SVIQUEL: A Spatial Visual Query and Exploration Language

Sudhir Kaushik and Elke A. Rundensteiner

Department of Computer Science
Worcester Polytechnic Institute, Worcester, MA 01609
sudhir@cs.wpi.edu, rundenst@cs.wpi.edu
http://davis.wpi.edu/dsrg/index.html

Abstract. *This paper represents the first effort to apply direct-manipulation (VIS) techniques to spatial data exploration providing mechanisms for visually querying as well as browsing spatial data and for reviewing the visual results for trend discovery. We present a visual query language (SVIQUEL) which allows us to specify the relative spatial position (both topology and direction) between objects using direct manipulation of spatial query filters, called S-sliders. The S-sliders for the selection of precise numeric ranges for query dimensions are complimented by Active-Picture-for-Querying (APIQ), a qualitative direct-manipulation query paradigm. SVIQUEL thus supports spatial browsing and spatial trend discovery with no particular query in mind. The tight coupling between the quantitative S-sliders and qualitative APIQ allows the users to work either qualitatively or quantitatively depending on his/her particular needs and freely switch between the two while working in a continuous data exploration mode.*

1 Introduction

The exploration of large information spaces remains a challenging task especially with the growth of the World Wide Web and other such huge repositories [1]. Many users are not able to cope with the flood of information. Visual Information Seeking (VIS) addresses this problem by recognizing the enormous capacity of human visual information processing [1]. By presenting information visually and allowing user interaction through direct-manipulation, users can traverse larger information spaces in shorter time. Our goal is to apply the concept of VIS to *special-purpose data types* such as *spatial* and *temporal* data, as to the best of our knowledge VIS has up to now only been applied to simple numeric data types [1] [18]. Our previous work focussed on designing a direct manipulation environment for *one-dimensional temporal data*, called TVQL [8] [7] [6] [9], that allowed a user to visually specify temporal queries over video data giving an instantaneous visualization of the results of the query. Formal user studies of TVQL show that users could more accurately specify and more quickly adjust queries using TVQL than using forms [8] [9], indicating the promise of our approach.

In this paper, we study the extension of the above concepts for *two-dimensional* spatial data. In particular, our contributions are:

- An integrated framework for spatial data analysis incorporating both direction and topology queries into one uniform exploration environment.
- A visual query interface for spatial data (SVIQUEL) over this integration of direction and topology expanding VIS to more complex data types (spatial in this case) which has not been attempted before.
- An active picture (APIQ) for query specification at the qualitative level and for visual query disambiguation of the quantitative S-sliders.
- Mapping functions between quantitative settings of S-sliders to qualitative query semantics in APIQ, providing for a tight coupling between the two[13].
- A working implementation of SVIQUEL implemented using the AWT library of JDK1.1.3 (http://davis.wpi.edu/dsrg/MMVIS/svql/svql.html).

Section 2 reviews related work. Section 3 gives an overview of our approach. The primitive spatial relations and the integrated neighborhood concept are introduced in Section 4 while APIQ is introduced in Section 5. Section 6 concludes the paper.

2 Related Work

Surprisingly little has been done thus far in applying the direct manipulation paradigm to spatial environments. Visual query environments based on VIS [18] are typically restricted to simple numeric domains. The Dynamic Homefinder, a Real Estate Information Exploration System based on VIS [18], allows users to search for a house that meets their criteria by manipulating sliders for different input parameters (such as number of rooms, price, distance from a fixed point, etc). However, it does not allow the user to specify the relative *spatial* position of two objects. Our goal is to provide a new class of dynamic filters tuned for continuous spatial querying.

Most spatial query languages are textual (SQL-based) [3]. The possibly most popular GIS system MapInfo [1] uses an SQL-style language to specify queries to query primarily over non-spatial and a few spatial attributes such as area and distance of the underlying data. This language however is neither incremental nor direct-manipulation based and thus our proposed interface is complimentary to such technology.

