Chapter 1

Spatial Databases

1.1 Introduction

1.1.1 Spatial Database

Spatial database management systems [43, 58, 120, 119, 97, 74] aim at the effective and efficient management of data related to

- a space such as the physical world (geography, urban planning, astronomy, human anatomy, fluid flow or an electromagnetic field);
- biometrics (fingerprints, palm measurements, facial patterns);
- engineering design (very large scale integrated circuits, layout of a building, or the molecular structure of a pharmaceutical drug); and
- conceptual information space (virtual reality environments, multidimensional decision support systems).

A Spatial Database Management System (SDBMS) can be characterized as follows:

- A SDBMS is a software module that can work with an underlying database management system, for example, an Object-Relational database management system, or Object-oriented database management system.
- SDBMSs support multiple spatial data models, commensurate spatial abstract data types (ADTs), and a query language from which these ADTs are callable.
- SDBMSs support spatial indexing, efficient algorithms for spatial operations, and domain-specific rules for query optimization.

Spatial database research have been an active area for a couple of decades. The results of this research are being used in a number of areas. To cite a few examples, the filter-and-refine technique used in spatial query processing has been applied to subsequence mining; multidimensional-index structures such as R-tree and Quad-tree used in accessing spatial data are applied in the field of computer graphics and image processing; and space-filling curves used in spatial query processing and data storage are applied in dimension reduction problems. The field of spatial databases can be defined by its accomplishments; current research is aimed at improving its functionality, extensibility, and performance. The impetus for improving functionality comes from the needs of existing application such as Geographic Information Systems (GIS), Location Based Services (LBS), ecology and environmental management, public safety, transportation, Earth science, epidemiology, and climatology.

Commercial examples of spatial database management include ESRI's ArcGIS Geodatabase [10], Oracle Spatial [17], Informix's spatial datablades (i.e., 2D, 3D, Geodetic, etc.), IBM's DB2 Spatial Extender, and future systems such as Microsoft's Katmai. Spatial databases have played a major role in the commercial industry such as Google Earth [39] and Microsoft's Virtual Earth [75]. Research prototype examples of spatial database management systems include spatial datablades with PostGIS, MySQL's GIS, Sky Server [3] and spatial extensions. The functionalities provided by these systems include a set of spatial data types such as a points, line-segments and polygons, and a set of spatial operations such as inside, intersection, and distance. The spatial types and operations may be made a part of a query language such as SQL, which allows spatial querying when combined with an object-relational database management system [26, 108]. The performance enhancement provided by these systems include a multi-dimensional spatial index and algorithms for spatial database modeling such as OGIS and 3D Topological modeling; spatial query processing including point, regional, range, and nearest neighbor queries; and spatial data methods using a variety of indexes such as quad trees and grid cells.

1.1.2 Related Work and Our Contributions

Spatial database systems have been widely researched for more than past two decades. Published work related to spatial databases can broadly be classified as follows:

- Textbooks [97, 120, 84, 74] explain in detail about various topics in spatial databases such as logical data models for spatial data, algorithms for spatial operations, and spatial data access methods. Recent textbooks [120, 42] deal with research trends in spatial databases such as spatio-temporal databases, and moving objects databases.
- Reference books [101, 91] are useful to study areas related to spatial databases, for example, multidimensional data structures, and Geographic Information Systems (GIS).
- Journals and conference proceedings [43, 58] are a source of in-depth technical knowledge of specific problem areas in spatial databases.
- Research Surveys [106, 43, 5] summarize key accomplishments and identify research needs in various areas of spatial databases at that time.

Spatial database research has continued to advance greatly since the last survey papers in this area were published [106, 43, 5]. Our contribution in this chapter is to summarize the most recent accomplishments in spatial database research, a number of which were identified as research needs in earlier surveys. For instance, bulk loading techniques and spatial join strategies are rereferenced here as well as other advances in spatial data mining and conceptual modeling of spatial data. In addition, this chapter provides an extensive updated list of research needs in such areas as management of 3D spatial data, visibility queries, and many others. The bibliography section at the end of this chapter contains a list of over 100 references, updated with the latest achievements in spatial databases.

1.1.3 Scope and Outline

The goal of this chapter is to provide the reader with a broad introduction to spatial database systems. Spatial databases are discussed in the context of object-relational databases [26, 108, 110], which provide extensibiity to many components of traditional databases to support the spatial domain. Three major areas that receive attention in database context - conceptual, logical and physical data models are discussed(see Table 1.1). In addition, applications of spatial data for spatial data mining is also explored.

Emerging needs for spatial database systems include handling of 3D spatial data, spatial data with temporal dimension, and effective visualization of spatial data. Emergence of hardware technology such as Storage Area Networks, availability of multi-core processors are two additional fields likely to have an impact on spatial databases. Such topics of research interest are introduced at the end of each section. References are provided for further exploration.

The rest of this chapter is organized as follows: Fundamental concepts helpful to understand spatial databases are presented in section 2. Sections 3 and 4 describe spatial database modeling at conceptual and logical level. Techniques for spatial query processing are discussed in section 5. File organizations and index data structures are presented in section 6. Spatial data mining patterns and techniques are explored in section 7.

1.2 Mathematical Framework

1.2.1 Accomplishments

Mathematical Framework

Spatial data is relatively more complex compared with traditional business data. Specific features of spatial data are: i) rich data types (e.g., extended spatial objects) ii) implicit spatial relationships among the variables, iii) observations that are not independent, and iv) spatial autocorrelation among the features.

Spatial Data can be considered to have two types of attributes: non-spatial attributes and spatial attributes. Non-spatial attributes are used to characterize non-spatial features of objects, such as name, population, and unemployment rate for a city. Spatial attributes are used to define the spatial location and extent of spatial objects [19]. The spatial attributes of a spatial object most often include information related to spatial locations, e.g., longitude, latitude, elevation, as well as shape. Relationships among non-spatial objects are explicit in data inputs, e.g., arithmetic relation, ordering, instance of, subclass of, and membership of. In contrast, relationships among spatial objects are often implicit, such as overlap, intersect, and behind.

Space is a framework to formalize specific relationships among a set of objects. Depending on the relationships of interest, different models of space such as set-based space, topological space, Euclidean space, metric space and network space can be used [120]. Set-based space uses the basic notion of elements, element-equality, sets and membership to formalize the set relationships such as set-equality, subset, union, cardinality, relation, function, and convexity. Relational and object-relational databases use this model of space.

Topological space uses the basic notion of a neighborhood and points to formalize the extended object relations such as boundary, interior, open, closed, within, connected, and overlaps, which are invariant under elastic deformation. Combinatorial topological space formalizes relationships such as Euler's formula (# faces + # vertices - # edges = 2 for planar configuration). Network space is a form of topological space in which the connectivity property among nodes formalizes graph properties such as connectivity, isomorphism, shortest-path, and planarity.

Euclidean coordinatized space uses the notion of a coordinate system to transform spatial properties and relationships to properties of tuples of real numbers. Metric spaces formalize the distance relationships using positive symmetric functions that obey the triangle inequality. Many multidimensional applications use Euclidean coordinatized space with metrics such as distance.

1.2.2 Research Needs

Many spatial applications manipulate continuous spaces of different scales and with different levels of discretization. A sequence of operations on discretized data can lead to growing errors similar to the ones introduced by finite-precision arithmetic on numbers. There are preliminary results [43] on the use of discrete basis and bounding errors with peg-board semantics. Another related problem concerns interpolation to estimate the continuous field from a discretization. Negative spatial autocorrelation makes interpolation error-prone. Further work is needed on a framework to formalize the discretization process, its associated errors, and on interpolation.

Mathematical Framework
Conceptual Data Model
Logical Data Model Query Languages
Query Processing File Organizations and Indices
Trends: Spatial Data Mining

Table 1.1: Spatial Database topics

1.3 Spatial Database Conceptual Modeling

1.3.1 Accomplishments

Modeling with Pictograms

Entity Relationship (ER) diagrams are commonly used in designing the conceptual model of a database. Many extensions [45] have been proposed to extend ER to make the conceptual modeling of spatial applications easier and more intuitive. One such extension is the use of pictograms [100]. A pictogram is a graphical icon that can represent a spatial entity or a spatial relationship between spatial entities. The idea is to provide constructs to capture the semantics of spatial applications and at the same time to keep the graphical representation simple. Figure 1.2 provides different types of pictograms for spatial entities and relationships. In the following text we define pictograms to represent spatial entities and relationships, and their grammar in graphical form.

Pictogram: A pictogram is a representation of the object inserted inside of a box. These iconic representations are used to extend ER diagrams and are inserted at appropriate places inside the entity boxes. An entity pictogram can be of a basic shape or a user-defined shape.

Shape: Shape is the basic graphical element of a pictogram that represents the geometric types in the spatial data model. It can be a basic shape, a multishape, a derived shape, or an alternate shape. Most objects have simple(basic) shapes (Figure 1.1 B).

Basic Shape: In a vector model the basic elements are point, line and polygon. In a forestry example, the user may want to represent a facility as a point (0-D), a river or road network as lines (1-D) and forest areas as polygons (2-D) (Figure 1.1 D).

Multi-Shape: To deal with objects which cannot be represented by the basic shapes, we have defined a set of aggregate shapes. Cardinality is used to quantify multi-shapes. For example, a river network that is represented as a line pictogram scale will have cardinality 0 (Figure 1.1 B and E).

Derived Shape: If the shape of an object is derived from the shapes of other objects, its pictogram is italicized. For example, we can derive a forest boundary (polygon) from its "forest type" boundaries (polygon), or a country boundary from the constituent state boundaries (Figure 1.1 C and G).

Alternate Shape: Alternate shapes can be used for the same object depending on certain conditions; for example, objects of size less than x units are represented as points while those greater than x units are represented as polygons. Alternate shapes are represented as a concatenation of possible pictograms. Similarly, multiple shapes are needed to represent objects at different scales; for example, at higher scales lakes may be represented as points, and at lower scales as polygons (Figure 1.1 D and H).

Any Possible Shape: A combination of shapes is represented by a wild card * symbol inside a box, implying that any geometry is possible (Figure 1.1 E).

User-Defined Shape: Apart from the basic shapes of point, line and polygon, user-defined shapes are possible. User-defined shapes are represented by an exclamation symbol (!) inside a box (Figure 1.1 A).

Relationship Pictograms: The relationship pictograms are used to model the relationship between the entities. For example, *part_of* is used to model the relationship between a route and a network, or it can be used to model the partition of a forest into forest stands (Figure 1.1 C).

The popularity of object-oriented languages such as C++ and Java has encouraged the growth of object-oriented database systems (OODBMS). The motivation behind this growth in OODBMS is that the direct mapping of the conceptual database schema into an object-oriented language leads to a reduction of impedance mismatch encountered when a model on one level is converted into a model on another level.

UML is one of the standards for conceptual level modeling for object-oriented software design. It may also be applied to an OODBMS to capture the design of the system conceptually. A UML design consists of the following building blocks.

Class: A class is the encapsulation of all objects which share common properties in the context of the application. It is the equivalent of the entity in the ER model. The class diagrams in a UML design can further be extended by adding pictograms. In a forestry example, classes can be Forest, Facility, Forest Stand, etc.

