Spatial Databases— Accomplishments and Research Needs

Shashi Shekhar, *Senior Member*, *IEEE*, Sanjay Chawla,
Siva Ravada, *Member*, *IEEE*, Andrew Fetterer, *Member*, *IEEE*,
Xuan Liu, *Student Member*, *IEEE*, Chang-tien Lu, *Student Member*, *IEEE*

Abstract—Spatial databases, addressing the growing data management and analysis needs of spatial applications such as Geographic Information Systems, have been an active area of research for more than two decades. This research has produced a taxonomy of models for space, spatial data types and operators, spatial query languages and processing strategies, as well as spatial indexes and clustering techniques. However, more research is needed to improve support for network and field data, as well as query processing (e.g., cost models, bulk load). Another important need is to apply spatial data management accomplishments to newer applications, such as data warehouses and multimedia information systems. The objective of this paper is to identify recent accomplishments and associated research needs of the near term.

Index Terms—Spatial databases, multidimensional, object-relational, databases, Geographic Information Systems.

1 Introduction

1.1 Spatial Databases

Spatial database [11], [15], [35] management systems aim at the effective and efficient management of data related to

- a space such as the physical world (geography, urban planning, astronomy);
- parts of living organisms (anatomy of the human body);
- engineering design (very large scale integrated circuits, the design of an automobile, or the molecular structure of a pharmaceutical drug); and
- conceptual information space (a multidimensional decision support system, fluid flow, or an electromagnetic field).

The field of spatial database research has been an active area of research for more than two decades. The results of this research, e.g., spatial multidimensional indexes, are being used in a number of areas. The field of spatial databases can be defined by its accomplishments; current research is aimed at improving its functionality and its performance. The impetus for improving functionality comes from the needs of existing applications such as Geographic Information Systems (GIS) and Computer Aided Design (CAD), as well as from potential applications such as Multimedia Information System (MMIS), Data Warehousing (DWH), and NASA's Earth Observation System (EOS). The acceptance of GIS as an important tool in governmental decision-making is also documented [34], and military planners have embraced GIS technology at all levels of tactical, operational and strategic planning, including battlefied visualization and terrain analysis [20].

Commercial examples of spatial database management include Informix's spatial data-blades (i.e., 2D, 3D, Geodetic), Oracle's Universal server with either Spatial Data Option or Spatial Data Cartridge and ESRI's Spatial Data Engine (SDE). Research prototype examples of spatial database management systems include spatial datablades with Postgres [30], Predator, and Paradise [9]. The functionalities provided by these systems include a set of spatial data types such as a point, line-segment and polygon, and a set of spatial operations such as inside, intersection, and distance. The spatial types and operations may be made part of a query language such as SQL, which allows spatial querying when combined with an object-relational database management system [6], [32]. The performance enhancement provided by these systems includes a multidimensional spatial index and algorithms for spatial access methods, spatial range queries, and spatial joins. Spatial indexing with concurrency control may be implemented in the object-relational server for performance reasons.

Existing and emerging applications require new functionalities including the modeling of network spaces and continuous fields. The performance needs of emerging applications require not only the management of large data sets, but also new processing strategies for spatial set-operations, field operations (e.g., slope), and network analysis (e.g., shortest-path, route-evaluation).

1.2 Related Work and Our Contributions

Recent reports [11], [15], [35], [1] have described the accomplishments of spatial database research and have prioritized research needs. A broad survey of spatial database requirements and an overview of research results

- *S. Shekhar, S. Chawla, X. Liu, and C.-t. Lu are with the Computer Science Department, University of Minnesota, 200 Union St. SE, Minneapolis, MN 55455. E-mail: {shekhar, chawla, xliu, ctlu}@cs.umn.edu.*
- *S. Ravada is with the Oracle Corporation.*
- *A. Fetterer Panttaja Consulting Group in San Francisco.*

Manuscript received 3 June 1997; revised 13 Aug. 1998.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 108311.

is provided in [35], [11], [1]. Basic modeling requirements for spatial objects such as points, lines, and polygons are given in terms of their geometry, topology and object relationships (topological, directional, metric, network). Requirements are given for other user-level issues such as graphical input and output and query language support. Spatial clustering and indexing techniques [23] such as Grid-files, Z-order, Quad-tree, Kd-trees, R-trees [12], and associated join strategies are described. Finally, an architecture for spatial databases is given in terms of the object-relational model.

Research needed to improve the performance of spatial databases in the context of object-relational databases was listed in [15]. The primary research needs identified were concurrency control techniques for spatial indexing methods, the development of cost models for query strategies, and the development of new spatial join algorithms beyond nested-loop and tree matching.

Many of the research needs identified in [15] have since been addressed. For example, concurrency control techniques for R-trees have been studied in the context of R-link [16] trees. Also, new spatial join strategies using space partitioning [22] have been explored. In this paper, we identify the recent accomplishments in spatial databases as well as current research needs, based on publications in journals and conference proceedings and recent commercial trends.

1.3 Scope and Outline

The role of the spatial database component is dependent on the type of database management system (DBMS) involved: relational, object-oriented or object-relational. In this paper, we focus the discussion of spatial databases in the context of the object-relational [6], [32], [31] databases, which provide extensibility to many components of traditional databases to support new application domains. These and other important issues including architectural options, Raster DBMS and Network spaces are covered in detail in our forthcoming book [24]. Spatial databases have been one of the most common applications of object-relational databases and have influenced their design a great deal. Object-relational databases allow the inclusion of spatial datatypes, spatial operations, and multidimensional indexing systems. This three-layer architectural framework is shown in Fig. 1, and it consists of an object-relational database management system, a spatial database, and a spatial application such as a GIS or MMIS. The interface between the application and the spatial data system maps application-specific constructs to the spatial database. The spatial database associates the application requirements to the functionality provided by the DBMS. The interface to the DBMS supports specialized query processing, which in turn supports the core database requirements for achieving acceptable performance.

Emerging trends such as World Wide Web interfaces, multimedia data, and image processing are likely to impact the data sharing and analysis needs of spatial databases. Scaling up to large datasets requires new research in many areas beyond spatial databases, including research on file-systems, device-drivers for tertiary storage, computer networks, and visualization software and algorithms related to graphics and computational geometry. This paper does not explore those issues.