Cigales [17] takes a Query by Example approach that specifies the query using an actual drawing, while GEO-QUEL [2] uses an image depicting the relationship as means for query specification.

3 Overview of Our Spatial Exploration Environment

The four major components of our system include the spatial slider interface (S-sliders), the Active Picture for Querying interface (APIQ), the Spatial Query Processor and the Spatial Visualization of results (SVIZ) [12, 15, 14]. A user

[1] More information about MapInfo can be found at "http://www.mapinfo.com"

specifies a query using a combination of the S-sliders and APIQ interfaces. The S-slider interface is tightly coupled to APIQ allowing both qualitative and quantitative spatial data exploration. APIQ also presents a valuable aid in confirming the query specified in a visually intuitive manner. The quantitative queries expressed by the S-sliders are mapped to range queries to be processed by the query processor [6] [10]. Once processed against the spatial database, a graphical visualization of the results is presented to the user via SVIZ. The visualization is immediately updated whenever users manipulate one of the interfaces, giving the users a sense of causality between adjusting a query interface and the visualization of the results.

4 SVIQUEL Interface

The following are the goals in designing SVIQUEL:

- a visual interface to specify relative position spatial queries of any type such as topological or directional relationships in a precise yet simple manner via direct manipulation.
- ability to quickly adjust and incrementally refine spatial queries via direct manipulation.
- power to both specify particular queries as well as to browse over spatial relationships, with no particular query in mind.
- support for composing complex queries such as a combination of spatial relationships in a simple manner.

Given two spatial objects A and B, the spatial (topological, directional or distance) relationship between the two objects can be inferred from the relationships between the extremities of each object along both the X and Y dimension [2]. Fully constraining relative object placement gives complete information about how B is spatially placed with respect to A, such as whether B is to the north and overlaps with A. Our interface constrains the placement of one object B called the *primary* object with respect to the second object A that serves as the *reference* object [16, 11]. To query over the relative spatial position, we design our SVIQUEL language to specify any primitive spatial relationship as well as combinations of these primitive relationships as discussed below.

4.1 Relative Spatial Queries

Given two spatial objects A and B, our SVIQUEL interface distinguishes between eight possible primitive topological relationships *(disjoint, meet, overlap, covers, coveredBy, contains, inside and equal)* [4] and eight possible directional

[2] One important assumption here is that we are approximating each object by its Minimum Bounding Rectangle (MBR) with its sides oriented along the X and Y axes, and the spatial relationship between the two objects is expressed in terms of its MBR.

relationships *(north, south, east, west, north east, north west, south east and south west)* between the two objects and two special relationships of same-longitude and same-latitude. Since it is possible for two regions to have both kinds of relationships, each combination of a topology and a direction forms a *primitive* spatial relation (a SPRIM).

In our model, we have identified 45 meaningful primitives among the 64 possible ones (Fig. 1), which capture both the topological and the directional relationship between the regions. The reason we have 45 spatial primitives (SPRIMs) and not 64 is that for certain topologies like *inside*, *contains* or *equals* having a directional component is not meaningful [5].