Attributes: Attributes characterize the objects of the class. The difference between an attribute ER and UML model design is that there is no notion of a key attribute in UML. This is because in an object-oriented system, each object has an implicit system-generated unique identification. In UML, attributes also have a

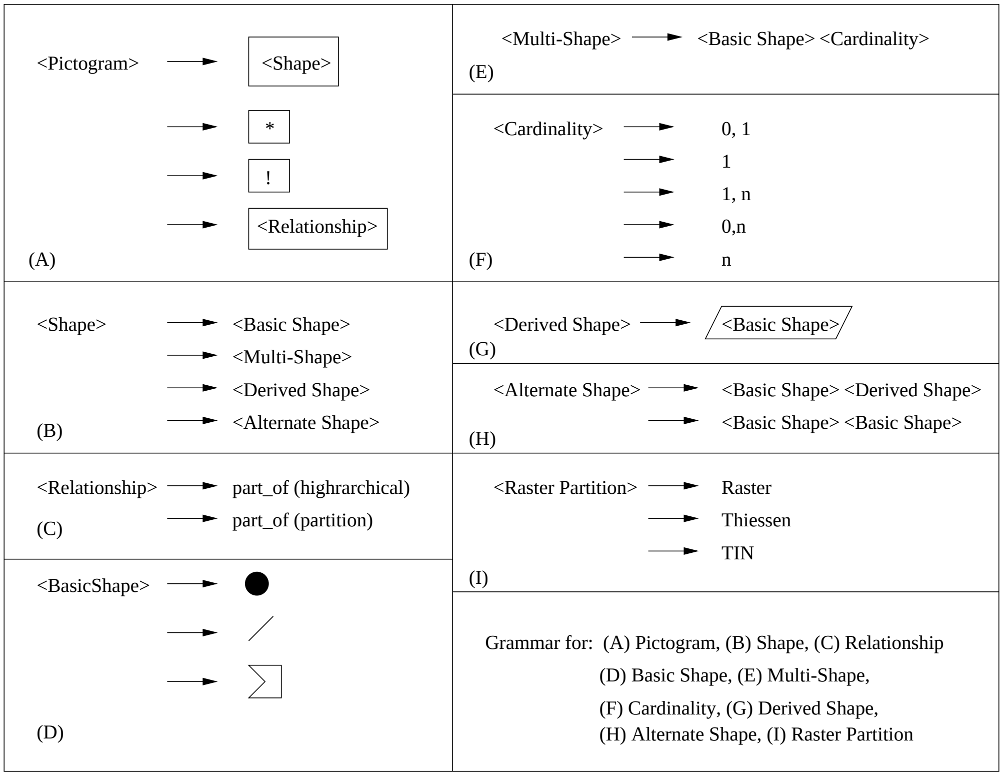


Figure 1.1: Pictograms

scope that restricts its access by other classes. There are three levels of scope, and each has a special symbol: + Public: This allows the attribute to be accessed and manipulated from any class. - Private: Only the class that owns the attribute is allowed to access the attribute. # Protected: Other than the class that owns the attribute, classes derived from the class that owns can access the attibute.

Methods: Methods are functions and a part of class definition. They are responsible for modifying the behavior, or state of the class. The state of the class is embodied in the current values of the attributes. In object-oriented design, attributes should only be accessed through methods.

Relationships: Relationships relate one class to another or to itself. This is similar to the concept of relationship in the ER model. There are three important categories of relationships:

- Aggregation: This is a specific construct to capture the part-whole relationship. For instance, a group of Forest-Stand classes may be aggregated into a Forest class.

- Generalization: This is a relationship in which a child class can be generalized to a parent class. For example, classes such as Point, Line and Polygon can be generalized to a Geometry class.

- Association: This shows how objects of different classes are related. An association is binary if it connects two classes or ternary if it connects three classes. An example of a binary association is *supplies_water_to* between the classes River and Facility.

Figure 1.2 and Figure 1.3 provides an example for modeling a *State-Park* using ER and UML with pictograms, respectively.

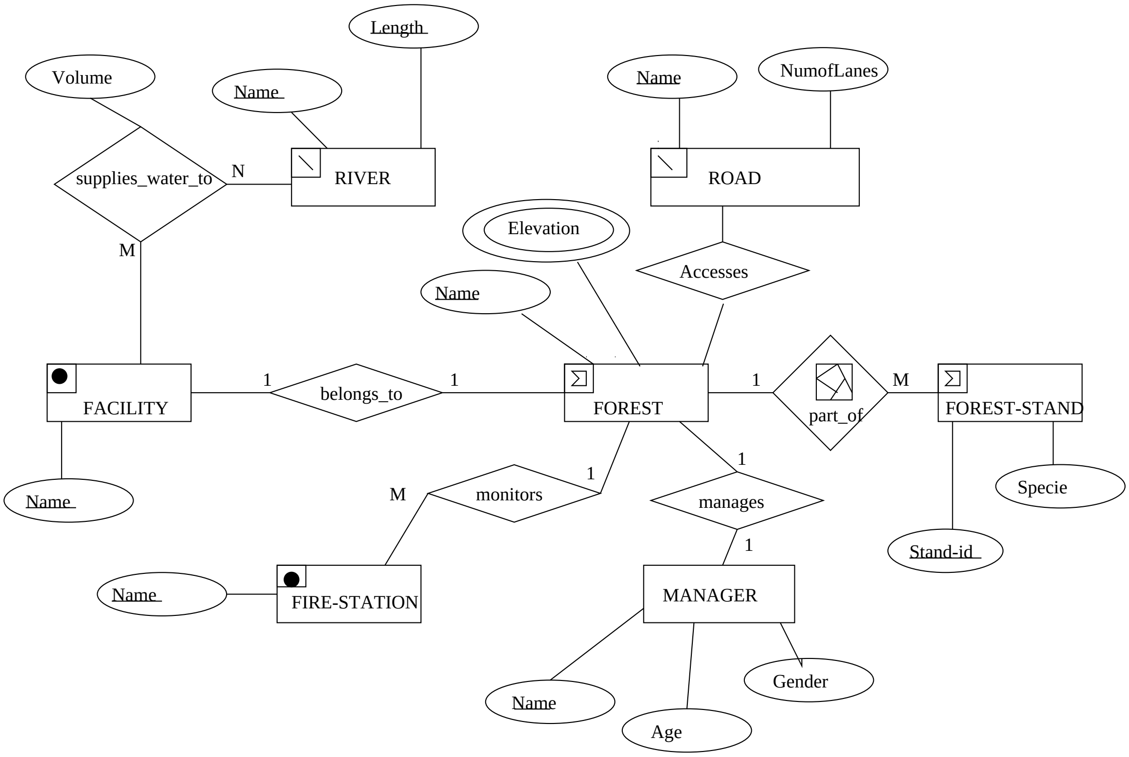


Figure 1.2: Example ER diagram with pictograms

1.3.2 Research Needs

Conceptual modeling for spatio-temporal and moving-object data needs to be researched. Pictograms as introduced in this section may be extended to handle such data. Another area of research is the use of ontology for knowledge management. An ontology defines a common vocabulary that allows knowledge to be shared and reused across different applications. Ontologies provide a shared and common understanding of some domain that can be communicated across people and computers. **Geospatial ontology** [35] is specific to the geospatial domain. Research in geospatial ontology is needed to provide interoperability between geospatial data and software. Geospatial ontologies is one of the long-term research challenges for University Consortium for Geographic Information System (UCGIS) [1].

1.4 Spatial Data Models and Query Languages

1.4.1 Accomplishments

Data Models A spatial data model provides the data abstraction necessary to hide the details of data storage. The two commonly used models are the field-based model and the object-based model. While the field-based model adopts a functional viewpoint, object-based model treats the information space as a collection of discrete, identifiable, spatially-referenced entities. Based on the type of data model used, the spatial operations may change. Table 1.2 lists the operations specific to the field-based and object-based models. In the context of object-relational databases, a spatial data model is implemented using a set of spatial data types and operations. Over the last two decades, an enormous amount of work has been done in the design and development of spatial abstract data types and their embedding in a query language. Serious efforts are being made to arrive at a consensus on standards through the OGIS consortium [55].
OGIS proposed the general feature model [55] where features are considered to occur at two levels, namely, feature instances and feature types. A geographic feature is represented as a discrete phenomenon characterized by its geographic and temporal coordinates at the instance level and the instances with common

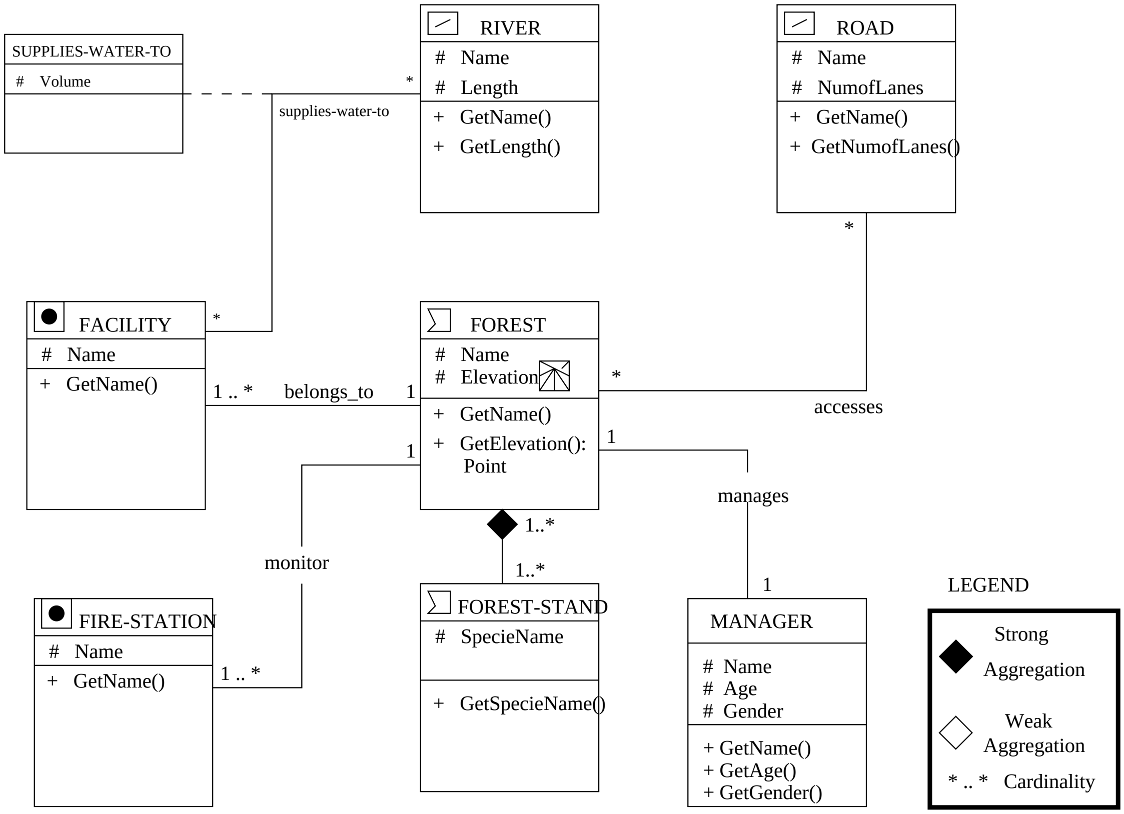


Figure 1.3: Example UML class diagram with pictograms

characteristics are grouped into classes called feature types. Direction is another important feature used in spatial applications. A direction feature can be modeled as a spatial object [94]. Research has also been done to efficiently compute the cardinal direction relations between regions that are composed of sets of spatial objects [103].

Query Languages

When it comes to database sytems, spatial database researchers prefer object-based models because the data types provided by object-based database sytems can be extended to spatial data types by creating abstract data types (ADT). OGIS provides a framework for object-based models. Figure 1.4 shows the OpenGIS approach to modeling geographic features. This framework provides conceptual schemas to define abstract feature types and provides facilities to develop application schemas that can capture data about feature instances. Geographic phenomena fall into two broad categories, discrete and continuous. Discrete phenomena are objects that have well-defined boundaries or spatial extent, examples being buildings and streams. A continuous phenomena vary over space and have no specific extent (e.g., temperature,elevation). A continuous phenomenon is described in terms of its value at a specific position in space (and possibly time). OGIS represents discrete phenomena (also called vector data) by a set of one or more geometric primitives (points, curves, surfaces, or solids). Continuous phenomenon is represented through a set of values, each associated with one of the elements in an array of points. OGIS uses the term ”coverage” to refer to any data representation that assigns values directly to spatial position. A coverage is a function from a spatio-temporal domain to an attribute domain. OGIS provides standardized representations for spatial characteristics through geometry and topology. Geometry provides the means for the quantitative description of the spatial characteristics including dimension, position, size, shape, and orientation. Topology deals with the characteristics of geometric figures that remain invariant if the space is deformed elastically and continuously. Figure 1.5 shows the hierarchy of geometry data types. Objects under Primitive will be open (i.e., they will not contain their boundary points) and the objects under Complex will be closed.

In addition to defining the spatial data types, OGIS also defines spatial operations. Table 1.3 lists basic

Data Model	Operator Group	Operation
Vector Object	Set-Oriented	equals, is a member of, is empty, is a subset of, is disjoint from, intersection, union, difference, cardinality
	Topological	boundary, interior, closure, meets, overlaps, is inside, covers, connected, components, extremes, is within
	Metric	distance, bearing/angle, length, area, perimeter
	Direction	east, north, left, above, between
	Network	successors, ancestors, connected, shortest-path
	Dynamic	translate, rotate, scale, shear, split, merge
Raster Field	Local	Point-wise sums, differences, maximums, means, etc
	Focal	slope, aspect, weighted average of neighborhood
	Zonal	sum or mean or maximum of field values in each zone

Table 1.2: Data Model and Operations

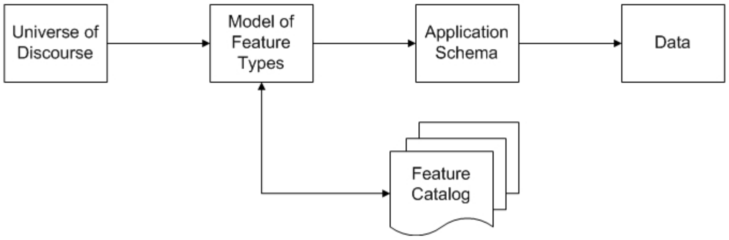


Figure 1.4: Modeling Geographic Information(Source: [55])

operations operative on all spatial data types. The topological operations are based on the ubiquitous nine-intersection model. Using the OGIS specificatin, common spatial queries can be intuitively posed in SQL. For example, the query *Find all lakes which have an area greater than 20 sq. km. and are within 50 km. from the campgrounds* can be posed as shown in Table 1.4 and Figure 1.6. Other example GIS and LBS queries are provided in Table 1.5. The OGIS specification is confined to topological and metric operations on vector data types.