The remainder of the paper is organized as follows: Section 2 describes the recent advances in spatial databases. Section 3 states the research needs for spatial databases. Section 4 highlights our conclusions and motivates exploration of applications whose needs are not currently met by spatial databases.

2 Accomplishments

Research into spatial databases has mainly focused on developing a space taxonomy, spatial data models, spatial query languages and processing strategies, and spatial access methods. This section lists recent important accomplishments, not only for the current applications of spatial databases, but also for the emerging database problems that have spatial dimensions.

2.1 Space Taxonomy

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 [35]. 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 relationships 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 = 1 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.

2.2 Spatial Data Model and Query Language

A spatial data model [25], [35] is a type of data-abstraction that hides the details of data-storage. There are two common models of spatial information: field-based and object-based. The field-based model treats spatial information such as altitude, rainfall and temperature as a collection of spatial functions transforming a space-partition to an attribute domain. The object-based model treats the information space as if it is populated by discrete, identifiable, spatially referenced entities. The operations on spatial objects include *distance* and *boundary*. The operations on fields include local, focal, and zonal operations, as shown in Table 2. The fields may be continuous, differentiable, discrete, and

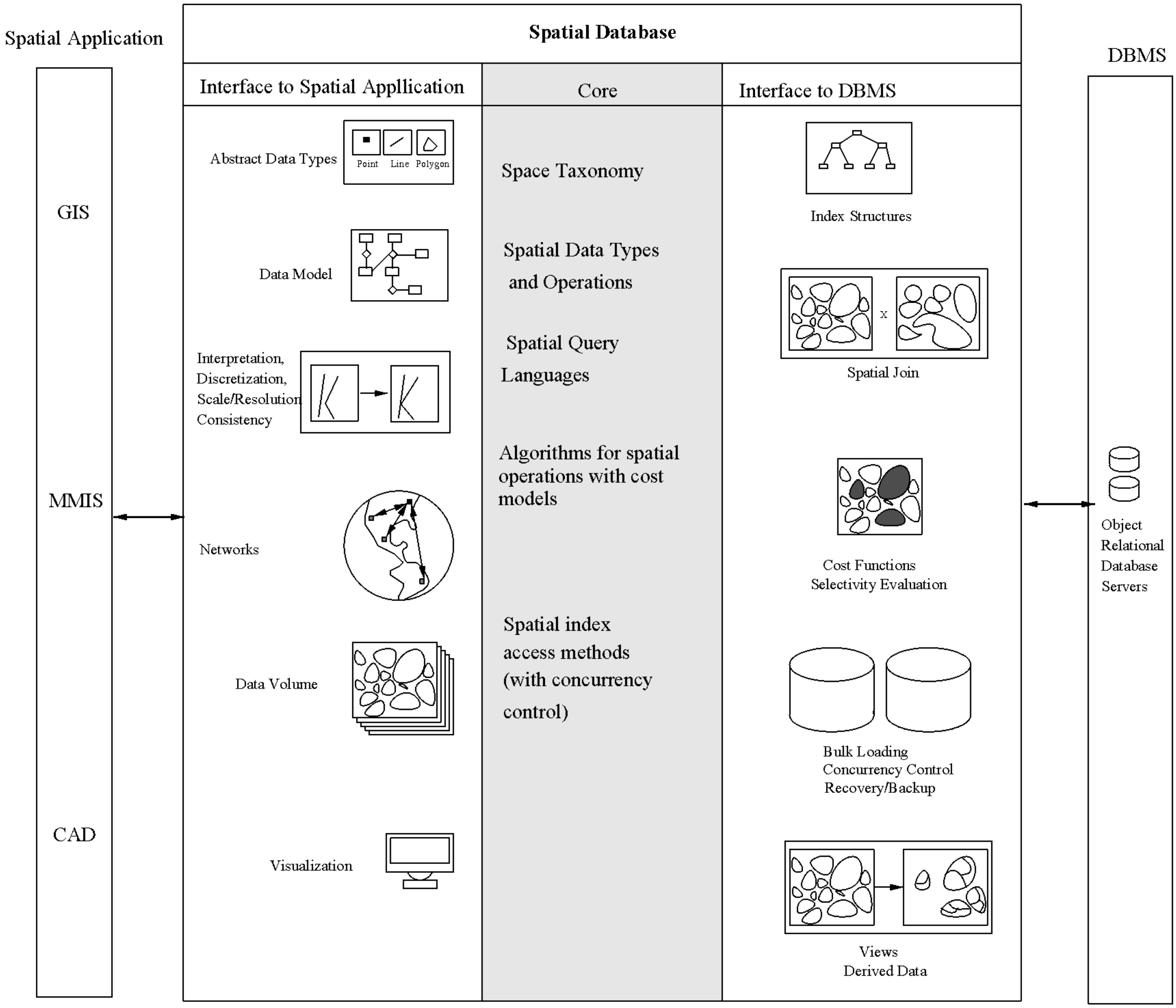


Fig. 1. Three-layer architecture.

isotropic or anisotropic, with positive or negative autocorrelation. Certain field operations (slope or interpolation) assume certain field properties (differentiable or positive autocorrelation).

An implementation of a spatial data model in the context of object-relational databases consists of a set of spatial data types and the operations on those types. Much work has been done over the last decade on the design of spatial Abstract Data Types (ADTs) and their embedding in a query language. Consensus is slowly emerging via standardization efforts, and recently the OGIS consortium [21] has proposed a specification for incorporating 2D geospatial ADTs in SQL. Fig. 3, which illustrates this spatial data-type hierarchy consists of *Point*, *Curve*, and *Surface* classes and a parallel class of *Geometry Collection*. The basic operations operative on *all* datatypes are shown in Table 1. The topological operations are based on the ubiquitous nine-intersection model [10]. Using the OGIS specification, common spatial queries can be intutively posed in SQL. For example, the query *Find all lakes which have an area greater than 5 sq. km. and are within 20 km. from the campgrounds* can be posed as shown in Fig. 2a.