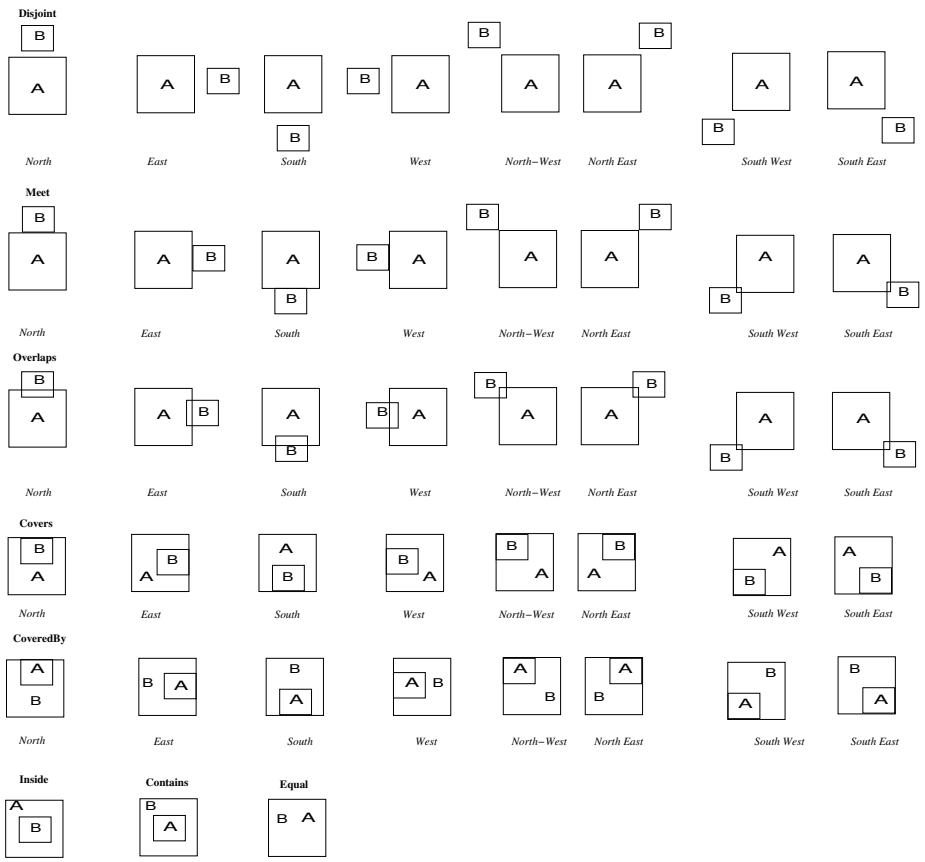

Fig. 1. Spatial Primitives (SPRIMs) with Topology and Direction Combined.

The SPRIMs as depicted in Fig. 1 can be defined in terms of the differences between the left and right ends in the X dimension and the top and bottom

ends in the Y dimension of the two objects A and B. The left and right ends of an object are its extremities in the X dimension while the top and bottom ends are its extremities in the Y dimension. Consider two objects A and B with the extremities for each object along the X dimension given by Axl, Axr, Bxl, Bxr [3]. The spatial relationship between the two objects is defined by a conjunction of the eight pairwise relationships between the extremities in each dimension. In many cases, fewer than eight differences are sufficient to represent the relative spatial position of two objects: ((Axl θ Bxl) AND (Axl θ Bxr) AND (Axr θ Bxl) AND (Axr θ Bxr)) AND ((Ayt θ Byt) AND (Ayt θ Byb) AND (Ayb θ Byt) AND (Ayb θ Byb)) where $\theta = \{ < , > , = \}$. [4] [5]

Each of these eight expressions combined by the AND operator shows the relative spatial position of the corresponding extremity of the two objects. For example, Axl $<$ Bxl indicates that the left end of A, denoted by Axl, has to be to the left ("$<$") of the left end of B, denoted by Bxl. In our SVIQUEL interface, dTl, dTr, dTlr and dTrl represent these differences in the X dimension and dTt, dTb, dTtb and dTbt represent these differences in the Y dimension.

Consider a sample scenario of two objects, a house and a lake, with the house the primary object B and the lake the reference object A. Suppose that the house is *disjoint* from the lake and to the *north*. This spatial relationship between the two objects could be expressed in terms of the eight differences: dTl = (positive or negative); dTlr = (negative); dTrl = (positive); dTr = (positive or negative); dTt =(negative); dTtb = (negative); dTbt =(negative); and dTb = (negative). The values dTt, dTbt, dTtb and dTb being negative implies that object B has to be above object A and so the topology is disjoint. Since dTlr is negative and dTrl is positive, it implies a portion of B has to be exactly above A. Thus we infer that B is to the north of A. So, the relative position of B with respect to A is *disjoint-north* (see top leftmost SPRIM in Fig. 1).