For spatial networks, commonly used spatial data types includes objects such as Node, Edge, and Graph. They may be constructed as an ADT in a database system. Query languages based on relational algebra are unable to express certain important graph queries without making certain assumptions about the graphs. For example, the transitive closure of a graph may not be determined using relational algebra. In the SQL3, a recursion operation RECURSIVE has been proposed to handle the transitive closure operation.

1.4.2 Research Needs

Map algebra [27] is a framework for raster analysis that has now evolved to become a preeminent language for dealing with field-based models.

Modeling 3D Data Efforts [57] have been made to develop three dimensional data models that emphasize the significance of volumetric shapes of physical world objects. This topological 3D data model relies on Poincaré algebra. The internal structure is based on a network of simplexes and the internal data structure used is a Tetrahedronized Irregular Network (TIN) [12, 57], which is the three-dimensional variant of the well-known Triangulated Irregular Network (TIN).

Modeling Spatial Temporal Networks Graphs have been extensively used to represent spatial networks. Considering the time-dependence of the network parameters and the topology, it has become critically important to incorporate the temporal nature of these networks into their models to make them more accurate and effective. For example, in a transportation network the travel times on road segments are often depenedent on the time of the day and there can be intervals when certain road segments are not available for service. These aspects make the network time-dependent and it becomes important to model the time variance. Time expanded graphs [59] and time aggregated graphs [38] have been used to model time varying

Basic Functions	
SpatialReference()	Returns the underlying coordinate system of the geometry
Envelope()	Returns the minimum orthogonal bounding rectangle of the geometry
Export()	Returns the geometry in a different representation
IsEmpty()	Returns true if the geometry is an empty set.
IsSimple()	Returns true if the geometry is simple (no self-intersection)
Boundary()	Returns the boundary of the geometry
Topological/ Set Operators	
Equal	Returns true if the interior and boundary of the two geometries are spatially equal
Disjoint	Returns true if the boundaries and interior do not intersect.
Intersect	Returns true if the interiors of the geometries intersect
Touch	Returns true if the boundaries intersect but the interiors do not.
Cross	Returns true if the interior of the geometries intersect but the boundaries do not
Within	Returns true if the interior of the given geometry does not intersect with the exterior of another geometry.
Contains	Tests if the given geometry contains another given geometry
Overlap	Returns true if the interiors of two geometries have non-empty intersection
Spatial Analysis	
Distance	Returns the shortest distance between two geometries
Buffer	Returns a geometry that consists of all points whose distance from the given geometry is less than or equal to the specified distance
ConvexHull	Returns the smallest convex set enclosing the geometry
Intersection	Returns the geometric intersection of two geometries
Union	Returns the geometric union of two geometries
Difference	Returns the portion of a geometry which does not intersect with another given geometry
SymmDiff	Returns the portions of two geometries which do not intersect with each other

Table 1.3: A sample of operations listed in the OGIS standard for SQL

```
SELECT L.name
FROM Lake L, Facilities Fa
WHERE Area(L.Geometry) > 20 AND
Fa.name = ‘campground’ AND
Distance(Fa.Geometry, L.Geometry) < 50
```

Table 1.4: SQL Query with spatial operators

spatial networks. In the time expanded representation, a copy of the entire network is maintained for every time instant whereas the time aggregated graphs maintain a time series of attributes, associated to every node and edge.

Modeling Moving Objects A moving object database is considered to be a spatio-temporal database in which the spatial objects may change their position and extent over a period of time. To cite a few examples, the movement of taxi cabs, the path of a hurricane over a period of time, and geographic profiling of serial criminals are a few examples where a moving objects database may be considered. [42, 30] have provided a data model to support the design of such databases.

Visibility Queries Visibility has been widely studied in Computer Graphics. Visibility may be defined as the parts of objects and the environment visible from a point in space. A visibility query can be thought of as a query that returns the objects and part of the environment visible at the querying point. For example, within a city, if the coverage area of a wireless antenna is considered to be the visible area, then the union of coverage areas of all the antennas in the city will provide an idea about the area which is not covered. Such information may be used to strategically place a new antenna at an optimal location. In a visibility query, if the point in space moves, the area of visibility changes. Such a query may be called a continuous visibility query. To illustrate with an example, security for the President’s motorcade involves cordoning off the buildings which have route visibility. In such a case, the visibility query may be thought of as a query

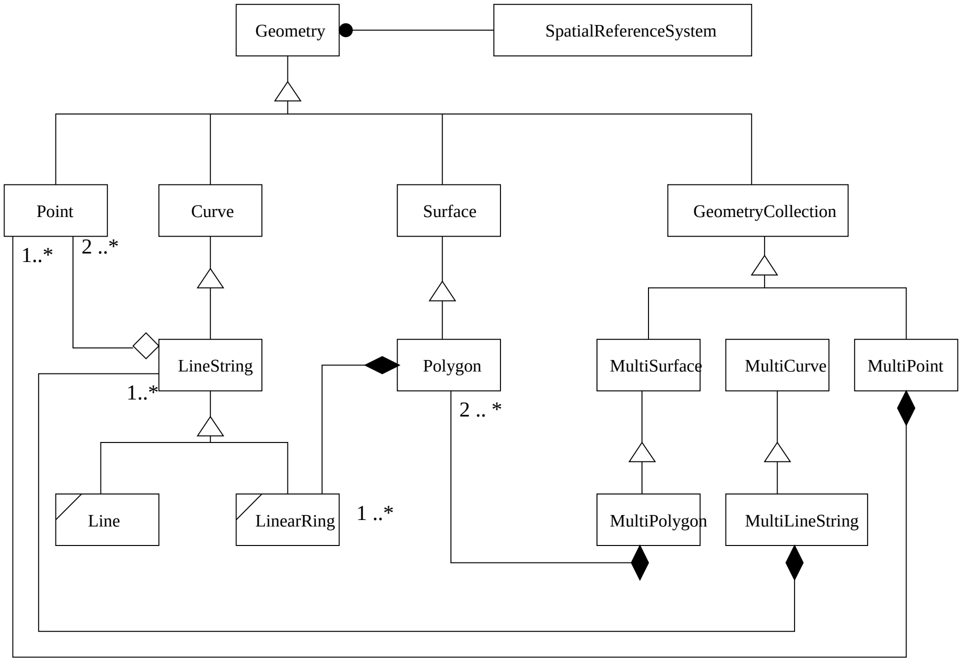


Figure 1.5: Hierarchy of Data Types

that returns the buildings visible at different points on the route.

Visual Querying Many spatial applications present results visually, in the form of maps which consist of graphic images, 3D displays, and animations. These applications allow users to query the visual representation by pointing to the visual representation using pointing devices like a mouse or a pen. Such graphical interfaces are needed to query spatial data without the need to write any SQL statements by the users of spatial applications. In recent years, map services, such as Google Earth and Microsoft Earth, have become very popular. Further work is needed to explore the impact of querying by pointing and visual presentation of results on database performance.

GML Geography Markup Language (GML) is an XML based markup language to represent geographic entities and the relationships between them. The goals of GML are to provide a standard for modeling language and data exchange formats for geographic data. Entities associated with geospatial data such as geometry, coordinate systems, attributes, etc. can be represented in a standard way using GML. CityGML [2] is a subclass of GML useful to represent 3D urban objects, such as buildings, bridges, tunnels, etc.

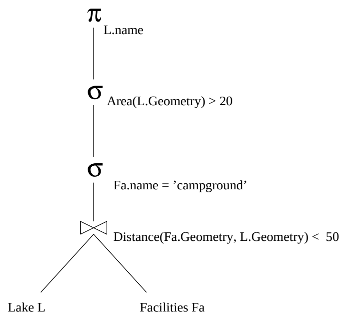
1.5 Spatial Query Processing

1.5.1 Accomplishments

The efficient processing of spatial queries requires both efficient representation and efficient algorithms. Common representations of spatial data in an object model include spaghetti, the node-arc-node (NAA) model, the doubly connected-edge-list (DCEL), and boundary representation, some of which are shown in Figure 1.7 using entity-relationship diagrams. The NAA model differentiates between the topological concepts (node, arc, areas) and the embedding space (points, lines, areas). The spaghetti-ring and DCEL focus on the topological concepts. The representation of the field data model includes a regular tessellation (triangular, square, hexagonal grid), as well as triangular irregular networks (TIN).

Query processing in spatial databases differs from that of relational databases because of the following three major issues:

- Unlike relational databases, spatial databases have no fixed set of operators that serve as building blocks for query evaluation.



(a)

Figure 1.6: SQL query tree

GIS Queries	
Grouping	Recode all land with silty soil to silt-loadm soil
Isolate	Select all land owned by Steve Steiner
Classify	If the population density is less than 100 people / sq. mi., land is acceptable
Scale	Change all measurement to the metric system
Rank	If the road is an Interstate, assign it code 1; if the road
	is a state or US highway, assign it code 2; otherwise assign it code 3
Evaluate	If the road code is 1, then assign it Interstate; if the road code is 2,
	then assign it Main Artery; if the road code is 3, assign it Local Road
Rescale	Apply a function to the population density
Attribute Join	Join the Forest layer with the layer containing forest-cover codes
Zonal	Produce a new map showing state populations given county population
Registration	Align two layers to a common grid reference
Spatial Join	Overlay the land-use and vegetation layers to produce a new layer

LBS Queries	
Nearest Neighbor	List the nearest gas stations
Directions	Display directions from a source to a destation (e.g. Google Maps, Map Quest)
Local Search	Search for restaurants in the neighborhood (e.g. Microsoft Live Local, Google Local)

Table 1.5: Typical Spatial Queries from GIS and LBS

- Spatial databases deal with extremely large volumes of complex objects. These objects have spatial extensions and cannot be naturally sorted in a one-dimensional array.

- Computationally expensive algorithms are required to test for spatial predicates, and the assumption that I/O costs dominate processing costs in the CPU is no longer valid.

In this section, we describe the processing techniques for evaluating queries on spatial databases, and discuss open problems in spatial query processing and query optimization.

Spatial Query Operations

Spatial query operations can be classified into four groups [37].

- **Update Operations:** These include standard database operations such as modify, create, delete.

