**) multicriteria methods

(1) *outranking relation-based decision rules*, and (2) *utility function-based decision rules*.

The uncertainty and fuzziness generally associated with any decision situation require a *sensitivity/robustness analysis* enabling the decision maker(s) to test the consistency of a given decision or its variation in response to any modification in the input data and/or in the decision maker preferences.

Continuous Methods

The starting point of most continuous methods are a set of *constraints* and *objective functions*. The former set contains inequalities which reflect natural or artificial restrictions on the values of the input data. This means that *feasible solutions* are *implicitly* defined in terms of these constraints.

For continuous methods, the decision maker's preferences generally take the form of *weights* that are assigned to different objective functions. They may also be represented as *target values* that should be satisfied with any feasible solution. The decision maker should also indicate, for each objective function, its *direction of optimization*, that is maximization or minimization. No other information than the weights and these directions of optimization are required to define the set of *non-dominated solutions*. This set contains solutions that are not dominated by any other one.

Generally, *local* and *interactive* aggregation algorithms are used to define the feasible solutions set. This permits the combination of the decision maker preferences and the computer to solve the decision problem, using methods that alternate calculation steps and dialogue steps. In reality, the local and interactive algorithms require the decision maker preferences to be expressed *progressively* throughout the resolution process. The decision maker preferences, however, may be expressed a priori (i.e., before the resolution process) or a posteriori (i.e., after the resolution process).

In many practical situations, the decision maker is called upon to relax some of its constraints in order to guarantee that the set of feasible solutions is not empty or, simply, to test the stability of the results.

Spatial Multicriteria Decision Making

A brief description of spatial multicriteria decision making concepts is provided in the following. In the rest of this entry, $F = \{1, 2, \cdots, m\}$ denotes the set of the indices of m evaluation criteria $g_1, g_2, \cdots, g_m$. Accordingly, $g_j(j \in F)$ is the evaluation criterion number j.

Spatial Decision Alternatives

Decision alternatives can be defined as alternative courses of action among which the decision maker must choose. A spatial decision alternative consists of at least two elements (Malczewski 1999): *action* (what to do?) and *location* (where to do it?). The spatial component of a decision alternative can be specified *explicitly* or *implicitly* (Malczewski 2006). The second case holds when there is a spatial implication associated with implementing an alternative decision.

The set of spatial decision alternatives may be discrete or continuous. In the first case, the problem involves a discrete set of pre-defined decision alternatives. Spatial alternatives are then modeled through one or a combination of the basic spatial primitives, namely point, line, or polygon. The second case corresponds to a high or infinite number of decision alternatives, often defined in terms of constraints. For practical reasons, the set of potential alternatives is often represented in a "discretized" form where each raster represents an alternative. Alternatives may be constructed as a collection of rasters.

Evaluation Criteria

In the spatial context, evaluation criteria are associated with geographical entities and relationships between entities, and can be represented in the form of maps. One should distinguish a simple map layer from a *criterion map*. In fact, a criterion map models the preferences of the decision maker concerning a particular concept, while a simple map layer is a representation of some spatial real data. A criterion map represents subjective preferential information. Two different persons may assign different values to the same mapping unit in a criterion map.

Constraints

A *constraint* (or *admissibility criterion*) represents natural or artificial restrictions on the potential alternatives. Constraints are often used in the pre-analysis steps to divide alternatives into two categories: "*acceptable*" or "*unacceptable*". An alternative is acceptable if its performance on one or several criteria exceeds a minimum or does not exceed a maximum.

In practice, constraints are often modeled through elementary multicriteria methods like the *conjunctive* or *disjunctive* aggregation procedures. With the conjunctive method, a *minimal satisfaction level* $\hat{g}_j$ is associated with each criterion g_j. If the performance of an alternative with respect to different criteria is equal or better to these minimal satisfaction levels (i.e., $g_j(a_i) \geq \hat{g}_j, \forall j \in F$), the alternative is considered as acceptable. Otherwise, the alternative is considered as unacceptable. With the disjunctive method, the alternative is considered acceptable as soon as at least one satisfaction level is exceeded.

Quantification

The evaluation of alternatives may be quantitative or qualitative. Several methods require quantitative evaluations. In the literature, there exist some totally qualitative methods such as the median ranking method. Other methods, such as the ELECTRE family of methods (see Figueira et al. (2005)), involve both types of evaluations. When most of the criteria are qualitative, quantitative criteria may be converted into qualitative ones and a qualitative method used. Otherwise, a *quantification method* (i.e., assignment of numeric values to qualitative data) is applied; the *scaling approach* is the one most used.

Application of a quantification method requires the definition of a measurement scale. The most used measurement scale is the *Likert-type*. This scale is composed of approximatively the same number of favorable and unfavorable levels. An example with five levels is: *very unfavorable*, *unfavorable*, *neutre*, *favorable*, *very favorable*. Other more detailed measurement scales may also be used. The quantification procedure consists of constructing a measurement scale like the one with five points mentioned above. Then, numerical values are associated with each level of the scale. For instance, the numbers 1, 2, 3, 4 or 5 may be associated with the five-point scale from *very unfavorable* to *very favorable*.

Standardization

The evaluation of alternatives may be expressed according to different scales (ordinal, interval, ratio). However, a large number of multicriteria methods (including practically all the utility function-based methods) require that all the criteria are expressed in a similar scale. Standardizing the criteria permits the rescaling of all the evaluation dimensions between 0 and 1. This allows between and within criteria comparisons.

There are a large number of standardization procedures. In all procedures, standardization starts from an initial vector $(g_j(a_1), g_j(a_2), \cdots, g_j(a_m))$ to obtain a standardized vector $(r_{1j}, r_{2j}, \cdots, r_{mj})$ with $0 \leq r_{ij} \leq 1$; $\forall j \in F$ and $i = 1, \cdots, n$ (n is the number of alternatives). The most used standardization procedure in GIS-based multicriteria decision making is the *linear transformation procedure*. It associates with each alternative a_i and for each criterion g_j the percentage of the maximum over all alternatives:

$$r_{ij} = \frac{g_j(a_i)}{\max_i g_j(a_i)}, \quad i = 1, \ldots, n; \quad j \in F.$$

Pre-analysis of Dominance

In the absence of any preferential information, the only possible operation on the performance table is to eliminate the dominated alternatives. Let a and b be two alternatives from A. The alternative a *dominates* the alternative b in respect to F, noted as $a\Delta b$, if and only if:

$$g_j(a) \geq g_j(b); \quad j \in F,$$

with at least one strict inequality. Then, an alternative a from A is said to be *efficient*, *admissible* or *Pareto optimal* if and only if there is no other alternative b in A such that: $b\Delta a$.