4.2 The SVIQUEL Sliders (S-sliders)

As outlined above, both the directional and topological content of a spatial relationship can be represented using these eight differences. In fact, different value combinations of these eight differences can uniquely specify all primitive spatial relationships (topological and directional) as shown in Fig. 1. Each of these differences assumes a continuous range of values and therefore dynamic query sliders can be used to specify each of these eight differences. Hence, we use eight dynamic query sliders (S-sliders) as part of our Spatial Visual Query and Exploration Language (SVIQUEL). These eight are split up as four along

[3] Axl and Axr are the left and right extremities of A in the X dimension respectively and the same holds true for B also and along the Y dimension given by Ayt, Ayb, Byt, Byb. Ayt and Ayb are the top and bottom extremities of A in the Y dimension respectively and the same holds true for B also.

[4] We assume that each object has a finite non-zero width and height and also Axl $<$ Axr, Ayt $<$ Ayb, Bxl $<$ Bxr, Byt $<$ Byb always.

[5] The sizes of A and B do not convey any information about the size of the actual objects that they represent.

each dimension as each of the two 2D objects has two endpoints along each dimension (see Fig. 2).

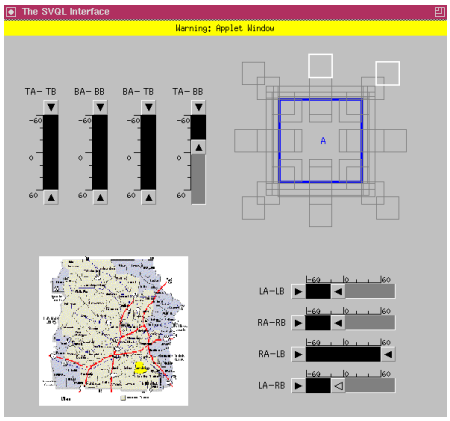

Fig. 2. APIQ for Query "Find all cases where B is disjoint from A and north or north east of A"

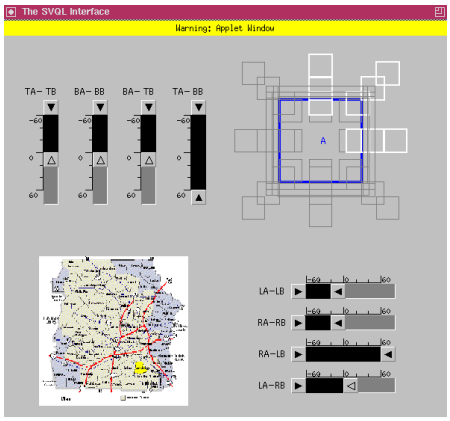

Fig. 3. APIQ for Query "Find all cases where B is disjoint from A and north, north east or east of A"

In order to prevent the specification of illegal queries, constraints and dependencies are introduced between the sliders so that only legal combinations of values can be specified using our interface [13]. There are two main advantages of creating such dependencies: it avoids the specification of illegal queries by users, and it simplifies the process of query specification whenever possible.

4.3 Integrated Neighborhood Concept for Topology and Direction

In addition to selecting one SPRIM among the 45 distinct possible spatial primitives, our SVIQUEL interface is designed to also allow the user to specify a combination of these primitives. Since our sliders enable us to specify a range of values for each of the sliders, we can also use them to specify combinations of spatial relations by adjusting the sliders to include a larger set of values. [6] Let us consider the scenario where we have a number of houses in the vicinity of a lake and the user is interested in buying a house in the region. Suppose he asks the query, "Give me those houses which are to the *north* or the *north east* of the lake". In our example above, the user essentially has two spatial relations specified between the house and the lake, one is the *north* directional relation and the other is the *north east* directional relation. Since the sliders specify a range

[6] For simplicity, we follow the common approach of modeling both the primary and the reference objects as two rectangles (the Minimum Bounding Rectangles of the two objects). The sliders can be used to manipulate the left, right, top and bottom extremities of the rectangle.

of values, the user could ask for these two relations in a single query by proper adjustment of the appropriate sliders. This way, the user is actually querying for combinations of spatial relations. If the user now adjusts the slider "Right A - Left B" to also include negative values, the configuration *disjoint-north east* is also included (see Fig. 2).