Other example GIS queries which can be implemented using OGIS operations are provided in Table 3. The OGIS specification is confined to topological and metric operations on vector data types. Other interesting classes of operations are network, direction, dynamic and the field operations of focal, local and zonal (see Table 2). While standards for field based raster data types are still emerging, Map Algebra [33], specifically designed for cartographic modeling and RaSQL, based on Image Algebra [3], for general multidimensional discrete objects(satellite images, X-rays, etc.), are important milestones.

2.3 Spatial Query Processing

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-area (NAA) model, the doubly connected-edge-list (DCEL), and boundary representation [17], some of

TABLE 1
REPRESENTATIVE FUNCTIONS SPECIFIED BY OGIS [21]

Basic Functions	SpatialReference()	Returns the Reference System of the geometry
	Envelope()	The minimum bounding rectangle of the geometry
	Export()	Convert the geometry into a different representation.
	IsEmpty()	Tests if the geometry is a empty set or not
	IsSimple()	Returns True if the geometry is simple(no self-intersection)
	Boundary()	Returns the boundary of the geometry
Topological/ Set Operators	Equal	Tests if the geometries are spatially equal
	Disjoint	Tests if the geometries are disjoint
	Intersect	Tests if the geometries intersect
	Touch	Tests if the geometries touch each other
	Cross	Tests if the geometries cross each other
	Within	Tests if the given geomtry is within another given geometry
	Contains	Tests if the given geometry contains another given geometry
	Overlap	Tests if the geometry overlaps another geometry
Spatial Analysis	Distance	Returns the shortest distance between two geometries
	Buffer	Returns a geometry that represents all points whose distance from the given is less than or equal to the specified distance
	ConvexHull	Returns the convex hull of the geometry
	Intersection	Returns the intersection of two geometries
	Union	Returns the union of two geometries
	Difference	Returns the difference of two geometries
	SymDiff	Returns the symmetric difference of two geometries

TABLE 2
A SAMPLE OF SPATIAL OPERATIONS

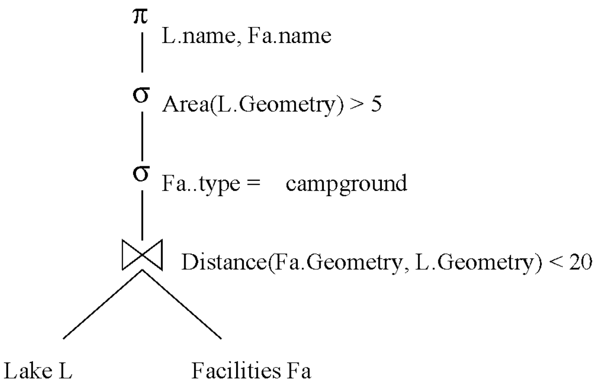
Data model	Operator Group	Operation
Vector Object	Set-Oriented	equals, is a member of, is empty, is a subset of, is disjoint from, intersection, union, difference, cardinality
	Topological	boundary, interior, closure, meets, overlaps, is inside, covers, connected, components, extremes, is within
	Metric	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 3
TYPICAL SPATIAL QUERIES FROM GIS

Single Table Queries	
Grouping	Recode all land with silty soil to silt-loam 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
Multi-Table Queries	
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

```
SELECT   L.name, Fa.name
FROM     Lake L, Facilities Fa
WHERE    Area(L.Geometry) > 5 AND
         Fa.type =   campground    AND
         Distance(Fa.Geometry, L.Geometry) <  20
```

(a)



(b)

Fig. 2: (a) SQL query with spatial operators; (b) corresponding query tree.

which are shown in Fig. 4 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).

The spatial queries [7], shown in Table 3, are often processed using *filter* and *refine* techniques. Approximate geometry such as 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. Strategies for range-queries include a scan and index-search in conjunction with the plane-sweep algorithm [5]. Strategies for the spatial-join include the nested loop, tree matching [5], when indices are present on all participating relations, and space partitioning [22], in the absence of indices. To speed up computation for large spatial objects (it is common for polygons to have 1,000 or more edges), object indices are used in extended filtering. Strategies such as object approximation and tree matching originated in spatial-databases, and can potentially be applied in other domains with similar characteristics.