Criteria Weights

Generally, in multicriteria problems the decision maker considers one criterion to be more important than another. This *relative importance* is usually expressed in terms of numbers, often called *weights*, which are assigned to different criteria. These weights deeply influence the final choice and may lead to a non-applicable decision mainly when the interpretations of such weights are misunderstood by the decision maker.

In the literature, many direct weighting techniques have been proposed. When a *simple arrangement technique* is used, the decision maker sets the criteria in an order of preference. The *cardinal simple arrangement technique* involves each criterion being evaluated according to a pre-established scale. Other indirect methods are also available such as the *interactive estimation method*. There are also relatively complex weight assignment techniques such as the *indifference trade-offs* technique (Keeney and Raffia 1976) and the *analytic hierarchy process* (AHP) (Saaty 1980).

Preference Structure and Preference Parameters

When comparing two alternatives a and b, the decision maker will generally have one of the three following reactions: (i) preference for one of the two alternatives, (ii) indifference between the two alternatives or (iii) impossibility to compare the alternatives. These situations are generally denoted as follows: (i) *aPb* if a is preferred to b (*bPa* if it is the opposite), (ii) *aIb* if there is indifference between a and b, and (iii) *aRb* if there is an incomparability. The binary relations of *preference* P, *indifference* I, and *incomparability* R are respectively the sets of tuples (a, b) such that *aPb*, *aIb*, *aRb*. It is generally admitted that I is reflexive and symmetric, P is asymmetric, and R is irreflexive and symmetric. The three relations (I, P, R) constitute a *structure of preference* over A if and only if they have the properties mentioned above and only one of the following situations holds (Vincke 1992): *aPb*, *bPa*, *aIb*, *aRb*.

Preference models require the definition of one or several thresholds, called *preference parameters*. The most commonly used preference parameters are the *indifference*, *preference* and *veto* thresholds. These three parameters are used essentially within the outranking relation-based decision rules. The first two parameters are for modeling imprecision and uncertainty in the decision maker's preferences. The latter is often used to compute the *discordance index*.

Decision Rules

To compare alternatives in A, it is necessary to aggregate the *partial evaluations* (i.e., with respect to each criterion) into a global one by using a given *decision rule* (or *aggregation procedure*). As mentioned earlier, within the discrete family, there are usually two aggregation approaches: (i) *utility function-based approach*, and (ii) *outranking relation-based approach*. The basic principle of the first family is that the decision maker looks to maximize a utility function $U(a) = U(g_1(a), g_2(a), \cdots, g_m(a))$, aggregating the partial evaluations of each alternative into a global one. The simplest and most often used utility function has an additive form: $U(a) = \Sigma_{j \in F} u_j(g_j(a))$; where u_j are the partial utility functions. Within this form, the preference P and indifference I binary relations are defined for two alternatives a and b as follows:

$$aPb \Leftrightarrow U(a) > U(b) \quad \text{and}$$
$$aIb \Leftrightarrow U(a) = U(b)\,.$$

In contrast with the first family, the second one uses *partial aggregation procedures*. Different criteria are aggregated into a partial binary relation S, with *aSb* used to indicate that "*a is at least as good as b*". The binary relation S is called an *outranking relation*. The most well known method in this family is ELECTRE (see, e.g., (Figueira et al. 2005)). To construct the outranking relation S, for each pair of alternatives (a, b), a *concordance index* $C(a, b) \in [0,1]$ – measuring the power of criteria that are in favor of the assertion *aSb* – and a *discordance index* $ND(a, b) \in [0,1]$ – measuring the power of criteria that are opposed to *aSb* – are computed. Then, the relation S is defined as follows:

$$\begin{cases} C(a,b) \geq \hat{c} \\ ND(a,b) < q\hat{d} \end{cases}$$

where $\hat{c}$ and $\hat{d}$ are the *concordance* and the *discordance thresholds*, respectively. Often an exploitation phase is needed to extract information from S on how alternatives compare to each other. At this phase, the concordance $C(a, b)$ and discordance $ND(a, b)$ indices are used to construct an index $\sigma(a, b) \in [0,1]$, representing the *credibility* of the proposition *aSb*, $\forall(a, b) \in A \times A$. The proposition *aSb* holds if $\sigma(a, b)$ is greater or equal to a given *cutting level*, $\lambda \in [0.5,1]$.

In the continuous formulation of a multicriteria problem, decision rules implicitly define the set of alternatives in terms of a set of *objective functions* and a set of *constraints* imposed on the decision variables. Here, *multiobjective mathematical programming* is often used. A multiobjective mathematical program is a problem where the aim is to find a vector $\mathbf{x} \in \mathbf{R}^p$ satisfying constraints of type

$$h_i(\mathbf{x}) < q0\,; \quad (i = 1, 2, \ldots, n),$$

respecting eventual integrity conditions and optimizing the objective functions:

$$z_j(\mathbf{x}), \quad j = 1, 2, \ldots, m.$$

The general form of a multiobjective mathematical program is as follows:

$$\begin{cases} \text{Optimize} \quad [z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_m(\mathbf{x})] \\ h_i(\mathbf{x}) < q0 \quad (i = 1, \ldots, n) \\ \mathbf{x} \in X \end{cases}$$

A multiobjective mathematical program is in fact a multicriteria decision problem where (Vincke 1992): (i) $A = \{\mathbf{x} : h_i(\mathbf{x}) \leq 0, \forall i\} \subset \mathbf{R}^p$ is the set of decision alternatives and (ii) $F = \{z_1(\mathbf{x}), z_2(\mathbf{x}), \cdots, z_m(\mathbf{x})\}$ is a set of criteria where each criterion is expressed by an objective function in terms of the decision variables.

Sensitivity/Robustness Analysis

The analysts should examine, through *sensitivity analysis*, the stability of results with respect to the variation of different parameters. Sensitivity analysis is the basis for *robustness analysis*. There are several proposals to enhance GIS-based multicriteria decision making with sensitivity analysis procedures (e.g., Feick and Hall (2004)). Robustness analysis in multicriteria decision making is a relatively new research topic. Proposals for enhancing GIS-based multicriteria decision making with robustness analysis are still lacking.