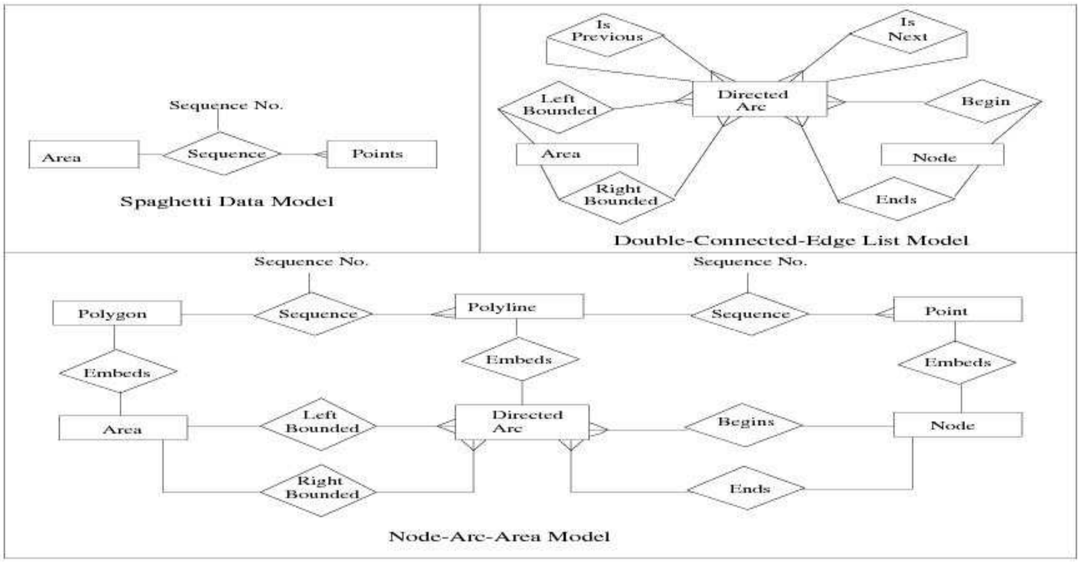


Figure 1.7: Entity relationship diagrams for common representations of spatial data

- **Spatial Selection:** These can be of two types:
  - **Point Query**: Given a query point, find all spatial objects that contain it. For an example, consider the following query, "Find all river flood-plains which contain the SHRINE.".
  - **Regional Query**: Given a query polygon, find all spatial objects which intersect the query polygon. When the query polygon is a rectangle, this query is called a window query. These queries are sometimes also referred to as range queries. An example query could be "Identify the names of all forest stands that intersect a given window."
  - **Spatial Join:** Like the join operator in relational databases, the spatial join is one of the more important operators. When two tables are joined on a spatial attribute, the join is called a spatial join. A variant of the spatial join and an important operator in GIS is the *map overlay.* This operation combines two sets of spatial objects to form new ones. The "boundaries" of a set of these new objects are determined by the non-spatial attributes assigned by the overlay operation. For example, if the operation assigns the same value of the non-spatial attribute to two neighboring objects, then the objects are "merged". Some examples of spatial join predicates are *intersect, contains, is_enclosed_by, distance, northwest, adjacent, meets, overlap.* A query example of a spatial join is "Find all forest-stands and river flood-plains which overlap".
  - **Spatial Aggregate:** An example of a spatial aggregate is "Find the river closest to a camp-ground". Spatial aggregates are usually variants of the *Nearest Neighbor* [50, 88, 82] search problem: given a query object, find the object having minimum distance from the query object.

    A *Reverse Nearest Neighbor* (RNN) [60, 107, 78, 121] query is another example of a spatial aggregate. Given a query object, a RNN Query finds objects for which the query object is the nearest neighbor. Applications of RNN include army strategic planning where a medical unit, A, in the battlefield is always in search of wounded soldier to whom *A* is the soldeir's nearest medical unit.

Two-Step Query Processing of Spatial Operations

Since spatial query processing involves complex data types, a lake boundary might need a thousand vertices for exact representation. Spatial operations typically follow a two-step algorithm (*filter* and *refinement*) as shown in Figure 1.8 to efficiently process complex spatial objects [21]. Approximate geometry such as

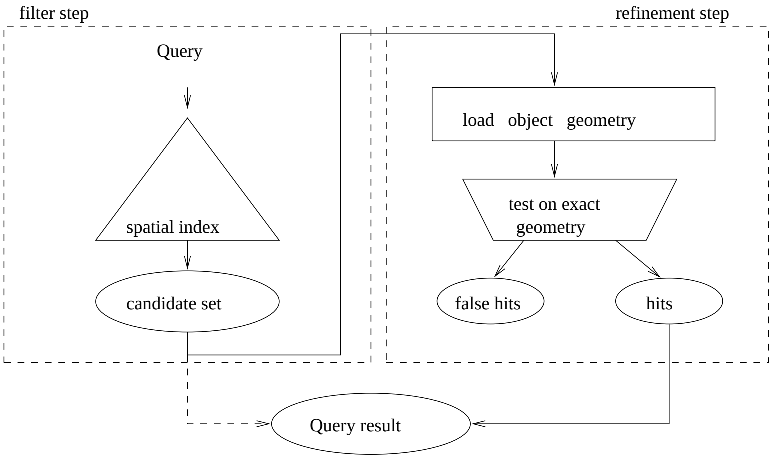


Figure 1.8: Two-step processing

the minimal orthogonal bounding rectangle of an extended spatial object is first used to filter out many irrelevant objects quickly. Exact geometry is then used for the remaining spatial objects to complete the processing.

- **Filter step:** In this step, the spatial objects are represented by simpler approximations like the minimum bounding rectangle(MBR). For example, consider the following point query, "Find all rivers whose flood-plains overlap the SHRINE". In SQL this will be:

  ```
  SELECT   river.name
  FROM     river
  WHERE    overlap(river.flood-plain, :SHRINE)
  ```

  If we approximate the flood-plains of all rivers with MBRs, then it is less expensive to determine whether the point is in a MBR than to check if a point is in an irregular polygon, that is, in the exact shape of the flood-plain. The answer from this approximate test is a superset of the real answer set. This superset is sometimes called the candidate set. Even the spatial predicate may be replaced by an approximation to simplify a query optimizer. For example, touch(river.flood-plain, :SHRINE) may be replaced by overlap(MBR(river.flood-plain, :SHRINE), and MBR(:SHRINE)) in the filter step. Many spatial operators, for example, inside, north-of and buffer can be approximated using the overlap relationship among corresponding MBRs. Such a transformation guarantees that no tuple from the final answer using exact geometry is eliminated in the filter step.

- **Refinement step:** Here, the exact geometry of each element from the candidate set and the exact spatial predicate is examined. This usually requires the use of a CPU-intensive algorithm. This step may sometimes be processed outside the spatial database in an application program such as GIS, using the candidate set produced by the spatial database in the filter step.

Techniques for Spatial Operations

Selection Operation

Similar to traditional database systems, the selection operation can be performed on indexed or non-indexed spatial data. The difference is in the technique used to evaluate the predicate and the type of indexes. As discussed in the previous section, a two-step approach, where the geometry of a spatial object is approximated by a rectangle, is commonly used to evaluate a predicate. Popular indexing techniques for spatial data are R-tree, and space-filling curves. An R-tree is a height-balanced tree that is a natural extension of a B-tree for k-dimensions. It allows a point search to be processed in $O(log\ n)$ time. Hash filling curves provides one-to-one continuous mappings which map points of multi-dimensional space into one-dimensional space. This allows the user to impose order on higher dimensional spaces. Common examples of space-filling curves are row-order Peano, Z-order, and Hilbert curves. Once the data has been ordered by a space-filling curve, a B-tree index can be imposed on the ordered entries to enhance the search. Point search operations can be performed in $O(log\ n)$ time.

Spatial Join Operation

Conceptually a join is defined as a cross-product followed by a selection condition. In practice, this viewpoint can be very expensive, because it involves materializing the cross-product before applying the selection criterion. This is especially true for spatial databases. Many ingenious algorithms have been proposed to preempt the need to perform the cross-product. The two-step query processing technique described in the previous section is the most commonly used. In this way, the spatial join operation will be reduced to a rectangle-rectangle intersection, the cost of which is relatively modest compared to the I/O cost of retrieving pages from secondary memory for processing.

A number of strategies have been proposed for processing spatial joins. Interested readers are encouraged to refer to [72, 66, 124].

Aggregate Operation: Nearest Neighbor, Reverse Nearest Neighbor

Nearest Neighbor queries are common in many applications. For example, a person driving on the road may want to find the nearest gas station from current location. Various algorithms exist for nearest neighbors queries [50, 88, 82, 53, 122]. Techniques based on Voronoi diagrams, Quad-tree indexing, Kd-trees have been discussed in [91]

Reverse Nearest Neighbor queries were introduced in [60] in the context of decision support systems. For example, a RNN query can be used to find a set of customers who can be influenced by opening of a new store outlet location.

Bulk Loading

Bulk operations affect potentially a large set of tuples, unlike other database operations, such as insert into a relation, which affects possibly one tuple at a time. Bulk loading refers to the creation of an index from scratch on a potentially large set of data. Bulk loading has its advantages because the properties of the data set may be known in advance. These properties may be used to efficiently design the space-partitioning index structures commonly used for spatial data. An evaluation of generic bulk loading techniques is provided in [29].

Parallel GIS

A High Performance Geographic Information System (HPGIS) is a central component of many interactive applications like real-time terrain visualization, situation assessment, and spatial decision-making. The Geographic Information System (GIS) often contains large amounts of geometric and feature data (e.g. location, elevation, soil type, etc.) represented as large sets of points, chains of line segments, and polygons. This data is often accessed via range queries. The existing sequential methods for supporting the GIS operations do not meet the realtime requirements imposed by many interactive applications.

Hence, parallelization of GIS is essential for meeting the high performance requirements of several real-time applications. A GIS operation can be parallelized either by function-partitioning [6, 8, 105] or by data-partitioning [13, 22, 36, 51, 52, 64, 117, 123, 95] . Function-Partitioning uses specialized data structures (e.g. distributed data structures) and algorithms which may be different from their sequential counterparts. Data-Partitioning techniques divide the data among different processors and independently execute the sequential algorithm on each processor. Data-Partitioning in turn is achieved by declustering [33, 69] the spatial data. If the static declustering methods fail to equally distribute the load among different processors, the load-balance may be improved by redistributing parts of the data to idle processors using Dynamic Load-Balancing (DLB) techniques.

1.5.2 Research Needs

Query processing Many open research areas exist at the logical level of query processing, including query-cost modeling and queries related to fields and networks. Cost models are used to rank and select the promising processing strategies, given a spatial query and a spatial data set. Traditional cost models may not be accurate in estimating the cost of strategies for spatial operations, due to the distance metric as well as the semantic gap between relational operators and spatial operation. Cost models are needed to estimate the selectivity of spatial search and join operations toward comparison of execution-costs of alternative processing strategies for spatial operations during query optimization. Preliminary work in the context of the R-tree, tree-matching join, and fractal-model is promising [18, 114], but more work is needed.

Many processing strategies using the overlap predicate have been developed for range queries and spatial join queries. However, there is a need to develop and evaluate strategies for many other frequent queries such as those listed in Table 1.6. These include queries on objects using predicates other than overlap, queries on fields such as slope analysis, and queries on networks such as the shortest path to a set of destinations.

Depending on the type of spatial data and the nature of query, other research areas also need to be investigated. A **moving objects query** involves spatial objects that are mobile. Examples include "Which is the nearest taxi cab to the customer?", "Where is the hurricane expected to hit next?", "What is a possible location of a serial criminal?" are few examples of moving objects queries. Several techniques [31, 41, 42] have been proposed to execute such queries.

A skyline query [20] is a query to retrieve a set of a interesting points (records) from a potentially huge collection of points (records) based on certain attributes. For example, considering a set of hotels to be points, the skyline query may return a set of interesting hotels based on a user's preferences. The set of hotels returned for a user who prefers cheap hotel may be different from the set of hotels returned for a user who prefers hotels which are closer to the coast. Research needed for skyline query operation includes computation of algorithms, and processing for higher dimensions (attributes). Other query processing techniques where research is required are querying on **3D spatial data** and **spatio-temporal data**.

Query optimization

The *query optimizer*, a module in the database software, generates the different evaluation plans and determines the appropriate execution strategy. Before the query optimizer can operate on the query, the high level declarative statement must be scanned through a *parser*. The parser checks the syntax and transforms the statement into a *query tree*. In traditional databases, the data types and functions are fixed and the parser is relatively simple. Spatial databases are examples of an extensible database system and have provisions for user-defined types and methods. Therefore, compared to traditional databases, the parser for spatial databases has to be considerably more sophisticated to identify and manage user-defined data types and map them into syntactically correct query trees. In the query tree, the leaf nodes correspond to the relations involved and the internal nodes correspond to the basic operations that constitute the query. Query processing starts at the leaf nodes and proceeds up the tree until the operation at the root node has been performed. Consider the following query, "Find all lakes which have an area greater than 20 sq. km. and are within 50 km. from the campground."