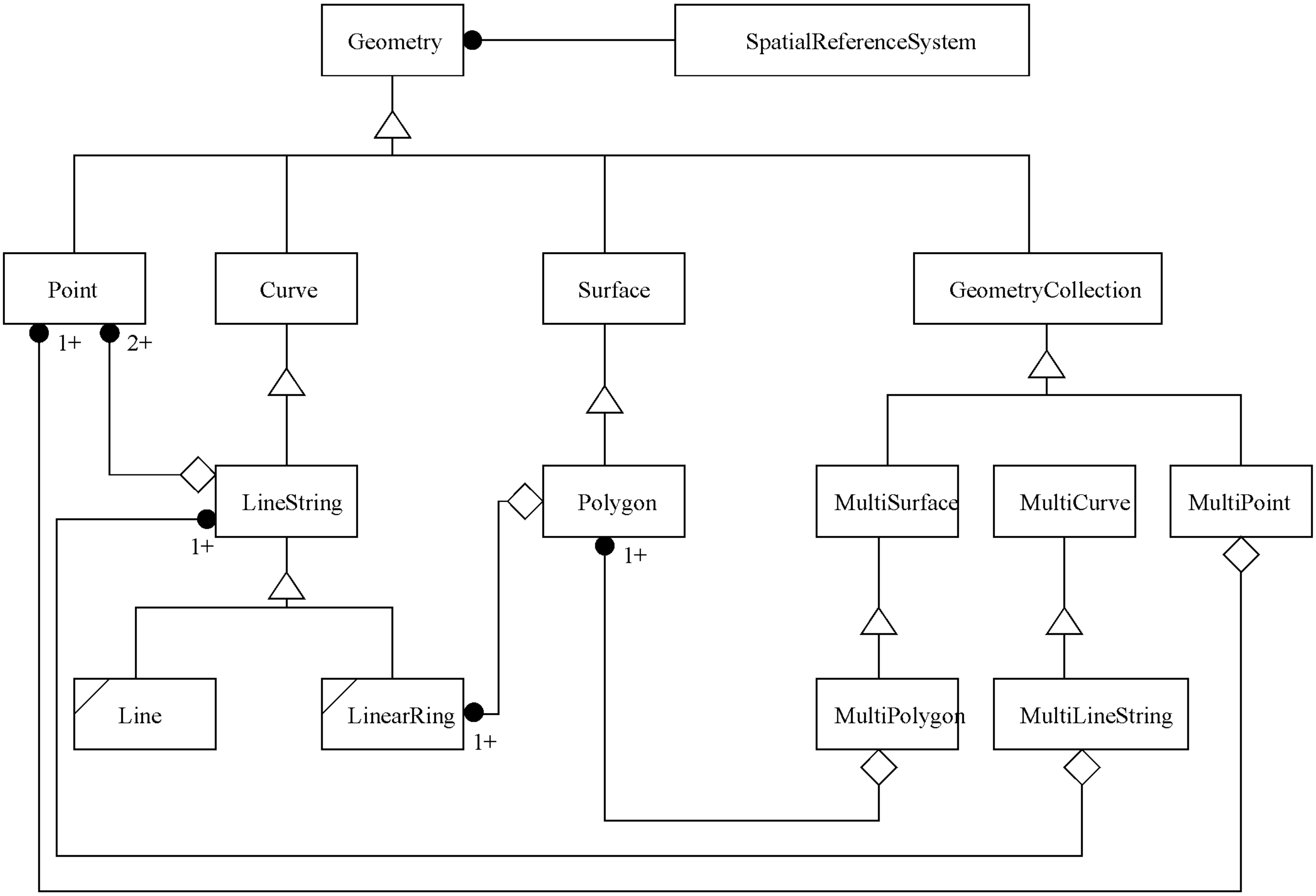


Fig. 3. Spatial data type hierarchy [21].

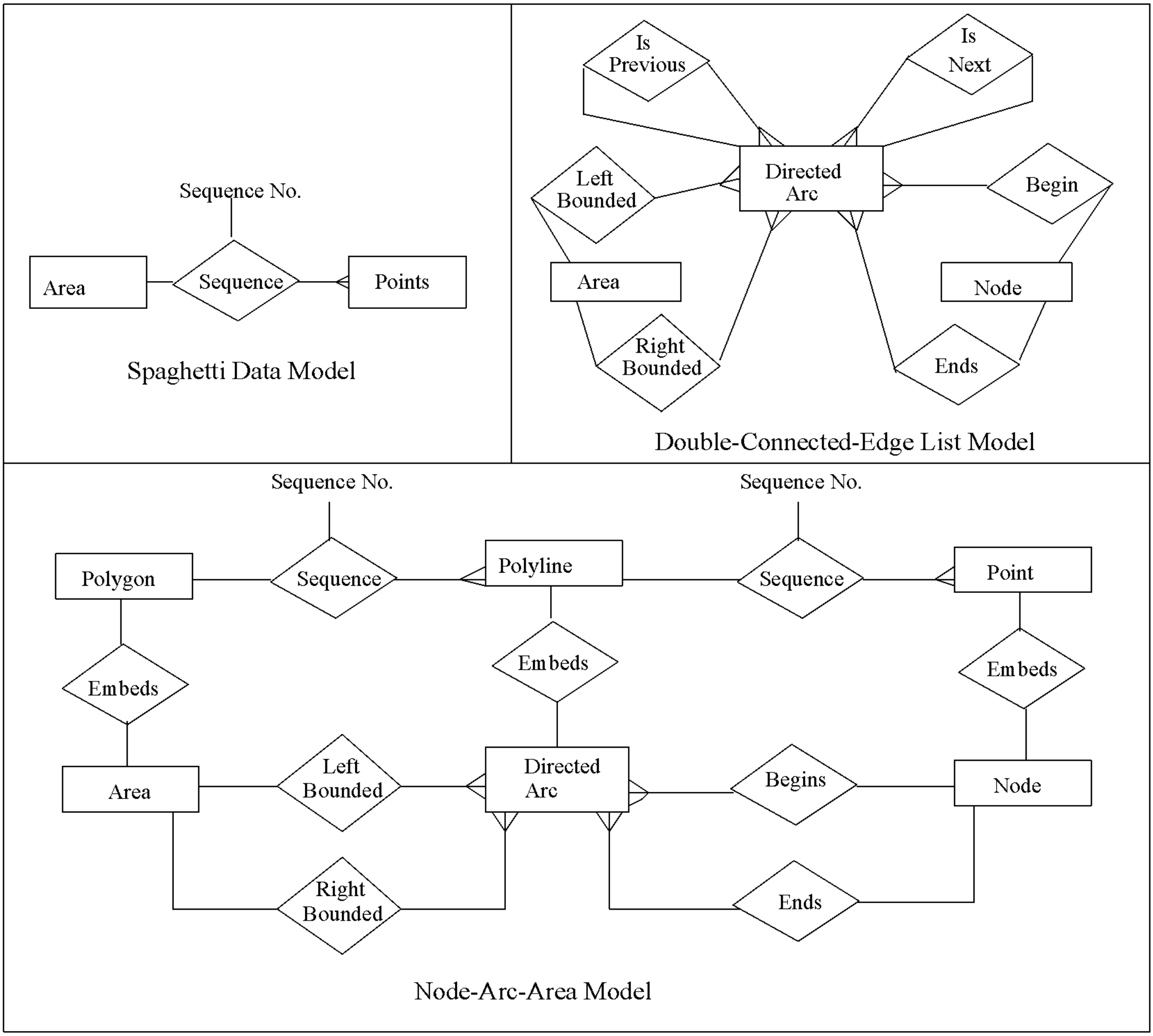


Fig. 4. Entity relationship diagrams for common representations of spatial data.

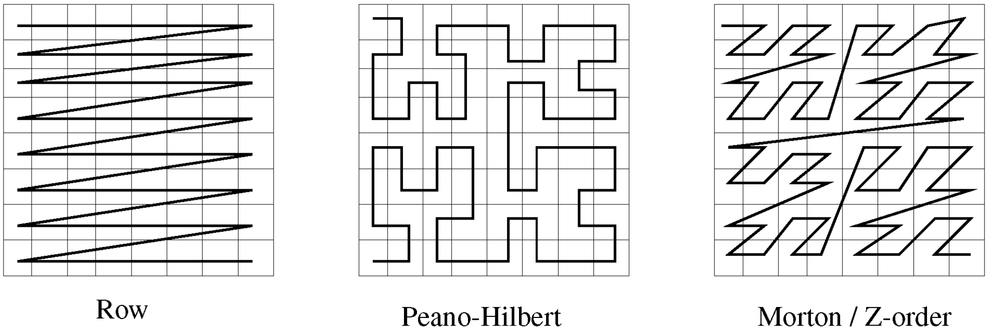


Fig. 5. Space-filling curves to linearize a multidimensional space.