Final Recommendation

The final recommendation in multicriteria analysis may take different forms according to the manner in which a problem is stated. Roy (1996) identifies four types of results corresponding to four ways for stating a problem: (i) *choice*: selecting a restricted set of alternatives, (ii) *sorting*: assigning alternatives to different predefined categories, (iii) *ranking*: classifying alternatives from best to worst with eventually equal positions or (iv) *description*: describing the alternatives and their follow-up results.

Key Applications

GIS-based multicriteria analysis is used in a wide range of decision and management situations. In a recent literature review, Malczewski (2006) enumerates about 319 papers devoted to GIS-based multicriteria analysis between 1990 and 2004. The complete list of these papers is available at http://publish.uwo.ca/~jmalczew/gis-mcda.htm.

Environment Planning and Ecology Management

GIS-multicriteria evaluation has been intensively used in environment planning and ecology management. Most analyses within this application area concern land suitability, resource allocation, plan/scenario evaluation, impact assessment and site search/selection problems.

Transportation

Within the transportation application domain, GIS-based multicriteria evaluation is used essentially in vehicle routing and scheduling, and land suitability problems.

Urban and Regional Planning

Major uses of GIS-multicriteria analysis in urban and regional planning concern resource allocation, plan/scenario evaluation, site search/selection and land suitability problems.

Waste Resource Management

The problems tackled in this application domain concern land suitability, plan/scenario evaluation and site search/selection.

Hydrology and Water Resources

In the hydrology and water resources application domain, GIS-multicriteria analysis is used essentially for plan/scenario evaluation. There are also some works for site search/selection and land suitability problems.

Forestry

Major problems tackled within the forestry application domain are land suitability, site search/selection and forestry resources allocation.

Agriculture

The problems considered here are essentially land suitability for different agricultural uses and resources allocation for agricultural activities. Some works are concerned with site search/selection and plan/scenario evaluation problems.

Natural Hazard Management

The problems considered within this application domain mainly concern land suitability and plan/scenario evaluation.

Recreation and Tourism Management

Within this application area, the most treated problem is site search/selection.

Health Care Resource Allocation

Major works in this application domain concern health care site search/selection.

Housing and Real Estate

The problems that are treated here concern land suitability for habitat and real estate, plan/scenario evaluation and site selection for habitation restoration.

Future Directions

There are many important proposals concerning GIS-based multicriteria spatial decision making. However, these proposals present some limitations that prevent them from going beyond the academic contexts. Some of these limitations are cited in the following section.

Integration of Utility-Based Decision Rules

A major part of GIS and multicriteria analysis integration works use utility-based decision rules. However, outranking relation-based decision rules are generally more appropriate to deal with ordinal aspects of spatial decision problems. The natural explanation for this is that the outranking relation-based decision rules have computational limitations with respect to the number of alternatives they consider (Marinoni 2006). One possible solution to facilitate the use of decision rules based outranking relation is to reduce the number of potential alternatives. The idea that is generally used consists of subdividing the study area into a set of homogenous zones which are then used as decision alternatives or as a basis for constructing these alternatives.

Spatial and Temporal Dimensions in Multicriteria Modeling

Two points need to be addressed here: the construction of criteria involving divergent consequences and the modeling of preferences that vary across time and space. In the literature, there are some papers that deal with the construction of criteria based on divergent consequences and the modeling of time-dependent preferences. With respect to GIS-based multicriteria analysis, there are a few papers that take these aspects into account (Feick and Hall 2004).

Fuzzy Spatial Multicriteria Decision Making

Malczewski (2006) estimates that 77 % of the papers that were published between 1990 and 2004 related to GIS multicriteria analysis used deterministic information. There are several plans to incorporate multicriteria methods supporting imprecision, uncertainty and fuzziness into GIS (Jiang and Esatman 2000). The integration of such methods in a geographical information system has the potential to enhance its analytical strength.

Multicriteria Group Spatial Decision Making

Spatial decision problems naturally involve several different kinds of stakeholders. However, the majority of the GIS-multicriteria articles consider individual decision maker's approaches and only a few works (e.g., Jankowski et al. (2001)) are devoted to multicriteria *group spatial decision making*.

Web-Based Multicriteria Spatial Decision Making

There is an increasing interest in the development of Web-based GIS multicriteria evaluation systems (Carver 1999). Research on this topic is worthwhile since it promotes the sharing and access of geographical information and facilitates multicriteria collaborative spatial decision making.

Cross-References

- ▶ Data Analysis, Spatial
- ▶ Decision-Making Effectiveness with GIS
- ▶ Internet GIS
- ▶ Multicriteria Spatial Decision Support Systems
- ▶ Raster Data
- ▶ Sensitivity Analysis
- ▶ Temporal GIS and Applications
- ▶ Uncertainty, Modeling with Spatial and Temporal
- ▶ Vector Data
- ▶ Web Services, Geospatial

References

Carver S (1999) Developing web-based GIS/MCE: improving access to data and spatial decision support tools. In: Thill JC (eds) Multi-criteria decisionmaking and analysis: a geographic information sciences approach. Ashgate Publishing Ltd., Aldershot, pp 49–75

Feick RD, Hall BG (2004) A method for examining the spatial dimension of multicriteria weight sensitivity. Int J Geogr Inf Sci 20(7):703–726

Figueira J, Greco S, Ehrgott M (2005) Multiple criteria decision analysis: state of the art surveys. Springer, New York

Jankowski P, Andrienko N, Andrienko G (2001) Map-centered exploratory approach to multiple criteria spatial decision making. Int J Geogr Inf Sci 15:101–127

Jiang H, Esatman JR (2000) Applications of fuzzy measures in multi-criteria evaluation in GIS. Int J Geogr Inf Sci 14(2):173–184

Keeney RL, Raffia H (1976) Decisions with multiple objectives: preferences and value trade-offs. Wiley, New York

Malczewski J (1999) GIS and multicriteria decision analysis. Wiley, New York

Malczewski J (2006) A GIS-based multicriteria decision analysis: a survey of the literature. Int J Geogr Inf Sci 20(7):703–726