Let us assume that the **Area()** function is not pre-computed and that its value is computed afresh every time it is invoked. A query tree generated for the query is shown in Figure 1.9 (a). In the classical situation, the rule "select before join" would dictate that the **Area** function be computed before the join predicate function, **Distance()**(Figure 1.9 (b)), the underlying assumption being that the computational cost of executing the select and join predicate is equivalent and negligible compared to the I/O cost of the

Table 1.6: Difficult Spatial Queries form GIS

Voronoize	Classify households as to which supermarket they are closest to
Network	Find the shortest path from the warehouse to all delivery stops
Timedependentnetwork	Find the shortest path where the road network is dynamic
Allocation	Where is the best place to build a new restaurant
Transformation	Triangulate a layer based on elevation
BulkLoad	Load a spatial data file into the database
Raster ↔ Vector	Convert between raster and vector representations
Visibility	Find all points of objects and environment visible from a point
EvacuationRoute	Find evacuation routes based on capacity and availability constraints
PredictLocation	Predict the location of a mobile person based on personal route patterns

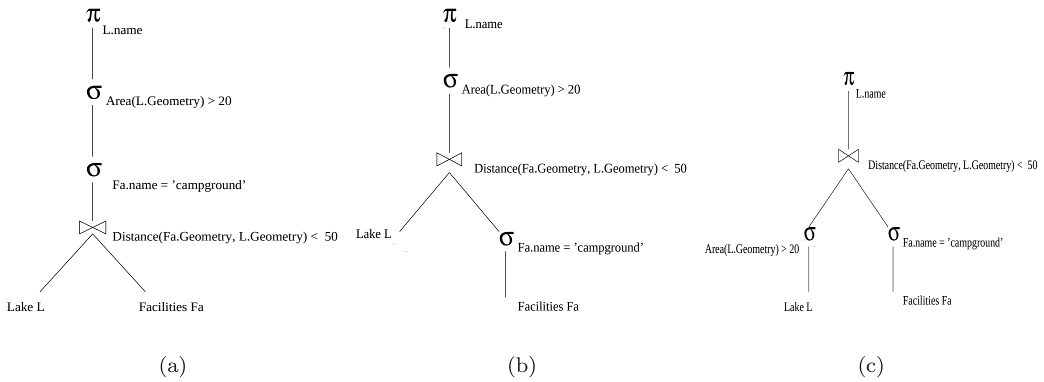


Figure 1.9: (a) Query tree (b) "pushing down": select operation (c) "pushing down" may not help

operations. In the spatial situation, the relative cost per tuple of **Area()** and **Distance()** is an important factor in deciding the order of the operations [49]. Depending upon the implementation of these two functions, the optimal strategy may be to process the join before the select operation(Figure 1.9 (c)). Thus approach thus violates the main heuristic rule for relational databases, which states "Apply select and project before the join and binary operations" are no longer unconditional. There is a cost-based optimization technique to determine the optimal execution strategy from a set of execution plans. A quantitative analysis of spatial index structures is used to calculate the expected number of disk accesses that are required to perform a spatial query [113]. However, query optimization techniques for spatial data need further study.

1.6 Spatial File Organization and Indices

1.6.1 Accomplishments

Space-Filling Curves

The physical design of a spatial database optimizes the instructions to storage devices for performing common operations on spatial data files. File designs for secondary storage include clustering methods as well as spatial hashing methods. Spatial clustering techniques are more difficult to design than traditional clustering techniques because there is no natural order in multidimensional space where spatial data resides. This is only complicated by the fact that the storage disk is a logical one-dimensional device. Thus, what is needed is a mapping from a higher dimensional space to a one-dimensional space which is distance–preserving: so that elements that are close in space are mapped onto nearby points on the line, and one-one: no two points in the space are mapped onto the same point on the line [14]. Several mappings, none of them ideal, have been proposed to accomplish this. The most prominent ones include row-order, Z-order and the Hilbert-curve (Figure 1.10).

Metric clustering techniques use the notion of distance to group nearest neighbors together in a metric space. Topological clustering methods like connectivity clustered access methods [93] use the min-cut partitioning of a graph representation to efficiently support graph traversal operations. The physical organization of files can be supplemented with indices, which are data structures to improve the performance of search operations.

Classical one-dimensional indices such as the B^{+}-tree can be used for spatial data by linearizing a multidimensional space using a space-filling curve such as the Z-order. A large number of spatial indices [91] have been explored for multidimensional euclidean space. Representative indices for point objects include Grid files, multidimensional grid files [65], Point-Quad-Trees, and Kd-trees. Representative indices for extended objects include the R-tree family, the Field-tree, Cell-tree, BSP-tree, and Balanced and Nested grid files.

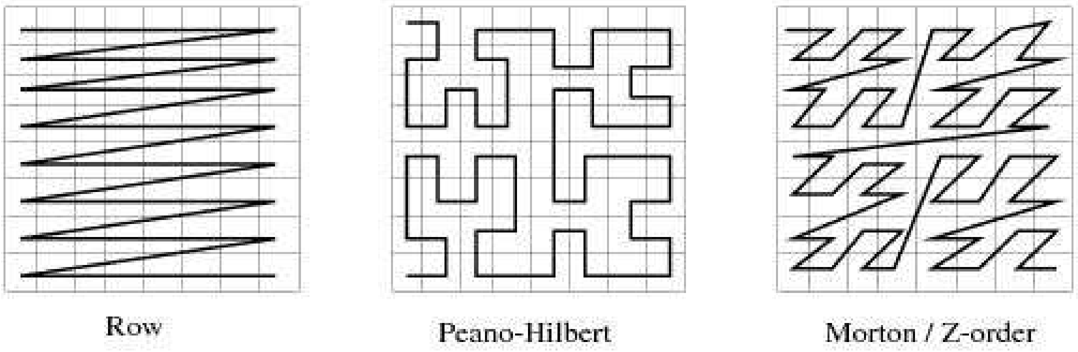


Figure 1.10: Space-Filling curves to Linearize a Multidimensional Space

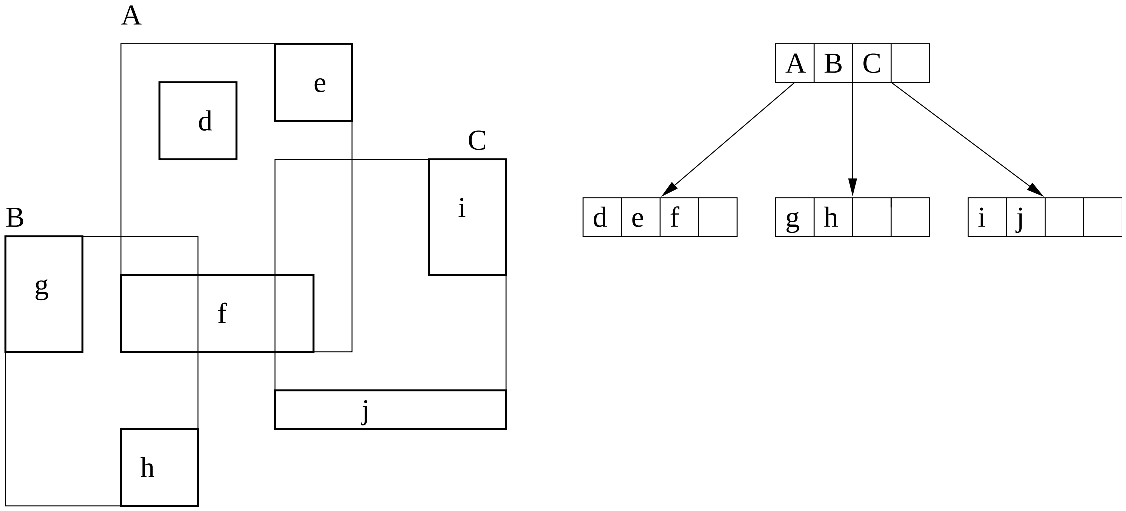


Figure 1.11: Spatial Objects (d, e, f, g, h, i) arranged in R-tree hierarchy

Grid Files

Grid files were introduced by Nievergelt [80]. A grid file divides the space into n-dimensional spaces which can fit into equal-size buckets. The structures are not hierarchical and can be used to index static uniformly distributed data. However, due to its structure the directory of a grid file can be so sparse and large that a large main memory is required. There are several variations of grid files to index data efficiently and to overcome these limitations [81, 118]. An overview of grid files is given in [91].

Tree indexes

R-tree aims to index objects in a hierarchical index structure [44]. The R-tree is a height-balanced tree which is the natural extension of the B-tree for k-dimensions. Spatial objects are represented in the R-tree by their minimum bounding rectangle (MBR). Figure 1.11 illustrates spatial objects organized as a R-tree index. R-trees can be used to process both point and range queries.

Several variants of R-tree exists for better performance of queries and storage utilization. The R$^+$-tree [92] is used to store objects by avoiding overlaps among the MBRs, which increases the performance of the searching. R*-trees [16] relies on combined optimization of area, margin, and overlap of each MBR in intermediate nodes of the tree, that results in better storage utilization.

Many R-tree based index structures [112, 89, 116, 79, 79, 90, 111] have been proposed to index spatio-temporal objects. A survey on spatio-temporal access methods has been provided in [76].

Quad tree [34] is a space-partitioning index structure in which the space is recursively divided into quads. This recursive process is implemented until each quad is homogeneous. There are several variations of quad trees to store point data, raster data, and object data. There are also other quad tree structures to index spatio-temporal datasets, such as Overlapping Linear Quad Trees [115], Multiple Overlapping Features (MOF) trees [73].

The **Generalized Search Tree (GiST)** [48] provides a framework to build almost any kind of tree index on any kind of data. Tree index structures, such as B^+-tree, and R-tree, can be built using GiST. A Spatial-Partitioning Generalized Search Tree (SP-GiST) [11] is an extensible index structure for space-paritioning trees. Index trees such as Quad tree, and *k*d-tree can be built using SP-GiST.

Graph indexes

Most of the spatial access methods provide methods and operators for point and range queries over collections of spatial points, line segments, and polygons. However, it is not clear if spatial access methods can efficiently support network computations which traverse line-segments in a spatial network based on connectivity rather than geographic proximity. **A Connectivity-Clustered Access Medhod for Spatial Network (CCAM)** is proposed to index spatial networks based on graph partitioning [93] by supporting network operations. An auxiliary secondary index, such as B^+-tree, R-tree, and Grid File, is used to support network operations such as $Find()$, $get\text{-}a\text{-}Successor()$, and $get\text{-}Successors()$.

1.6.2 Research Needs

Concurrency Control The R-link tree [61] is among the few approaches available for concurrency control on the R-tree. New approaches for concurrency-control techniques are needed for other spatial indices. Concurrency is provided during operations such as search, insert, and delete. The R-link tree is also recoverable in a write-ahead logging environment. [62] provides general algorithms for concurrency control for GiST, that can also be applied to tree-based indexes. Research is required for concurrency control on other useful spatial data structures.

1.7 Trends: Spatial Data Mining

1.7.1 Accomplishments

The explosive growth of spatial data and widespread use of spatial databases emphasize the need for the automated discovery of spatial knowledge. Spatial data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from spatial databases. Some of the applications are: location-based services, studying the effects of climate, land-use classification, predicting the spread of disease, creating high resolution three-dimensional maps from satellite imagery, finding crime hot spots, and detecting local instability in traffic. A detailed review about spatial data mining can be found in [102].

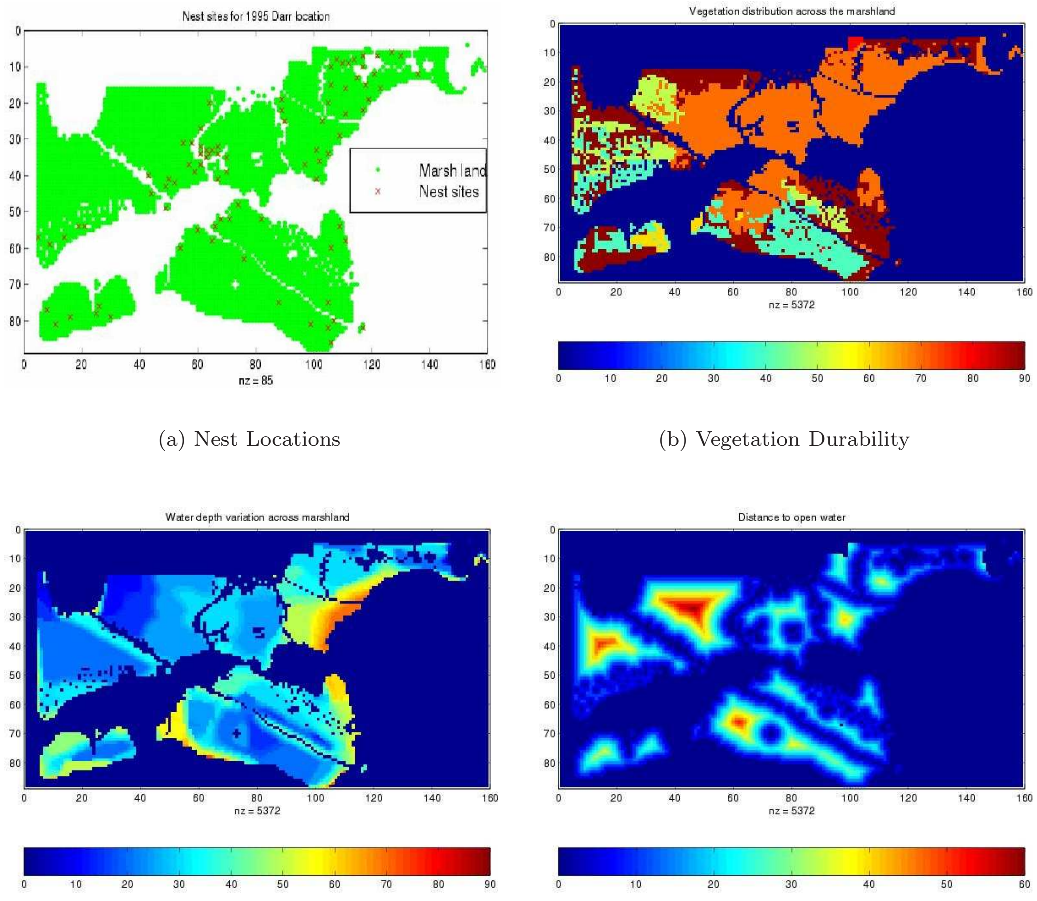
The requirements of mining spatial databases are different from those of mining classical relational databases. The difference between classical and spatial data mining parallels the difference between classical and spatial statistics. One of the fundamental assumptions that guides statistical analysis is that the data samples are independently generated, as with successive tosses of a coin, or the rolling of a die. When it comes to the analysis of spatial data, the assumption about the independence of samples is generally false. In fact, spatial data tends to be highly self-correlated. For example, changes in natural resources, wildlife, and temperature vary gradually over space. The notion of *spatial autocorrelation*, the idea that similar objects tend to cluster in geographic space, is unique to spatial data mining.