2.4 Spatial File Organization and Indices

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. The design of spatial clustering techniques is more difficult compared to the design of traditional clustering 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 that 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 [2]. 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* (see Fig. 5).

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 [27] 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 (see Fig. 5). A large number of spatial indices [23] have been explored for multidimensional Euclidean space. Representative indices for point objects include Grid files, multidimensional grid files [18], 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.

One of the first access methods created to handle extended objects was Guttman's R-tree structure [12]. The R-tree is a height balanced natural extension of the B+ tree for higher dimensions. Objects are represented in the R-tree by their minimum bounding rectangles (MBRs). Nonleaf nodes are composed of entries of the form (*R*, *child–pointer*), where *R* is the MBR of all entries contained in the *child-pointer*. Leaf nodes contain the MBRs of the data objects. To guarantee good space utilization and height-balance, the parent MBRs are allowed to overlap. Fig. 6a illustrates the spatial objects organized in an R-tree, while Fig. 6b shows the file structure where the nodes correspond to disk pages. Many variations of the R-tree structure exist whose main emphasis is on discovering new strategies to maintain the balance of the tree in case of a split and to minimize the overlap of the MBRs in order to improve the search time.

Concurrency control for spatial access methods [16] is provided by the R-link tree, which is a variant of the R-tree with additional sibling pointers that allow the tracking of modifications. Concurrency is provided during operations such as search, insert, and delete. The R-link tree is also recoverable in a write-ahead logging environment.

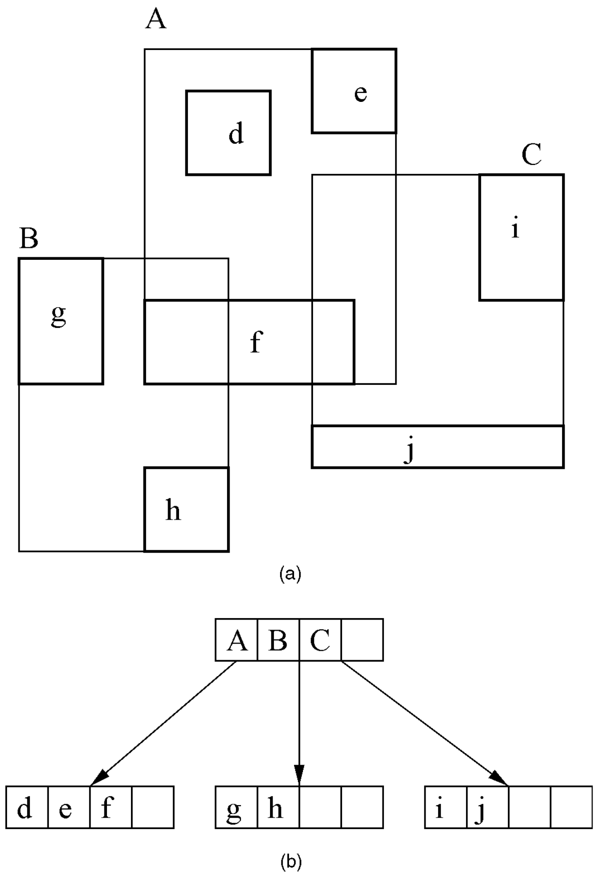


Fig. 6: (a) Spatial objects (bold) arranged in R-tree hierarchy; (b) R-tree file structure on disk.

2.5 Other Accomplishments

Spatial applications like NASA's Earth Observation System (EOS) have some of the largest data sets encountered in any application to date. This has prompted new research in database-file design for storage on tertiary storage devices such as juke-boxes. Representative results include those from the Sequoia 2000 project [30]. High-performance spatial applications such as flight simulators with geographic accuracy have triggered the development of new parallel formalizations for the range query and the spatial join query, including declustering methods and dynamic-load balancing techniques for multidimensional spatial data [28], [19]. Other interesting developments include hierarchical algorithms for shortest path computation [14] and view materialization [26].

3 Research Needs

Spatial databases are being used for an increasing number of new applications, such as Intelligent Transportation Systems, NASA's Earth Observation System, Multimedia Information Systems (MMIS) and Data Warehouses. This section lists representative research needs.

3.1 Space Taxonomy

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 [11] 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.

3.2 Spatial Data Model

Spatial data models have been developed for topological, metric and coordinatized Euclidean space. The OGIS specification alluded to in Section 2.2 is confined to topological operators [8], and more work is needed to incorporate relationships which involve directional [29] and metric properties (see Table 2 for examples). In addition, there has been very little work toward developing data models, data types (e.g., node, edge, path), and a kernel set of operations (e.g., get-successors, shortest path) for network space, despite their critical role in applications like transportation and utility management (telephone, gas, electric).

Similarly, there is a need for developing the field data model [33] toward a field-based query language. Operations on fields will be needed to help derive new information such as land-cover classification; the fields involved include temperature, texture, and water content, and are obtained through imaging in different bands such as infrared, visible bands, or microwave.

3.3 Spatial Query Processing

Many open research areas exist at the logical level of query processing, including query-cost modeling and strategies for nearest neighbor, bulk loading as well as 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-models is promising [4], [36], but more work is needed.

Similarly, common strategies employed in traditional databases for the *logical transformation* step in query optimization may not be always applicable in the context of spatial databases. For example consider the query (see Fig. 2a). Let us assume that the **Area()** function is not precomputed and that its value is computed afresh every time it is invoked. A query tree generated for the query is shown in Fig. 2b.