Marinoni O (2006) A discussion on the computational limitations of outranking methods for land-use suitability assessment. Int J Geogr Inf Sci 20(1):69–87

Roy B (1996) Multicriteria methodology for decision aiding. Kluwer Academic Publishers, Dordrecht

Saaty TL (1980) The analytic hierarchy process. McGraw-Hill, New York

Vincke PH (1992) Multicriteria decision-aid. Wiley, Chichester

Recommended Reading

Chakhar S, Martel JM (2003) Enhancing geographical information systems capabilities with multi-criteria evaluation functions. J Geogr Inf Decis Anal 7(2): 47–71

Keeney RL (1992) Valuedfocused thinking: a path to creative decision. Harvard University Press, Cambridge

Multicriteria Spatial Decision Support Systems

Salem Chakhar and Vincent Mousseau
LAMSADE, University of Paris Dauphine,
Paris, France

Synonyms

Spatial multicriteria decision support systems

Definition

A *spatial decision support system* (SDSS) is an interactive, computer-based system designed to support a user or a group of users in achieving a higher effectiveness of decision making while solving a semi-structured spatial decision problem (Malczewski 1999). It lies at the intersection of two major trends in the spatial sciences: *geographic information sciences* and *spatial analysis* (Malczewski 1999). What really differentiates a SDSS and a traditional *decision support system* (DSS) is the particular nature of the geographic data considered in different spatial problems and the high level of complexity of these problems. An effective SDSS requires enhancing conventional DSS with a range of specific techniques and functionalities used especially to manage spatial data. According to Densham (1991), a SDSS should (i) provide mechanisms for the input of spatial data, (ii) allow representation of spatial relations and structures, (iii) include the analytical techniques of spatial analysis, and (iv) provide output in a variety of spatial forms, including maps. *Multicriteria spatial decision support systems* (MC-SDSS) can be viewed as part of the broader fields of SDSS. The specificity of MC-SDSS is that it supports *spatial multicriteria decision making*. Spatial multicriteria decision making refers to the use of *multicriteria analysis* (MCA) in the context of spatial decision problems. MCA (Figueira et al. 2005) is a family of operations research tools that have experienced very successful applications in different domains since the 1960s. It has been coupled with geographical information systems (GIS) since the early 1990s for enhanced decision making.

Historical Background

The concept of SDSS has evolved in parallel with DSSs (Marakas 2003). The first MC-SDSS were developed during the late 1980s and early 1990s (Malczewski 1999). Early research on MC-SDSS is especially devoted to the physical integration of the GIS and MCA. These first tools emphasize interactively and flexibility since GIS and MCA softwares are coupled indirectly through an intermediate system. Later research concerns the development of MC-SDSS supporting collaborative and participative multicriteria spatial decision making (Jankowski et al. 1997). Web-based MC-SDSS is an active research topic which will be the subject of considerable interest in the future (Carver 1999).

Scientific Fundamentals

General Structure of SDSS/MC-SDSS

A typical SDSS contains three generic components (Malczewski 1999) (see Fig. 1): a database management system and geographical database, a model-based management system and model base, and a dialogue generation system. The data management subsystem performs all data-related tasks; that is, it stores, maintains, and retrieves data from the database, extracts data from various sources, and so on. It provides access to data as well as all of the control programs necessary to get those data in the form appropriate for a particular decision making problem. The model subsystem contains the library of models and routines to maintain them. It keeps track of all possible models that might be run during the analysis as well as controls for running the models. The model base management system component provides links between different models so that the output of one model can be the input into another model. The dialogue subsystem contains mechanisms whereby data and information are input into the system and output from the system. These three components constitute the software portion of the SDSS. A fourth important component of any decision support system is the user which may consist of simple users, technical specialists, decision makers and so on.

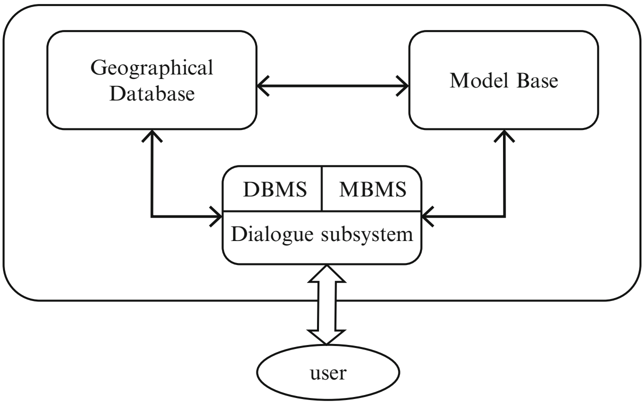
MC-SDSS can be viewed as a part of a broader field of SDSS. Accordingly, the general structure of a MC-SDSS is the same as that of a SDSS. However, the model-based management system is enhanced to support multicriteria spatial modeling and the model base is enriched with different multicriteria analysis techniques.

GIS and Multicriteria Analysis Integration Modes

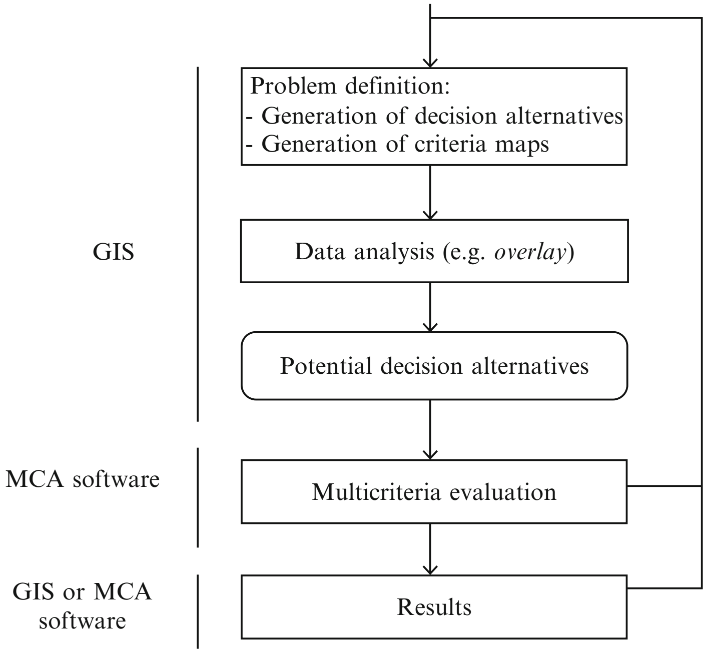
The conceptual idea on which most of GIS-based multicriteria analysis relies is to use the GIS capabilities to prepare an adequate platform for using multicriteria methods (Chakhar and Martel 2003) (see Fig. 2). The GIS-based multicriteria analysis starts with the problem identification, where the capabilities of the GIS are used to define the set of feasible alternatives and the set of criteria. Then, the *overlay* procedures are used in order to reduce an initially rich set of alternatives into a small number of alternatives which are easily evaluated by using a multicriteria method. Finally, the drawing and presenting capabilities of the GIS are used to present results.