For detailed discussion of spatial analysis, readers ar encouraged to refer to [4, 46].

Spatial Patterns

Location Prediction

Location prediction is concerned with the discovery of a model to infer locations of a spatial phenomenon from the maps of other spatial features. For example, ecologists build models to predict habitats for endangered species using maps of vegetation, water bodies, climate, and other related species. Figure 1.12


Nest sites for 1995 Darr location
Marsh land
Nest sites
nz = 85
Vegetation distribution across the marshland
nz = 5372
Water depth variation across marshland
nz = 5372
Distance to open water
nz = 5372
0
10
20
30
40
50
60
70
80
90
100
120
140
160

(a) Nest Locations

(b) Vegetation Durability

(c) Water Depth

(d) Distance to Open Water

Figure 1.12: (a) Learning dataset: The geometry of the Darr wetland and the locations of the nests, (b) The spatial distribution of *vegetation durability* over the marshland, (c) The spatial distribution of *water depth*, and (d) The spatial distribution of *distance to open water*.

shows the learning dataset used in building a location prediction model for red-winged blackbirds in the Darr and Stubble wetlands on the shores of Lake Erie in Ohio. The dataset consists of nest location, vegetation durability, distance to open water and water depth maps. Spatial data mining techniques that capture the spatial auto-correlation [56, 104] of nest location such as the Spatial Autoregression Model (SAR) and Markov Random Fields (MRF) are used for location prediction modeling.

Spatial Autoregression Model Linear regression models are used to estimate the conditional expected value of a dependent variable y given the values of other variables X. Such a model assumes that the variables are independent. The Spatial Autoregression Model [4, 40, 67, 96] is an extension of the linear regression model that takes spatial autocorrelation into consideration. If the dependent values y and X are related to each other, then the regression equation [9] can be modified as

$$\mathbf{y} = \rho \mathbf{W} \mathbf{y} + \mathbf{X}\beta + \epsilon \qquad (1.1)$$

Here $\mathbf{W}$ is the neighborhood relationship contiguity matrix and ρ is a parameter that reflects the strength of the spatial dependencies between the elements of the dependent variable. Notice that when $\rho = 0$, this equation collapses to the linear regression model. If the spatial autocorrelation coefficient is statistically significant, then SAR will quantify the presence of spatial autocorrelation. In such a case, the spatial autocorrelation coefficient will indicate the extent to which variations in the dependent variable (y) are explained by the average of neighboring observation values.

Markov Random Field Markov Random Field-based [68] Bayesian classifiers estimate the classification model, $\hat{f}_C$, using MRF and Bayes' rule. A set of random variables whose interdependency relationship is represented by an undirected graph (i.e., a symmetric neighborhood matrix) is called a Markov Random Field. The Markov property specifies that a variable depends only on its neighbors and is independent of all other variables. The location prediction problem can be modeled in this framework by assuming that the class label, $l_i = f_C(s_i)$, of different locations, s_i, constitutes an MRF. In other words, random variable l_i is independent of l_i if $W(s_i, s_j) = 0$.

The Bayesian rule can be used to predict l_i from feature value vector X and neighborhood class label vector L_i as follows:

$$Pr(l_i|X, L_i) = \frac{Pr(X|l_i, L_i)Pr(l_i|L_i)}{Pr(X)} \tag{1.2}$$

The solution procedure can estimate $Pr(l_i|L_i)$ from the training data, where L_i denotes a set of labels in the neighborhood of s_i excluding the label at s_i, by examining the ratios of the frequencies of class labels to the total number of locations in the spatial framework. $Pr(X|l_i, L_i)$ can be estimated using kernel functions from the observed values in the training dataset.

A more detailed theoretical and experimental comparison of these methods can be found in [32]. Although MRF and SAR classification have different formulations, they share a common goal, estimating the posterior probability distribution. However, the posterior probability for the two models is computed differently with different assumptions. For MRF, the posterior is computed using Bayes' rule, while, in SAR, the posterior distribution is directly fitted to the data.

Spatial Clustering Spatial clustering is a process of grouping a set of spatial objects into clusters so that objects within a cluster have high similarity in comparison to one another, but are dissimilar to objects in other clusters. For example, clustering is used to determine the "hot spots" in crime analysis and disease tracking. Many criminal justice agencies are exploring the benefits provided by computer technologies to identify crime hot spots in order to take preventive strategies such as deploying saturation patrols in hot spot areas.

Spatial clustering can be applied to group similar spatial objects together; the implicit assumption is that patterns in space tend to be grouped rather than randomly located. However, the statistical significance of spatial clusters should be measured by testing the assumption in the data. One of the methods to compute this measure is based on quadrats (i.e., well defined area, often rectangular in shape). Usually quadrats of random location and orientations in the quadrats are counted, and statistics derived from the counters are computed. Another type of statistics is based on distances between patterns; one such type is Ripley's K-function [28]. After the verification of the statistical significance of the spatial clustering, classical clustering algorithms [47] can be used to discover interesting clusters.

Spatial Outliers A spatial outlier [15] is a spatially referenced object whose non-spatial attribute values differ significantly from those of other spatially referenced objects in its spatial neighborhood. Figure 1.13 gives an example of detecting spatial outliers in traffic measurements for sensors on highway I-35W (North bound) for a 24-hour time period. Station 9 seems to be a spatial outlier as it exhibits inconsistent traffic flow as compared with its neighboring stations. The reason could be that the sensor at Station 9 is malfunctioning. Detecting spatial outliers is useful in many applications of geographic information systems and spatial databases, including transportation, ecology, public safety, public health, climatology, and location-based services.

Spatial attributes are used to characterize location, neighborhood, and distance. Non-spatial attribute dimensions are used to compare a spatially referenced object to its neighbors. Spatial statistics literature provides two kinds of bi-partite multidimensional tests, namely graphical tests and quantitative tests. Graphical tests, which are based on the visualization of spatial data, highlight spatial outliers i.e., Variogram clouds [28] and Moran scatterplots [71]. Quantitative methods provide a precise test to distinguish spatial outliers from the remainder of data. A unified approach to detect spatial outliers efficiently is discussed in [99]. [70] provides algorithms for multiple spatial outlier detection.

Spatial Co-location The co-location pattern discovery process finds frequently co-located subsets of spatial event types given a map of their locations. For example, the analysis of the habitats of animals

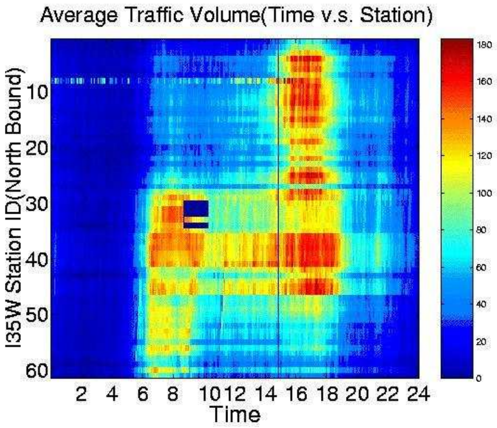


Figure 1.13: Spatial outlier (Station ID 9) in traffic volume data

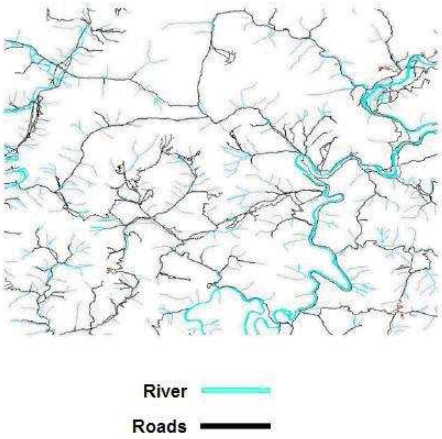


Figure 1.14: Co-location between Roads and Rivers in a Hilly Terrain (Courtesy: Architecture Technology Corporation)

and plants may identify the co-locations of predator-prey species, symbiotic species, or fire events with fuel, ignition sources etc. Figure 1.14 gives an example of the co-location between roads and rivers in a geographic region.

Approaches to discovering co-location rules can be categorized into two classes, namely spatial statistics, and data mining approaches. Spatial statistics-based approaches use measures of spatial correlation to characterize the relationship between different types of spatial features. Measures of spatial correlation include the cross K-function with Monte Carlo simulation, mean nearest-neighbor distance, and spatial regression models.

Data mining approaches can be further divided into transaction-based approaches and distance-based approaches. Transaction-based approaches focus on defining transactions over space so that an Apriori-like algorithm can be used. Transactions over space can be defined by a reference-feature centric model. Under this model, transactions are created around instances of one user-specified spatial feature. The association rules are derived using the *Apriori* [7] algorithm. The rules formed are related to the reference feature. Generalizing the paradigm of forming rules related to a reference feature to the case where no reference feature is specified is non-trivial. Also, defining transactions around locations of instances of all features may yield duplicate counts for many candidate associations.

In a distance-based approach [77, 98, 54], instances of objects are grouped together based on their Euclidean distance from each other. This approach can be considered to be an event-centric model which

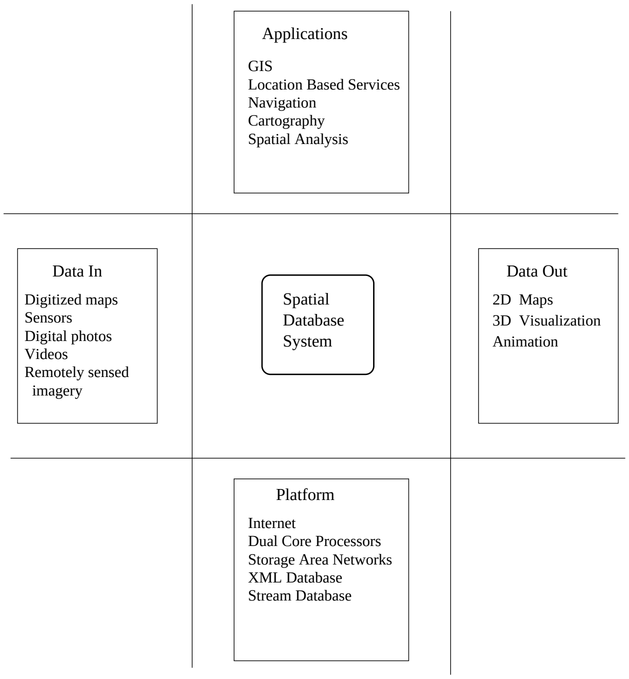


Figure 1.15: Topics driving future research needs in spatial database systems

finds subsets of spatial features likely to occur in a neighborhood around instances of given subsets of event types.

1.7.2 Research Needs

Spatio-temporal Data Mining

Spatio-temporal data mining aims to develop models and objective functions as well as to discover patterns which are more suited to the spatio-temporal databases and their unique properties [87]. An extensive survey of spatio-temporal databases, models and languages, and access methods can be found in [63]. A bibliography of spatio-temporal data mining can be found in [86].

Spatio-temporal pattern mining focuses on discovering knowledge that is frequently located together in space and time. [23, 25, 24] defined the problems of discovering mixed-drove and sustained emerging spatio-temporal co-occurrence patterns and proposed interest measures and algorithms to mine such patterns.

In several application domains such as sensor networks, mobile networks, moving object analysis and image analysis, the need for Spatio-temporal data mining is increasing drastically. It is vital to develop new models and techniques, to define new spatio-temporal patterns, and to formulize monotonic interest measures to mine these patterns [85].