In the classical situation, the rule "select before join" would dictate that the **Area()** function be computed before the join predicate function, **Distance()** (Fig. 7a), the underlying assumption being that the computational cost of executing the select and join predicate are equivalent and

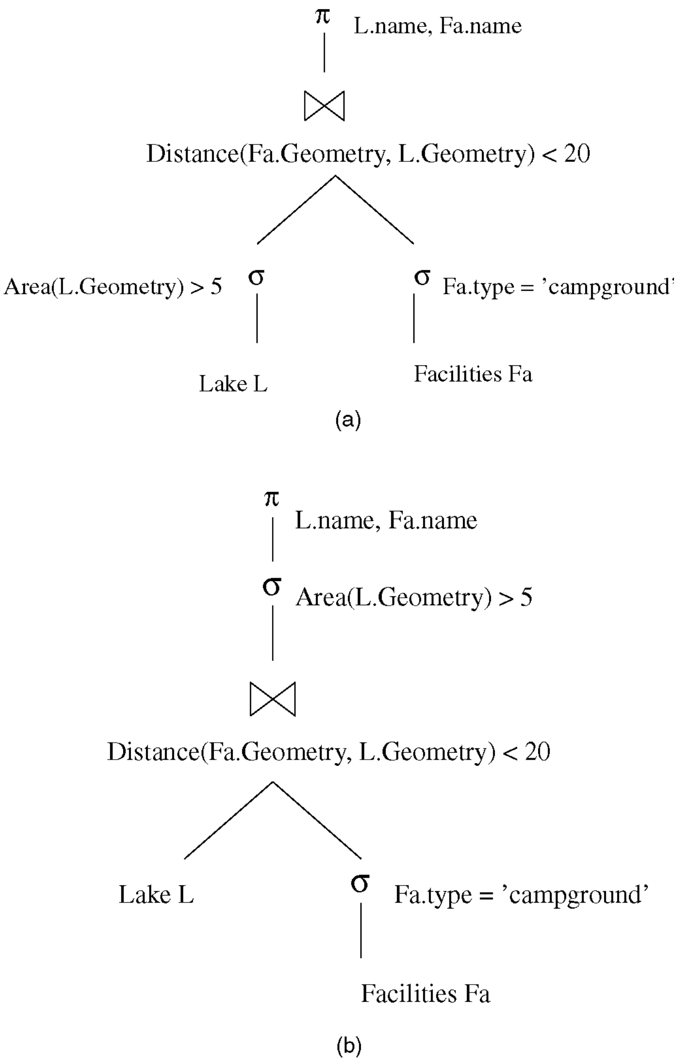


Fig. 7: (a) Area() before distance(); (b) Distance() before Area().

negligible compared to the I/O cost of the 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 [13]. Depending upon the implementation of these two functions the optimal strategy may be to process the join before the select operation(see Fig. 7b).

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 in Table 4. These include queries on objects using predicates other than overlap and queries on fields such as slope analysis as well as queries on networks such as the shortest path to a set of destinations. Bulk loading strategies for spatial data also need further study.

3.4 Spatial File Organization and Indices: Physical Level

Many file organizations and indices with distance metrics have been developed for coordinatized Euclidean space. However, little work has been done on file clustering and on indices for network spaces such as road maps and

TABLE 4
DIFFICULT SPATIAL QUERIES FROM GIS

Buffer	Find the areas 500 ft. from power lines
Voronoize	Classify households as to which supermarket they are closest to
Neighborhood	Determine slope based on elevation
Network	Find the shortest path from the warehouse to all delivery stops
Allocation	Where is the best place to build a new restaurant
Transformation	Triangulate a layer based on elevation
Bulk Load	Load a spatial data file into the database
Raster $\leftrightarrow$ Vector	Convert between raster and vector representations

telephone networks. Further work is needed, both to characterize the access patterns of the graph algorithms that underlie network operations and to design access methods.

The R-link tree [16] 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. The data volume of emerging spatial applications such as NASA's EOS is among the highest of any database application. Sequoia 2000 [30] provides an approach toward tertiary storage files and indices. Other approaches for managing databases on tertiary storage need to be investigated.

3.5 Other

Other research needs include benchmarking, workflow modeling, and the visual presentation of results. The Sequoia 2000 [30] benchmark characterizes the data and queries in Earth Science applications. The performance of loading data, raster queries, spatial selection, spatial joins, and recursion is addressed in 11 benchmark queries. A few more are provided in the Paradise system [9]. Similar benchmarks are needed to characterize the spatial data management needs of other applications such as GIS, DWH, and transportation.

The workflow in some spatial applications such as GIS is based on manipulating layers to produce new, derived layers. Typically, the layers are combined in a tree-based manner, starting with a large number of source layers and producing new layers until a final result layer is produced. Information about dependence among layers is useful for change propagation if the source layers are modified.

Spatial databases may require a different type of concurrency support than is needed by traditional databases. For example, transactions in traditional systems tend to be short (on the order of seconds). However, in spatial databases, these transactions can last up to a couple of hours for editing and browsing. Similarly, recovery and backup issues may also change, as the spatial objects tend be large (a few megabytes) when compared to their counterparts in traditional systems. There is a need to characterize the work flow of spatial applications.

Many spatial applications present results visually, in the form of maps which consist of graphic images, 3D displays, and animations. They also allow users to query the visual representation by pointing to the visual representation using devices like a mouse or a pen. Further work is needed to explore the impact of querying by pointing and visual presentation of results on database performance.

4 SUMMARY AND DISCUSSION

In this survey, we have presented the major research accomplishments and techniques which have emerged from the area of SDBMS. These include object-based data modeling, spatial data types, filter and refine techniques for query processing and spatial indexing. We have also identified areas where more research is needed. Some of these areas are spatial graphs, field based modeling, cost modeling and concurrency control, query processing techniques and discretization and propogation error.

Many of the spatial techniques highlighted in this survey are being used in an increasing number of applications such as GIS, CAD, and EOS. We believe that other emerging multidimensional applications such as multimedia information systems will use these methods to solve problems such as searching and indexing spatial content. We illustrate the possibilities in the context of multimedia information systems with text, audio and video data over the World Wide Web.

Multimedia data has a spatial content which can be queried using the same spatial operators that have become popular in geographic information systems. For example, the spatial operator *inside of* can be applied to text to locate sentences that contain the word "multimedia." Also, audio is often broken into channels with each channel containing input from a different source; for instance, trumpet, guitar, and voice. These channels are analogous to layers in GIS and can be manipulated similarly. A spatial join could determine all of the locations where the input from both piano and voice is over a certain decibel threshold.