Physically, there are four possible modes to integrate GIS and multicriteria analysis tools (Chakhar and Martel 2003; Malczewski 1999;

Multicriteria Spatial Decision Support Systems, Fig. 1 General structure of SDSS (Malczewski 1999)



Multicriteria Spatial Decision Support Systems, Fig. 2 Conceptual schema for GIS and multicriteria analysis integration


GIS
MCA software
GIS or MCA software
Problem definition:
- Generation of decision alternatives
- Generation of criteria maps
Data analysis (e.g. *overlay*)
Potential decision alternatives
Multicriteria evaluation
Results

Nyerges 1992): (i) no integration, (ii) loose integration, (iii) tight integration, and (iv) full integration. The first mode corresponds to the situation that dominated until the late 1980s, when the GIS and multicriteria analysis were used independently to deal with spatial problems. The next three modes correspond to increasing levels of complexity and efficiency (see Fig. 3).

Loose Integration Mode

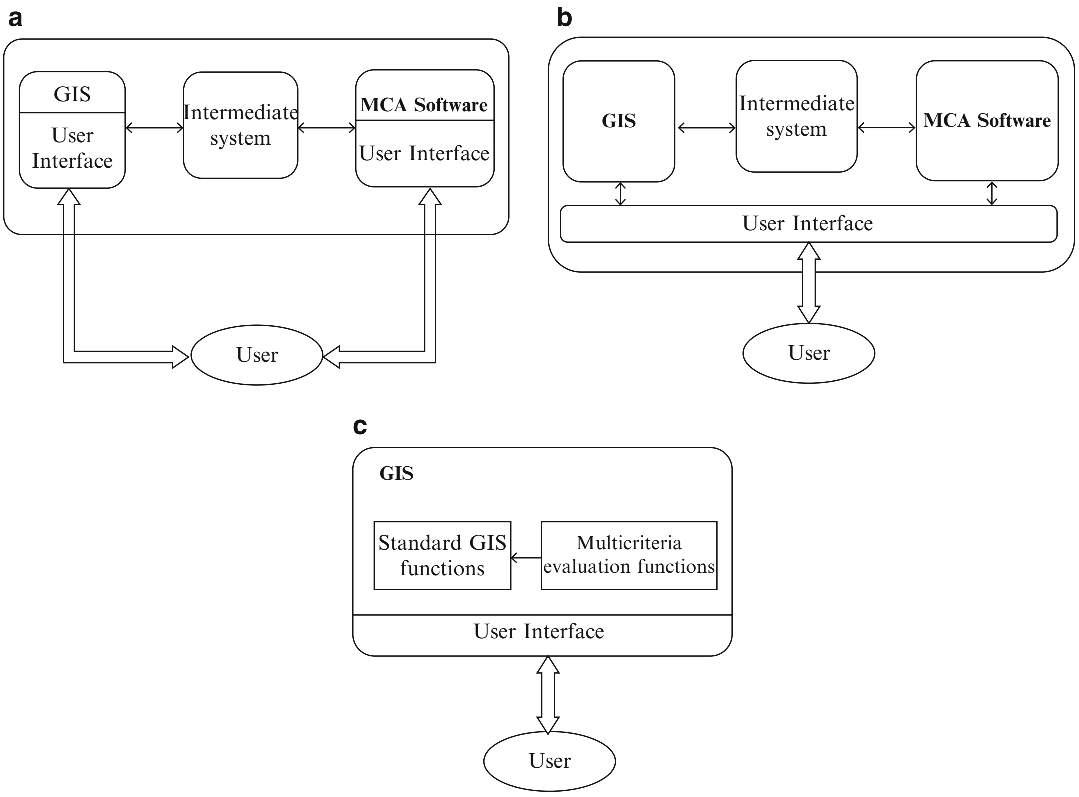
The integration of GIS software and a standalone multicriteria analysis software application is made possible by the use of an intermediate system. The intermediate system permits the reformulation and restructuring of the data obtained from the overlapping analysis performed through the GIS, and is converted into a form that is convenient to the multicriteria analysis software. The other parameters required for the analysis are introduced directly via the multicriteria analysis software interface. The results of the analysis-totally made in the multicriteria analysis software-may be visualized by using the presentation capabilities of the multicriteria analysis package, or feedback to the GIS part, via the intermediate system, for display and, eventually, for further manipulation. Each part has its own database and its own interface, which limits the user-friendliness of the system.

Tight Integration Mode

In this mode, a particular multicriteria analysis method is directly added to the GIS software. The multicriteria analysis method constitutes an integrated but autonomous part with its own database. The use of the interface of the GIS part alone increases the interactivity of the system. This mode is the first step toward a complete GIS-multicriteria analysis integrated system. Yet, with the autonomy of the multicriteria analysis method, the interactivity remains a problem.

Full Integration Mode

The third mode yields itself to a complete GIS-multicriteria analysis integrated system that has a unique interface and a unique database. Here, the multicriteria analysis method is activated directly from the GIS interface, as any GIS basic function. The GIS database is extended so as to support both the geographical and descriptive data, on the one hand, and the parameters required for the multicriteria evaluation techniques, on the other hand. The common graphical interface enhances the user-friendless of the global system.