1.8 Summary

In this chapter we presented the major research accomplishments and techniques which have emerged from the area of spatial databases in the past decade. These include spatial database modeling, spatial query processing, and spatial access methods. We have also identified areas where more research is needed, such as spatio-temporal databases, spatial data mining, and spatial networks.

Figure 1.15 provides a summary of topics which continue to drive the research needs of spatial database systems. Increasingly available spatial data in the form of digitized maps, remotely sensed images, spatio-temporal data (for example, from videos), and streaming data from sensors have to be managed and processed efficiently. New ways of querying techniques to visualize spatial data in more than one dimension are needed.

A number of advances have been made in computer hardware over the last few years, but may have yet to be fully exploited, including increases in main memory, more effective storage using Storage Area Networks, greater availability of multi-core processors, and powerful graphic processors. A huge impetus has been spatial data applications such as land navigation systems and location based services. To measure the quality of spatial database systems, new benchmarks have to be established. Some of the benchmarks [109, 83] established earlier have been dated. Newer benchmarks are needed to characterize the spatial data management needs of other systems and applications such as spatio-temporal databases, moving objects database, and location based services.

1.9 Acknowledgements

We thank professional organizations which have funded the research of Spatial Databases, in particular, the National Science Foundation, Army Research Laboratory, Topographic Engineering Center, Oak Ridge National Laboratory, Minnesota Department of Transportation, and Microsoft Corporation. We thank members of the Spatial Database and Spatial Data Mining research group at the University of Minnesota for refining the content of this chapter. We also thank Kim Koffolt for improving the readability of this chapter.

Bibliography

[1] UCGIS, 1998.

[2] CityGML, 2007. http://www.citygml.org/.

[3] Sky Server, 2007. http://skyserver.sdss.org/.

[4] Noel A.C.Cressie. Statistics for Spatial Data . Wiley-Interscience, 1993. ISBN: 978-0471002550.

[5] Nabil R. Adam and Aryya Gangopadhyay. Database Issues in Geographic Information Systems. Kluwer Academic Publishers, Norwell, MA, USA, 1997. ISBN: 0792399242.

[6] A. Aggarwal, B. Chazelle, L. Guibas, C. O'Dunlaing, and C. Yap. Parallel computational geometry. Proceedings of the 25th IEEE Symposium on Foundations of Computer Science, pages 468–477, 1985.

[7] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, Proc. 20th Int. Conf. Very Large Data Bases, VLDB, pages 487–499. Morgan Kaufmann, 12–15 1994.

[8] S. G. Akl and K. A. Lyons. Parallel Computational Geometry. Prentice Hall, Englewood Cliffs, 1993. ISBN: 0136520170.

[9] Luc Anselin. Spatial Econometrics: methods and models. Kluwer, Dordrecht, Netherlands, 1988.

[10] David K. Arctur and Michael Zeiler. Designing Geodatabases. ESRI Press, 2004. ISBN: 158948021X.

[11] Walid G. Aref and Ihab F. Ilyas. SP-GiST: An Extensible Database Index for Supporting Space Partitioning Trees. Journal of Intelligent Information Systems, 17(2-3):215 – 240, December 2001.

[12] C. Arens, J. Stoter, and P. Oosterom. Modeling 3D Spatial Objects in a Geo-DBMS Using a 3D Primitive. Computers and Geosciences, 31(2):165–177, act 2005.

[13] M. P. Armstrong, C. E. Pavlik, and R. Marciano. Experiments in the measurement of spatial association using a parallel supercomputer. Geographical Systems, 1:267–288, 1994.

[14] T. Asano, D. Ranjan, T. Roos, E. Wiezl, and P. Widmayer. Space-filling curves and their use in the design of geometric data structures. Theoretical Computer Science, 181(1):3–15, July 1997.

[15] Vic Barnett and Toby Lewis. Outliers in Statistical Data. John Wiley, 3rd edition edition, 1994.

[16] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. Proc of ACM SIGMOD International Conference on Management of Data, pages 322–331, 1990.

[17] Euro Beinat, Albert Godfrind, and Ravikanth V. Kothuri. Pro Oracle Spatial. Apress, 2004. ISBN: 978-1590593837.

[18] Alberto Belussi and Christos Faloutsos. Estimating the selectivity of spatial queries using the 'correlation' fractal dimension. In Proceedings of the 21st International Conference on Very Large Data Bases, VLDB, pages 299–310, 1995.

[19] Paul Bolstad. GIS Fundamentals: A First Text on Geographic Information Systems, 2nd Edition. Eider Press, 2005. ISBN: 978-0971764712.

[20] Stephan Borzsonyi, Donald Kossmann, and Konrad Stocker. The skyline operator. In Proceedings of the International Conference on Data Engineering, pages 421–430, Heidelberg, Germany, 2001.

[21] Thomas Brinkoff, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. Multi-step processing of spatial joins. In Proceedings of the ACM International Conference on Management of Data, SIGMOD, pages 197–208, 1994.

[22] G. Brunetti, A. Clematis, B. Falcidieno, A. Sanguineti, and M. Spagnuolo. Parallel processing of spatial data for terrain characterization. Proc of the ACM Geographic Information Systems, 1994.

[23] Mete Celik, Shashi Shekhar, James P. Rogers, and James A. Shine. Sustained emerging spatio-temporal co-occurrence pattern mining: A summary of results. 18th IEEE International Conference on Tools with Artificial Intelligence, pages 106–115, 2006.

[24] Mete Celik, Shashi Shekhar, James P. Rogers, James A. Shine, and James M. Kang. Mining at most top-k mixed-drove spatio-temporal co-occurrence patterns: A summary of results. To Appear in Proc. of the Workshop on Spatio-Temporal Data Mining (In conjunction with ICDE 2007), 2007.

[25] Mete Celik, Shashi Shekhar, James P. Rogers, James A. Shine, and Jin Soung Yoo. Mixed-drove spatio-temporal co-occurrence pattern mining: A summary of results. Sixth International Conference on Data Mining, IEEE, pages 119–128, 2006.

[26] D. Chamberlin. Using the New DB2: IBM's Object Relational System. Morgan Kaufmann, 1997. ISBN: 978-1558603738.

[27] K. K. L. Chan and C. D. Tomlin. Map Algebra as a Spatial Language. In D. M. Mark and A. U. Frank, editors, Cognitive and Linguistic Aspects of Geographic Space, pages 351–360. Kluwer Academic Publishers, Dordrecht, 1991. ISBN: 0792315375.

[28] N.A. Cressie. Statistics for Spatial Data (Revised Edition). Wiley, New York, 1993.

[29] Jochen Van den Bercken and Bernhard Seeger. An evaluation of generic bulk loading techniques. In VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases, pages 461–470, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[30] Kristin Eickhorst, Peggy Agouris, and Anthony Stefanidis. Modeling and Comparing Spatiotemporal Events. In Proceedings of the 2004 annual national conference on Digital government research, pages 1–10. Digital Government Research Center, 2004.

[31] Martin Erwig, Ralf Hartmut Guting, Markus Schneider, and Michalis Vazirgiannis. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. GeoInformatica, 3(3):269–296, 1999.

[32] S. Shekhar et al. Spatial Contextual Classification and Prediction Models for Mining Geospatial Data. IEEE Transaction on Multimedia, 4(2), 2002.

[33] M. T. Fang, R. C. T. Lee, , and C. C. Chang. The idea of de-clustering and its applications. Proc. of the International Conference on Very Large Data Bases, pages 181–188, August 25-28 1986.

[34] R. A. Finkel and J. L. Bentley. Quad trees: a data structure for retrieval on composite keys. Acta Informatica, 4:1–9, 1974.

[35] Frederico T. Fonseca and Max J. Egenhofer. Ontology-driven geographic information systems. In Claudia Bauzer Medeiros, editor, ACM-GIS '99, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems, November 2-6, 1999, Kansas City, USA, pages 14–19. ACM, 1999.

[36] W. R. Franklin, C. Narayanaswami, M. Kankanahalli, D. Sun, M. Zhou, and P. Y. F. Wu. Uniform grids: A technique for intersection detection on serial and parallel machines. Proceedings of 9th Automated Cartography, pages 100–109, 1989.

[37] Volker Gaede and Oliver Gunther. Multidimensional access methods. ACM Computing Surveys, 30, 1998.

[38] Betsy George and Shashi Shekhar. Time-aggregated graphs for modeling spatio-temporal networks. In ER (Workshops), pages 85–99, 2006.

[39] Google Earth, 2006. http://earth.google.com.

[40] D.A. Griffith. Advanced Spatial Statistics. Kluwer Academic Publishers, 1998.

[41] Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis. A foundation for representing and querying moving objects. ACM Transactions on Database Systems, 25(1):1–42, 2000.

[42] Ralf Hartmut Guting and Markus Schneider. Moving Objects Databases (The Morgan Kaufmann Series in Data Management Systems) (The Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. ISBN: 0120887991.

[43] R.H. Guting. An introduction to spatial database systems. VLDB Journal, Special Issue on Spatial Database Systems, 3(4):357–399, 1994.

[44] A. Guttman. R-trees: A Dynamic Index Structure for Spatial Searching. Proc. of SIGMOD International Conference on Management of Data, pages 47–57, 1984.

[45] Thanasis Hadzilacos and Nectaria Tryfona. An Extended Entity-Relationship Model for Geographic Applications. ACM SIGMOD Record, 26(3):24–29, 1997.

[46] Robert Haining. Spatial Data Analysis : Theory and Practice . Cambridge University Press, 2003. ISBN: 978-0521773195.

[47] J. Han, M. Kamber, and A. Tung. Spatial Clustering Methods in Data Mining: A Survey, Geographic Data Mining and Knowledge Discovery. Taylor and Francis, 2001.

[48] J. M. Hellerstein, J. F. Naughton, and A. Pfeffer. Generalized Search Trees for Database System. Proceedings of 21th International Conference on Very Large Database Systems, September 11-15 1995.

[49] Joseph M. Hellerstein and Michael Stonebraker. Predicate migration:optimizing queries with expensive predicates. In Proceedings of the ACM-SIGMOD International Conference on Management of Data, pages 267–276, 1993.

[50] Gisli R. Hjaltason and Hanan Samet. Ranking in spatial databases. In Symposium on Large Spatial Databases, pages 83–95, 1995.

[51] E. G. Hoel and H. Samet. Data Parallel RTree Algorithms. Proceedings of the International Conference on Parallel Processing, 1993.

[52] E. G. Hoel and H. Samet. Performance of dataparallel spatial operations. Proc of the 20th International Conference on Very Large Data Bases, pages 156–167, 1994.

[53] Haibo Hu and Dik Lun Lee. Range nearest-neighbor query. IEEE Transactions on Knowledge and Data Engineering, 18(1):78–91, 2006.

[54] Yan Huang, Shashi Shekhar, and Hui Xiong. Discovering Co-locaiton Patterns from Spatial Datasets: A General Approach. IEEE Transaction on Knowledge and Data Engineering, 16(12):1472–1485, December 2005.

[55] Open Geospatial Consortium Inc. OpenGIS Reference Model. http://orm.opengeospatial.org/, 2006.

[56] Yonhong Jhung and Philip H. Swain. Bayesian Contextual Classification Based on Modified M-Estimates and Markov Random Fields. IEEE Transaction on Pattern Analysis and Machine Intelligence, 34(1):67–75, 1996.

[57] W. Kainz, A. Riedl, and G. Elmes, editors. A Tetrahedronized Irregular Network Based DBMS Approach for 3D Topographic Data. Springer Berlin Heidelberg, September 2006. ISBN:978-3-540-35588-5.

[58] W. Kim, J. Garza, and A. Kesin. Spatial data management in database systems. In Advances in Spatial Databases, 3rd International Symposium, SSD'93, volume 652, Springer, ISBN: 3-540-56869-7, pages 1–13, Singapore, 1993.

[59] Ekkehard Köhler, Katharina Langkau, and Martin Skutella. Time-expanded graphs for flow-dependent transit times. In ESA '02: Proceedings of the 10th Annual European Symposium on Algorithms, pages 599–611, London, UK, 2002. Springer-Verlag.

[60] Flip Korn and S. Muthukrishnan. Influence Sets Based on Reverse Nearest Neighbor Queries. In Proceedings of the ACM International Conference on Management of Data, SIGMOD, pages 201–212, 2000.

[61] Marcel Kornacker and Douglas Banks. High-Concurrency Locking in R-Trees. 1995.

[62] Marcel Kornacker, C. Mohan, and Joseph M. Hellerstein. Concurrency and recovery in generalized search trees. In SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, pages 62–72, New York, NY, USA, 1997. ACM Press.