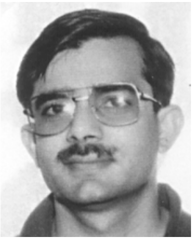
A video database such as a movie server can take advantage of techniques developed for spatial databases. Consider the movie *Toy Story*: Each frame contains spatial content with objects interacting in directional relationships. For instance, Buzz Lightyear could be above the trees when he is flying, and frames in the movie could be queried based on those relationships. For example, if you cannot remember when in the movie an important event occurred, but you can remember that Buzz Lightyear was in front of a tree, you would be able to query the movie using that relationship to determine when in the movie that event took place. Such queries exploit the directional relationships inherent between all tangible objects.

ACKNOWLEDGMENTS

This work is sponsored, in part, by the United States Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory Cooperative Agreement No. DAAH04-95-2-0003 and Contract No. DAAH04-95-C-0008, the contents of which do not necessarily reflect the position or the policy of the government; no official govenmental endorsement should be inferred. This work was also supported, in part, by the National Science Foundation under Grant No. 9631539. We thank Professor Jaideep Srivastava for technical commentary and Christiane McCarthy for helping to improve the readability of the paper.

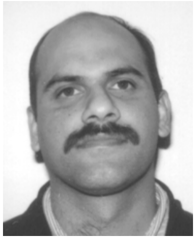
REFERENCES

[1] N. Adam and A. Gangopadhyay, *Database Issues in Geographical Information Systems*, Kluwer, 1997.
[2] 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*, vol. 181, no. 1, pp. 3–15, July 1996.
[3] P. Baumann, "Management of Multidimensional Discrete Data," *VLDB J.*, special issue on spatial database systems, vol. 3, no. 4, pp. 401–444, Oct. 1994.
[4] A. Belussi and C. Faloutsos, "Estimating the Selectivity of Spatial Queries Using the 'Correlation' Fractal Dimension," *Proc. 21st Int'l Conf. Very Large Data Bases*, pp. 299–310, Zurich, Sept. 1995.
[5] T. Brinkhoff, H.-P. Kriegel, and B. Seeger, "Efficient Processing of Spatial Joins Using R-Trees," *Proc. SIGMOD Conf. Management of Data*, pp. 237–246, Washington D.C., ACM, June 1993.
[6] D. Chamberlin, *Using The New DB2: IBM's Object Relational System*, Morgan Kaufmann, 1997.
[7] N. Chrisman, *Exploring Geographic Information Systems*, John Wiley and Sons, 1997.
[8] E. Clemintini and P. Di Felice, "Topological Invariants for Lines," *IEEE Trans. Knowledge and Data Eng.*, vol. 10, no. 1, pp. 38–54, 1998.
[9] D.J. DeWitt, N. Kabra, J. Luo, J.M. Patel, and J.-B. Yu, "Client-Server Paradise," *Proc. 20th Int'l Conf. Very Large Data Bases*, pp. 558–569, Santiago de Chile, Chile, Sept. 1994.
[10] M. Egenhofer, "Spatial SQL: A Query and Presentation Language," *IEEE Trans. Knowledge and Data Eng.*, vol. 6, no. 1, pp. 86–95, 1994.
[11] R.H. Güting, "An Introduction to Spatial Database Systems," *VLDB J.*, special issue on spatial database systems, vol. 3, no. 4, pp. 357–399, 1994.
[12] R. Guttman, "R-Tree: A Dynamic Index Structure for Spatial Searching," *Proc. SIGMOD Conf., Ann. Meeting*, pp. 47–57, Boston, ACM, 1984.
[13] J.M. Hellerstein and M. Stonebraker, "Predicate Migration: Optimizing Queries with Expensive Predicates," *Proc. SIGMOD Int'l Conf. Management of Data*, pp. 267–276, Washington, D.C., ACM, May 1993.
[14] N. Jing, Y. Huang, and E. Rundensteiner, "Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation," *IEEE Trans. Knowledge and Data Eng.*, vol. 10, no. 3, pp. 409–432, 1998.
[15] W. Kim, J. Garza, and A. Kesin, "Spatial Data Management in Database Systems," *Proc. Third Int'l Symp Advances in Spatial Databases*, pp. 1–13, Lecture notes in Computer Science 692, Springer-Verlag, Singapore, 1993.
[16] M. Kornacker and D. Banks, "High-Concurrency Locking in R-Trees," *Proc. 21st Int'l Conf. Very Large Data Bases*, pp. 134–145, Zurich, Sept. 1995.
[17] R. Laurini and D. Thompson, *Fundamentals of Spatial Information Systems*, Academic Press, 1992.
[18] J. Lee, Y. Lee, K. Whang, and I. Song, "A Physical Database Design Method for Multidimensional File Organization," *Information Sciences*, vol. 120, no. 1, pp. 31–65, Oct. 1997.
[19] D.-R. Liu and S. Shekhar, "A Similarity Graph-Based Approach to Declustering Problems and Its Application Toward Parallelizing Grid Files," *Proc. 11th Int'l Conf. Data Eng.*, pp. 373–381, Taipei, Taiwan, Mar. 1995.
[20] U.S. Army Corps of Engineers, Topographic Engineering Center, URL: http://www.tec.army.mil/gis-internet2.html.
[21] Open GIS Consortium, *OpenGIS Simple Features Specification for SQL*, URL: http://www.opengis.org/public/abstract.html, 1998.
[22] J.M. Patel and D.J. DeWitt, "Partition Based Spatial-Merge Join," *Proc. 1996 SIGMOD Int'l Conf. Management of Data*, pp. 259–270, Montreal, ACM, 1996.
[23] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990.
[24] S. Shekhar and S. Chawla, *Spatial Databases: Concepts, Implementation and Trends*, first draft, URL: http://www.cs.umn.edu/Research/shashi-group/ Book/index.html, 1998.
[25] S. Shekhar, M. Coyle, D.-R. Liu, B. Goyal, and S. Sarkar, "Data Models in Geographic Information Systems," *Comm. ACM*, vol. 40, no. 4, pp. 103–111, 1997.
[26] S. Shekhar, A. Fetterer, and B. Goyal, "Materialization Trade-Offs in Hierarchical Shortest Path Algorithms," *Proc. Fifth Int'l Symp. Advances in Spatial Databases*, Lecture Notes in Computer Science 1,262, Springer-Verlag, pp. 94–111, 1997.
[27] S. Shekhar and D.-R. Liu, "A Connectivity-Clustered Access Method for Networks and Network Computation," *IEEE Trans. Knowledge and Data Eng.*, vol. 9, no. 1, pp. 102–119, Jan. 1997.
[28] S. Shekhar, S. Ravada, V. Kumar, D. Chubb, and G. Turner, "Parallelizing A GIS on A Shared Address Space Architecture," *Computer*, vol. 29, no. 12, Dec. 1996.
[29] S. Shekhar and X. Liu, "Direction As A Spatial Object," *Proc. GIS Workshop*, Maryland, ACM, Nov. 1998; also available at URL: http://www.cs.umn.edu/Research/shashi-group/ paper_list.html.
[30] M. Stonebraker, J. Frew, and J. Dozier, "The Sequouia 2000 Storage Benchmark," *Proc. SIGMOD Conf. Management of Data*, pp. 2–11, Washington D.C., ACM, May 1993.
[31] M. Stonebraker and G. Kennitz, "Postgres Next-Generation Database Management System," *Comm. ACM*, vol. 34, no. 10, pp. 78–92, 1993.
[32] M. Stonebreaker and D. Moore, *Object Relational DBMSs: The Next Great Wave*, Morgan Kaufmann, 1997.
[33] C.D. Tomlin, *Geographic Information Systems and Cartographic Modeling*, Englewood Cliffs, N.J.: Prentice Hall, 1990.
[34] UCGIS Congressional breakfast, URL: http://urban.rutgers.edu/ucgis, 1998.
[35] M.F. Worboys, *Geographic Information Systems: A Computing Perspective*, Taylor and Francis, 1995.
[36] Y. Theodoridis, E. Stefanakis, and T. Sellis, "Cost Models for Join Queries in Spatial Databases," *Proc. 14th Int'l Conf. Data Eng.*, pp. 476–483, Orlando, Fla., Feb. 1998.