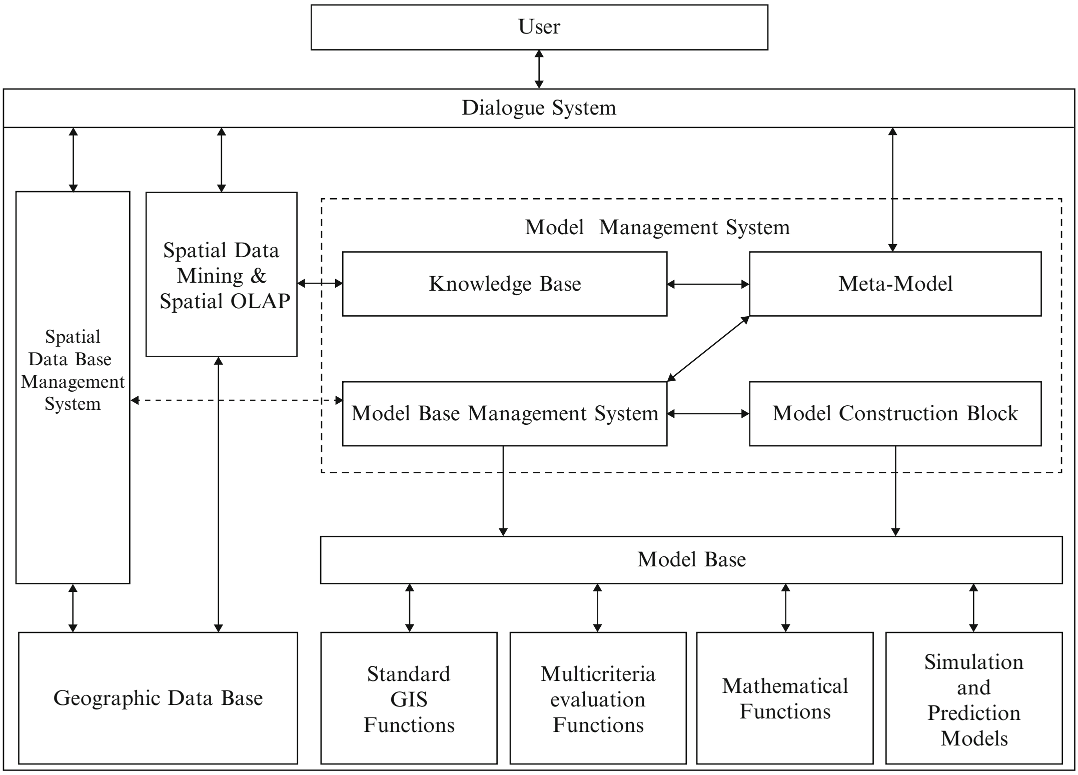
Multicriteria Spatial Decision Support Systems, Fig. 3 GIS and multicriteria loose (**a**), tight (**b**) and full (**c**) integration modes (Chakhar and Martel 2003; Malczewski 1999)

GIS and Multicriteria Analysis Interaction Directions

It is possible to distinguish five different directions of interaction (Malczewski 2006; Nyerges 1992): (i) no interaction, (ii) one-directional interaction with the GIS as the main software (iii) one-direction interaction with the multicriteria tool as the main software, (iv) bi-directional interaction, and (v) dynamic interaction. One-directional interaction provides a mechanism for importing and exporting information via a single flow that originates either in the GIS or multicriteria software. This type of interaction can be based on GIS or multicriteria as the main software. In the bi-directional interaction approach, the flow of data and information can both originate and end in the GIS and multicriteria decision making modules. Dynamic integration allows for a flexible moving of information back and forth between the GIS and multicriteria modules according to the user's needs.

Design of a MC-SDSS

Different frameworks for designing MC-SDSS have been proposed in the literature (Chakhar and Martel 2003; Jankowski et al. 1997; Malczewski 1999). Apart from differences in GIS capabilities and multicriteria techniques, most of these frameworks contain the major components introduced earlier. In the rest of this section, a revised version of the framework proposed in Chakhar and Martel (2003) is presented. This framework is conceived of in such a way that it supports GIS-MCA integration and is also open to incorporating any other OR/MS tool into the GIS (see Fig. 4).



Multicriteria Spatial Decision Support Systems, Fig. 4 A design of a multicriteria SDSS

Spatial Database Management System

The spatial database management system is an extension of the conventional database base management system. It is used specially to manage spatial data.

Geographic Database

The geographic database is an extended GIS database. It constitutes the repository for both (i) the spatial and descriptive data, and (ii) the parameters required for the different OR/MS tools.

Model Base

The model base is the repository of different analytical models and functions. These functions include the basic functions of a GIS, including statistical analysis, overlaying, spatial interaction analysis, network analysis, etc. The model base also contains other OR/MS models and perhaps the most important ones are multicriteria analysis tools. The system is also open to including any other OR/MS tool (e.g., mathematical models, simulation and prediction models, etc.), or any other ad hoc model developed by the model construction block.

Model Management System

The role of this component is to manage the different analysis models and functions. The model management system contains four elements: the meta-model, the model base management system, the model construction block and the knowledge base.

Meta-Model

This element is normally an expert system used by the decision maker to explore the model base. This exploration enables the decision maker to perform a “what-if” analysis and/or to apply different analytical functions. The meta-model uses a base of rules and a base of facts incorporated into the knowledge base. The notion of the meta-model is of great importance in the sense that it makes the system open for the addition of any OR/MS analysis tool. This requires the addition of the characteristics of the analytical tool to the base of rules, and, of course, the addition of this model to the model base.

Knowledge Base

The knowledge base is the repository for different pieces of knowledge used by the meta-model to explore the model base. The knowledge base is divided into a base of facts and a base of rules. The base of facts contains the facts generated from the model base. It also contains other information concerning the uses of different models, the number and the problems to which each model is applied, etc. The base of rules contains different production rules which are obtained from different experts, or automatically derived by the system from past experiences. For instance, this base may contain the following rule: *If the problem under study is the concern of many parties having different objective functions, then the appropriate tool to apply is multicriteria analysis (MCA).*

Model Base Management System

The role of the model base management system is to manage, execute and integrate different models that have been previously selected by the decision maker through the use of the meta-model.

Model Construction Block

This component gives the user the possibility to develop different ad hoc analysis models for some specific problems. The model that is developed can then be added directly to the model base and its characteristics can be introduced into the knowledge base.

Spatial Data Mining and Spatial on Line Analytical Processing

Data mining and *on line analytical processing* (OLAP) have been used successfully to extract relevant knowledge from huge traditional databases. Recently, several authors have been interested in the extension of these tools in order to deal with huge and complex spatial databases. In particular, Faiz (2000) underlines that *spatial data mining* is a very demanding field that refers to the extraction of implicit knowledge and spatial relationships which are not explicitly stored in geographical databases. The same author adds that *spatial OLAP* technology uses multidimensional views of aggregated, pre-packaged and structured spatial data to give quick access to information. Incorporating spatial data mining and spatial OLAP into the MC-SDSS will undoubtedly ameliorate the negative impacts when the quality of data is a problem and, consequently, add value to the decision-making process.