[63] Manolis Koubarakis, Timos K. Sellis, Andrew U. Frank, Stephane Grumbach, Ralf Hartmut Guting, Christian S. Jensen, Nikos A. Lorentzos, Yannis Manolopoulos, Enrico Nardelli, Barbara Pernici, Hans-Jörg Schek, Michel Scholl, Babis Theodoulidis, and Nectaria Tryfona, editors. Spatio-Temporal Databases: The CHOROCHRONOS Approach, volume 2520 of Lecture Notes in Computer Science. Springer, 2003. ISBN: 3540405526.

[64] V. Kumar, A. Grama, and V. N. Rao. Scalable load balancing techniques for parallel computers. Journal of Parallel and Distributed Computing, 22(1):60–69, July 1994.

[65] J. Lee, Y. Lee, K. Whang, and I. Song. A physical database design method for multidimensional file organization. Information Sciences, 120(1):31–65(35), November 1997.

[66] Min-Jae Lee, Senior Member-Kyu-Young Whang, Member-Wook-Shin Han, and Member-Il-Yeol Song. Transform-space view: Performing spatial join in the transform space using original-space indexes. IEEE Transactions on Knowledge and Data Engineering, 18(2):245–260, 2006.

[67] J. LeSage. Spatial Econometrics. 1998. http://www.spatial-econometrics.com/.

[68] S.Z. Li. A Markov Random Field Modeling. Computer Vision, Springer Verlag, 1995.

[69] D. R. Liu and S. Shekhar. A similarity graph-based approach to declustering problem and its applications. Proc of the Eleventh International Conference on Data Engineering, IEEE, 1995.

[70] Chang-Tien Lu, Dechang Chen, and Yufeng Kou. Algorithms for Spatial Outlier Detection. IEEE International Conference on Data Mining, 2003.

[71] Anselin Luc. Local Indicators of Spatial Association: LISA. Geographical Analysis, 27(2):93–115, 1995.

[72] Nikos Mamoulis and Dimitris Papadias. Slot index spatial join. IEEE Transactions on Knowledge and Data Engineering, 15(1):211–231, 2003.

[73] Yannis Manolopoulos, Enrico Nardelli, Apostolos Papadopoulos, and Guido Proietti. MOF-Tree: A Spatial Access Method to Manipulate Multiple Overlapping Features. Information Systems, 22(9):465–481, December 1997.

[74] Yannis Manolopoulos, Apostolos N. Papadopoulos, and Michael Gr Vassilakopoulos. Spatial Databases: Technologies, Techniques and Trends . Idea Group Publishing, 2004. ISBN: 978-1591403876.

[75] Microsoft Virtual Earth, 2006. http://www.microsoft.com/virtualearth.

[76] Mohamed F. Mokbel, Thanaa M. Ghanem, and Walid G. Aref. Spatio-temporal access methods. IEEE Data Engineering Bulletin, 26(2):40–49, June 2003.

[77] Y. Morimoto. Mining Frequent Neighboring Class Sets in Spatial Databases. In Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001.

[78] Alexandros Nanopoulos, Yannis Theodoridis, and Yannis Manolopoulos. C2P: Clustering based on Closest Pairs. In Proceedings of the International Conference on Very Large Data Bases, VLDB, pages 331–340, 2001.

[79] M. Nascimiento, R. Jefferson, J. Silva, and Y. Theodoridis. Evaluation of access structures for discretely moving points. Proc of the International Workshop on Spatio-temporal Database Management, pages 171–188, 1999.

[80] J. Nievergelt, H. Hinterberger, and K.C. Sevcik. The grid file: An adaptable, symmetric multikey file structure. ACM Trancsactions on Database Systems, 9(1):38–71, March 1984.

[81] M. Ouksel. The interpolation-based grid file. Proc of Fourth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, pages 20–27, 1985.

[82] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. Aggregate nearest neighbor queries in spatial databases. ACM Trans. Database Systems, 30(2):529–576, 2005.

[83] Jignesh M. Patel, Jie-Bing Yu, Navin Kabra, Kristin Tufte, Biswadeep Nag, Josef Burger, Nancy E. Hall, Karthikeyan Ramasamy, Roger Lueder, Curt Ellmann, Jim Kupsch, Shelly Guo, David J. DeWitt, and Jeffrey F. Naughton. Building a scaleable geo-spatial dbms: Technology, implementation, and evaluation. In ACM SIGMOD Conference, pages 336–347, 1997.

[84] Philippe Rigaux, Michel Scholl, and Agns Voisard. Spatial Databases: With Application to GIS. Morgan Kaufmann Series in Data Management Systems, 2000. ISBN: 9781558605886.

[85] John F. Roddick, Erik Hoel, Max J. Egenhofer, Dimitris Papadias, and Betty Salzberg. Spatial, temporal and spatio-temporal databases - hot issues and directions for phd research. SIGMOD record, 33(2), June 2004.

[86] John F. Roddick, Kathleen Hornsby, and Myra Spiliopoulou. An updated bibliography of temporal, spatial, and spatio-temporal data mining research. Proc of the First International Workshop on Temporal, Spatial and Spatio-temporal Data Mining, pages 147–164, 2001.

[87] John F. Roddick and Brian G. Lees. Paradigms for spatial and spatio-temporal data mining. Taylor and Frances, 2001.

[88] Nick Roussopoulos, Stephen Kelley, and Frédéric Vincent. Nearest neighbor queries. In SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data, pages 71–79, New York, NY, USA, 1995. ACM Press.

[89] S. Saltenis and C.S. Jensen. R-tree based indexing of general spatio-temporal data. Technical Report TR-45 and Chorochronos CH-99-18, TimeCenter, 1999.

[90] Simonas Saltenis, Christian S. Jensen, Scott T. Leutenegger, and Mario A. Lopez. Indexing the positions of continuously moving objects. In SIGMOD Conference, pages 331–342, 2000.

[91] Hanan Samet. Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann Publishers, 2006. ISBN: 0123694469.

[92] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R$^+$-tree: A dynamic index for multidimensional objects. Proc. 13th International Conference on Very Large Data Bases, pages 507–518, September 1987.

[93] S. Shekhar and D.R. Liu. A connectivity-clustered access method for networks and network computation. IEEE Transactions on Knowledge and Data Engineering, 9(1):102–119, January 1997.

[94] S. Shekhar and X. Liu. Direction as a Spatial Object: A Summary of Results. In R. Laurini, K. Makki, and N. Pissinou, editors, ACM-GIS '98, Proceedings of the 6th international symposium on Advances in Geographic Information Systems, November 6-7, 1998, Washington, DC, USA, pages 69–75. ACM, 1998.

[95] S. Shekhar, S. Ravada, V. Kumar, D. Chubband, and G. Turner. Declustering and load-balancing methods for parallelizing spatial databases. IEEE Transactions on Knowledge and Data Engineering, 10(4):632 – 655, July 1998.

[96] S. Shekhar, P. Schrater, R. Raju, and W. Wu. Spatial contextual classification and prediction models for mining geospatial data. IEEE Transactions on Multimedia, 4(2):174–188, 2002.

[97] Shashi Shekhar and Sanjay Chawla. Spatial Databases: A Tour. Prentice Hall, 2002. ISBN: 978-0130174802.

[98] Shashi Shekhar and Yan Huang. Co-location Rules Mining: A Summary of Results. Proceedings of Symposium on Spatial and Spatio-temporal Databases, 2001.

[99] Shashi Shekhar, Chang-Tien Lu, and Pusheng Zhang. A unified approach to detecting spatial outliers. GeoInformatica, 7(2), 2003.

[100] Shashi Shekhar, Ranga Raju Vatsavai, Sanjay Chawla, and Thomas E. Burk. Spatial pictogram enhanced conceptual data models and their translation to logical data models. Lecture Notes in Computer Science, 1737:77–104, 2000. ISBN 3-540-66931-0.

[101] Shashi Shekhar and Hui Xiong. Encyclopedia of GIS. Springer, 2007 (expected). ISBN: 9780387308586.

[102] Shashi Shekhar, Pusheng Zhang, Yan Huang, and Ranga Raju Vatsavai. Spatial data mining. In Hillol Kargupta and Anupam Joshi, editors, Book Chapter in Data Mining: Next Generation Challenges and Future Directions.

[103] Spiros Skiadopoulos, Christos Giannoukos, Nikos Sarkas, Panos Vassiliadis, Timos Sellis, and Manolis Koubarakis. Computing and managing cardinal direction relations. IEEE Transactions on Knowledge and Data Engineering, 17(12):1610–1623, 2005.

[104] A. H. Solberg, Torfinn Taxt, and Anil K. Jain. A Markov Random Field Model for Classification of Multisource Satellite Imagery. IEEE Transaction on Geoscience and Remote Sensing, 34(1):100–113, 1996.

[105] R. Sridhar, S. S. Iyengar, and S. Rajanarayanan. Range search in parallel using distributed data structures. International Conference on Databases, Parallel Architectures, and Their Applications, pages 14–19, 1990.

[106] S.Shekhar, R.R. Vatsavai, S. Chawla, and T.E. Burke. Spatial Pictogram Enhanced Conceptual Data Models and Their Translations to Logical Data Models. Integrated Spatial Databases: Digital Images and GIS, Lecture Notes in Computer Science, 1737:77–104, 1999.

[107] Ioana Stanoi, Divyakant Agrawal, and Amr ElAbbadi. Reverse Nearest Neighbor Queries for Dynamic Databases. In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pages 44–53, 2000.

[108] M. Stonebraker and D. Moore. Object Relational DBMSs: The Next Great Wave. Morgan Kaufmann, 1997. ISBN: 978-1558603974.

[109] Michael Stonebraker, James Frew, Kenn Gardels, and Jeff Meredith. The Sequoia 2000 Benchmark. In Peter Buneman and Sushil Jajodia, editors, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993, pages 2–11. ACM Press, 1993.

[110] Michael Stonebraker and Greg Kemnitz. The Postgres Next Generation Database Management System. Commun. ACM, 34(10):78–92, 1991.

[111] Yufei Tao, Dimitris Papadias, and Jimeng Sun. The TPR*-Tree: An Optimized Spatio-Temporal Access Method for Predictive Queries. In VLDB, pages 790–801, 2003.

[112] Y. Theodoridis, M. Vazirgiannis, and T. Sellis. Spatio-temporal indexing for large multimedia applications. International Conference on Multimedia Computing and Systems, pages 441–448, June 1996.

[113] Yannis Theodoridis and Timos Sellis. A model for the prediction of r-tree performance. In Proceedings of the 15th ACM Symposium on Principles of Database Systems, PODS, Symposium, pages 161–171. ACM, 1996.

[114] Yannis Theodoridis, Emmanuel Stefanakis, and Timos Sellis. Cost models for join queries in spatial databases. In Proceedings of the IEEE 14th International Conference on Data Engineering, pages 476–483, 1998.

[115] Theodoros Tzouramanis, Michael Vassilakopoulos, and Yannis Manolopoulos. Overlapping linear quadtrees: A spatio-temporal access method. ACM-Geographic Information Systems, pages 1–7, 1998.

[116] M. Vazirgiannis, Y. Theodoridis, , and T. Sellis. Spatio-temporal composition and indexing large multimedia applications. Multimedia Systems, 6(4):284–298, July 1998.

[117] F. Wang. A parallel intersection algorithm for vector polygon overlay. IEEE Computer Graphics and Applications, 13(2):74–81, March 1993.

[118] K. Y. Whang and R. Krishnamurthy. Multilevel grid files. IBM Research Laboratory, Yorktown Heights, NY, 1985.

[119] Wikipedia, 2007. http://en.wikipedia.org/wiki/Spatial_Database.

[120] Michael Worboys and Matt Duckham. GIS: A Computing Perspective. Second Edition. CRC, 2004. ISBN: 978-0415283755.

[121] Tian Xia and Donghui Zhang. Continuous Reverse Nearest Neighbor Monitoring. In Proceedings of the International Conference on Data Engineering, ICDE, 2006.

[122] Man Lung Yiu, Nikos Mamoulis, and Dimitris Papadias. Aggregate nearest neighbor queries in road networks. IEEE Transactions on Knowledge and Data Engineering, 17(6):820–833, 2005.

[123] Y. Zhou, S. Shekhar, and M. Coyle. Disk allocation methods for parallelizing grid files. Proc of the Tenth International Conference on Data Engineering, IEEE, pages 243–252, 1994.

[124] Manli Zhu, Dimitris Papadias, Jun Zhang, and Dik Lun Lee. Top-k spatial joins. IEEE Transactions on Knowledge and Data Engineering, 17(4):567–579, 2005.