Shashi Shekhar received the BTech degree in computer science from the Indian Institute of Technology, Kanpur, India, in 1985; and the MS degree in business administration and the PhD degree in computer science from the University of California, Berkeley, in 1989. He is currently an associate professor in the Department of Computer Science and an active member of the the United States Army High Performance Computing Research Center as well as at the Center for Transportation Studies at the University of Minnesota, Minneapolis. His research interests include databases, geographic information systems (GIS), and intelligent transportation systems. He has published more than 100 research papers in refereed journals, conference and workshop proceedings, and edited books. He is a member of the *IEEE Transactions on Knowledge and Data Engineering* Editorial Board and the IEEE Computer Society Computer Science Engineering Practice Board. He is guest editor of a special issue of *Communications of the ACM* on GIS, and he was program co-chair of the ACM International Workshop on Advances in GIS (1996). His work in GIS includes databases for managing spatial networks (e.g., road maps), parallelization of GIS, routing algorithms for Advanced Traveler Information Systems, and archiving traffic measurements. His group has developed some of the most efficient indexing methods for large roadmaps and algorithms for path evaluation as well

as for computing shortest paths. His sponsors include the U.S. National Science Foundation, the Army Research Laboratories, Control Data Inc., the USDOT, the MN/DoT, and the ITS Institute. His general area of research is data and knowledge engineering. Currently, his work is focused on storage, management, and analysis of scientific and geographic data, information, and knowledge. The research is motivated by, and has been applied to, application areas including transportation (ITS), manufacturing, and finance. In data engineering and database systems, his group has designed the Connectivity-Clustered Access Method (CCAM), a new storage and access method for spatial networks which outperforms alternative schemes in carrying out network computations. He has also worked with semantic query optimization, and high-performance geographic databases. In knowledge engineering, he has worked on the problem of discovery in databases. He has worked with symbolic machine learning techniques as well as neural networks. He has designed one of the fastest scalable parallel formulation of backpropagation learning algorithms for neural networks and these parallel formulations compute in excess of 1 gigabyte of connections per second. He is a senior member of the IEEE, and a member of the IEEE Computer Society, the ACM, and the AAAI.



Sanjay Chawla received his PhD degree in mathematics at the University of Tennessee in 1995. He is now a postdoctoral fellow in the Department of Computer Science at the University of Minnesota. Before that, he was an industrial postdoctoral fellow at the Institute for Mathematics and Its Application (IMA), also at the University of Minnesota. His research interests include Geographic Information Systems (GIS), spatial databases, and optimal control theory.

Siva Ravada received a BTech degree in computer science from Andhra University, India; and a PhD degree in computer science from the University of Minnesota. He is a senior member of technical staff in the Spatial Products Division at the Oracle Corporation. His main research interests are the design and analysis of parallel algorithms for query processing in spatial databases. He is a member of the ACM, the IEEE, and the IEEE Computer Society.

Andrew Fetterer received his BS degree in computer science from the Univeristy of Minnesota in 1995 and his MS degree in computer science in 1997. He is currently a data warehouse architect at the Panttaja Consulting Group in San Francisco, having previously worked at the Oracle Corporation. His research focuses on materialization issues in routing algorithms for large spatial networks. He is a member of the IEEE.

Xuan Liu received her bachelor of engineering degree in computer software in 1987 and her master of science degree in computer science in 1990, both from Xiamen University, Xiamen, Peoples Republic of China. She is currently a PhD student in the Computer Science Department at the University of Minnesota. She was an assistant professor at Xiamen University from 1990 to 1995. Her current research interests include spatial databases, object-oriented databases, Geographical Information Systems, spatial data modeling, spatial query processing and optimization, and query languages. She is a student member of the IEEE.

Chang-tien Lu received his bachelor's degree in computer science and engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1991; and the MS degree in computer science from the Georgia Institute of Technology, Atlanta, in 1996. He has been a PhD student in the Department of Computer Science at the University of Minnesota, Twin Cities, since 1996. His research interests include geographic information systems, multidimensional data and indexes, spatial query, and spatial join processing. He is a student member of the IEEE.