Dialogue System

The dialogue system represents the interface and tools used to support the dialogue between the user and the MC-SDSS. It permits the decision maker to enter queries and to retrieve the results.

Key Applications

MC-SDSS have been used in a wide range of practical applications of spatial multicriteria decision making problems. They include nuclear waste disposal facility location, solid waste management, land-use planning, corridor location, water resource management, residential site development, health care resource allocation and land suitability analysis. In the rest of this section, a brief description of some SDSS are provided.

- OSDM (Open Spatial Decision Making) (Carver 1996) is an Internet-based MC-SDSS designed to support the selection of suitable sites for radioactive waste disposal by the public in Great Britain. An important characteristic of OSDM is that it does not require prior knowledge of GIS or MCA.
- Spatial Groupe Choice (SGC) (Jankowski et al. 1997) is a GIS-based decision support system for collaborative spatial decision support making. The system has been successfully used for residential site selection

in the Duwamish Waterway and surrounding areas, and for health care resource allocation.

- IDRISI/Decision Support is a built-in decision support module for performing multicriteria decision analysis. This system has been applied in different real world applications. The case study described in Malczewski (1999) illustrates the use of the system for analyzing land suitabillity for a housing project in Mexico.
- DOCLOC has been designed for aiding health practitioners in the selection of practices in the state of Idaho (Jankowski and Ewart 1996). One limitation to this system is the use of the loose coupling strategy.
- Collaborative Planning Support System (CPSS) (Simonovic and Bender 1996) provides an example of a system employing multiobjective fuzzy decision analysis. It is a multicriteria collaborative spatial decision support system for sustainable water resource management.

Future Directions

Use Full Integration Modes

The first limitation concerning MC-SDSS is relative to the integration mode adopted. In fact, most of the proposed works use loose or tight integration modes. One possible solution to permit full integration is to identify a restricted set of *multicriteria evaluation functions* and incorporate them into the GIS (Chakhar and Martel 2003). These functions represent elementary operations required to implement the major part of multicriteria methods. This integration strategy avoids the necessity of programming the different multicriteria methods. In addition, it permits a full integration since the multicriteria evaluation functions are generic and can easily be incorporated in the available commercial GIS.

Incorporation of Large Number of Multicriteria Methods

It is well established that each multicriteria method has its advantages and disadvantages. This means that a given method may be useful in addressing some problems but not in others. One intuitive solution to this problem is to incorporate as many multicriteria methods in the MC-SDSS as possible. However, this idea has several limitations: (i) the obtained system is not flexible enough, (ii) it requires a considerable effort for programming the different methods, and (iii) there is no way to develop "personalized" methods. The integration strategy proposed in the previous paragraph permits the overall system to handle this limitation. In fact, the multicriteria evaluation functions are defined in a generic way and can be used to implement different multicriteria methods or even to create ad hoc methods adapted to the problem under consideration.

Formal Methodology to Select the Multicriteria Method to Apply

Employing a large number of multicriteria methods in the MC-SDSS permits the extension and reinforcement of the analytical potential of the GIS. However, a new problem appears: how to choose the method to use in a given problem? There are generally three possible solutions to the multicriteria method selection problem: (i) the use of a classification tree (ii) the use of a multicriteria method, and (iii) the use of an expert system or a decision support system. It is thought that the last solution is more appropriate from the perspective of GIS and multicriteria analysis integration. The development of a rule-based system needs the designers to work out (i) the characterization of the spatial decision problems, the multicriteria methods and the decision maker(s) (ii) the identification and quantification of knowledge about multicriteria methods, and (iii) the establishment of correspondences among the elements enumerated in (i). The result is a collection of rules. These last ones are then used by the inference system as a basis for selecting the most appropriate method.

Choice of the Standardization/Weighting Techniques

Among the problems that are not sufficiently treated in GIS-based multicriteria systems is the selection of the standardization and the weighting techniques. There are many different standardization/weighting techniques that can be used in MC-SDSS. It is important to note that different standardization/weighting techniques may lead to different results. The development of a formal framework for aiding the decision maker during the selection of the standardization/weighting technique-similar to the one proposed for the selection of the multicriteria method-is a good initiative.

Developing a Multicriteria Spatial Modeling Environment

The use of multicriteria analysis in the GIS is complicated by the lack of an appropriate multicriteria spatial modeling environment. A possible solution is to develop a script-like programming language that supports the different multicriteria evaluation functions. DMA, *decision map algebra*, proposed in Chakhar and Mousseau (2006) and inspired from Tomlin's (Tomlin 1990) map algebra, seems to be a good starting point.

Web-Based Multicriteria Spatial Decision Making

Web-based MC-SDSS is a recent and active research topic (Carver 1999). This is particularly important since it permits the sharing of geographical information and facilitates multicriteria collaborative spatial decision making.

Cross-References

▸ Data Analysis, Spatial
▸ Multicriteria Decision-Making, Spatial

Recommended Reading

Carver SJ (1996) Open spatial decision making on the Internet. School of Geography, University of Leeds, Yorkshire

Carver SJ (1999) Developing web-based GIS/MCE: improving access to data and spatial decision support tools. In: Thill JC (eds) Multi-criteria decision making and analysis: a geographic information sciences approach. Ashgate, Aldershot, pp 49–75

Chakhar S, Martel JM (2003) Enhancing geographical information systems capabilities with multi-criteria evaluation functions. J Geogr Inf Dec Anal 7(2): 47–71

Chakhar S, Mousseau V (2006) DMA: an algebra for multicriteria spatial modeling. In: Proceedings ICA workshop on geospatial analysis and modeling, Vienna, 8 July 2006, pp 155–185

Densham PJ (1991) Spatial decision support systems. In: Maguitre DJ, Goodchild MF, Rhind D (eds) Geographical information systems: principles and applications, vol 1. Longman, London, pp 403–412

Faiz S (2000) Data warehousing and data quality managing geographic data quality during spatial data mining and spatial OLAP. GIM Int 14(12):28–31

Figueira J, Greco S, Ehrgott M (2005) Multiple criteria decision analysis: state of the art surveys. Springer, New York

Jankowski P, Ewart G (1996) Spatial decision support system for health practitioners: selecting a location for rural health practice. Geogr Syst 3(2):297–299

Jankowski P, Nyerges TL, Smith A, Moore TJ, Horvath E (1997) Spatial group choice: a SDSS tool for collaborative spatial decision making. Int J Geogr Inf Syst 11:566–602

Malczewski J (1999) GIS and multicriteria decision analysis. Wiley, New York

Malczewski J (2006) A GIS-based multicriteria decision analysis: a survey of the literature. Int J Geogr Inf Sci 20(7):703–726

Marakas GM (2003) Decision support systems in the 21st century, 2nd edn. Prentice-Hall, Upper Saddle River

Nyerges TL (1992) Coupling GIS and spatial analytic models. In: Breshanan P, Corwin E, Cowen D (eds) Proceedings of the 5th international symposium on spatial data handling. Humanities and Social Sciences Computing Laboratory, University of South Carolina, Charleston, pp 534–543

Simonovic SP, Bender MJ (1996) Collaborative planning support system: an approach for determining evaluation criteria. J Hydrol 177(3–4):237–251

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

Multi-dimensional Access Structures

▸ Indexing, High Dimensional

Multidimensional Index

▸ Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing

Multi-dimensional Indexing

▶ Indexing, High Dimensional

Multi-dimensional Mapping

▶ Space-Filling Curves

Multi-dimensional Time Series Similarity

▶ Trajectories, Discovering Similar

Multilateration

▶ Indoor Localization

Multimedia Indexing

▶ Indexing, Hilbert R-Tree, Spatial Indexing, Multimedia Indexing

Multiple Resolution Database

▶ Abstraction of Geodatabases

Multiple Target Tracking

Baik Hoh and Marco Gruteser
Electrical and Computer Engineering/WINLAB,
Rutgers University, Piscataway, NJ, USA

Definition

The problem of associating a set of anonymous position observations with sets of prior observations to construct the traces of several moving objects or targets. More precisely, given n_t position observations at time t and previous observations $n_{t-1}, n_{t-2}, \ldots, n_{t-m}$, construct a set of traces, where each trace only contains the position observations from a single moving object. In a variation of the problem the number of actual moving objects may be greater than each n since objects may disappear and emerge during the observation interval.

Main Text

Multiple hypothesis tracking is one representative algorithm proposed by Reid in 1979 to solve this problem. This algorithm uses a linear Kalman model to represent the movement behavior of each object and to filter observation noise. The algorithm operates in three steps. First it predicts a new system state (which includes predicted positions of each object based on the prior trajectory). Then it generates hypotheses for the assignment of new samples to targets and selects the most likely hypotheses. Finally it adjusts the system state based on the Kalman equations with information from the new samples. In this process, one hypothesis is generated for each permutation of the sample set; each permutation represents one possible association of new observations to the prior targets. The algorithm then selects the hypothesis that minimizes the error between the predicted positions and actual positions across all objects.

The performance of MHT depends on the density of targets, the frequency of observations, the predictability of target movements, and the accuracy of the target movement model. Tracking performance increases with scenarios with low user density or when targets are moving on relatively straight trajectories such as cars driving on a highway. Further tracking performance improves when observations are obtained with higher frequency.

Many variations of this algorithm have been proposed to accommodate target maneuvering, insertion of new targets, and abrupt disappearance of existing targets. If objects move on well-known roadways, multi-hypothesis tracking can be improved by also considering road maps during the linking process (Civilis and Pakalnis 2005).

This problem frequently arises when several simultaneously moving objects are tracked by radar. Multi-target tracking algorithms are also a useful tool to understand the privacy of anonymized positions samples in GIS applications. This approach can be viewed as an inference attack; thus, privacy is compromised if these algorithms can recover longer traces from individual samples.

Cross-References

▶ Privacy Preservation of GPS Traces

References

Civilis A, Pakalnis S (2005) Techniques for efficient road-network-based tracking of moving objects. IEEE Trans Knowl Data Eng 17(5):698–712. Senior Member Christian S. Jensen

Reid D (1979) An algorithm for tracking multiple targets. IEEE Trans Autom Control 24(6):843–854

Multiple Worlds

▶ Smallworld Software Suite

Multiple-Image Bundle Block

▶ Photogrammetric Methods

Multirepresentation

▶ Modeling and Multiple Perceptions

Multi-resolution Aggregate Tree

Iosif Lazaridis and Sharad Mehrotra
Department of Computer Science, University of California, Irvine, CA, USA

Synonyms

aR-Tree; MRA-Tree; Ra*-Tree

Definition

A Multi-resolution Aggregate tree (MRA-tree) is a multi-dimensional indexing structure whose nodes are augmented with aggregate information about the indexed subsets of data. Typically, such indices subdivide space or group data objects; nodes contain routing information for their children nodes, e.g., in the form of spatial partitions (as in quad-trees) or bounding rectangles (as in R-trees). MRA-trees store, in addition to this information, aggregate properties of the indexed entities, e.g., the SUM of their values, the MIN value, etc. Several such aggregates can be stored or alternatively, only those that are often queried.

Main Text

MRA-trees are useful in answering aggregate queries approximately and in a progressive manner. Traditional multi-dimensional indexes help aggregate query answering by quickly gathering all relevant tuples. However, they have the limitation that each of those tuples must be handled individually. Moreover, approximate answers and answer quality guarantees cannot be easily computed. MRA-trees avoid visiting entire subsets of the data since they are summarized adequately at high-level tree index nodes and they can provide deterministic answer quality guarantees since the aggregate characteristics of the data at various levels of resolution are available throughout the tree. By exploring the tree progressively, the answer quality can improve all the way to the exact answer. They can therefore be used when the user specifies either a time deadline or answer quality requirement, trying to optimize the quality and running time, respectively, under these constraints.

Cross-References

▶ Aggregate Queries, Progressive Approximate
▶ Progressive Approximate Aggregation

Multiscale Databases

▶ Modeling and Multiple Perceptions

Multi-type Nearest Neighbor Query

▶ Trip Planning Queries in Road Network Databases

Mutation

▶ Geographic Dynamics, Visualization and Modeling

MX-Quadtree

▶ Quadtree and Octree