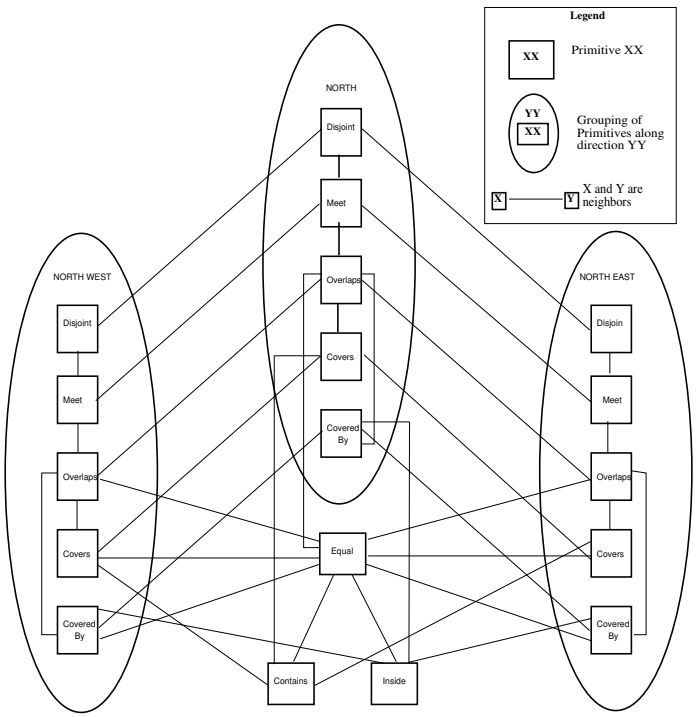

Fig. 4. Our Neighborhood Model Combining Topology and Direction Relations.

However, since the sliders operate over continuous ranges of values, they can only be used to specify those combinations of relations which are 'neighbors' of each other. The specification of continuous ranges of values by the sliders enables us to go from one spatial relation to the one closest to it by the simple move of the mouse and thus fosters rapid yet continuous exploration of the data set. Going to any arbitrary relation from the current one would require the skipping of values in the middle which is not supported by our query paradigm.

This observation requires us to establish a precise definition for this intuitive concept of neighborhood viz the relations closest to a given relation form a neighborhood. For topology, we borrow the concept of neighborhoods among topological spatial relations between a line and a region [4]. This model considers the intersections between the interior, boundary and the exterior of the line and

the region, and differentiates between the topologies based on these 9 intersection values. For direction, we use the concept of neighborhood as developed by Hernandez [5].

Given our goal of querying over both topology and direction, we developed a neighborhood model for an aggregate relationship integrating both direction and topology [5]. Our neighborhood model is a natural extension of the two individual models. In the sense, when we ignore one of the two spatial components (direction or topology), our model defaults to the individual known model. The neighbors for a particular SPRIM can be obtained in our model using the knowledge of its neighbors with respect to topology and with respect to direction. So, the neighbors for a particular spatial primitive can be deduced as follows:

- Those relations which have the same topology but are neighbors with respect to direction.
- Those relations which have the same direction but are neighbors with respect to topology.

The underlying assumption here is that the user is allowed to go from one spatial primitive to the other only by moving along the two axes of our 2D space, one at a time, i.e., he is not allowed to move diagonally along the two axes. [7] The chart showing the neighborhood relations of the SPRIMs in the *north*, *north east* and *north west* directions are shown in Fig. 4. The other neighborhood relationships can also be derived in a similar manner and are omitted here for space reasons. The configurations *same-longitude* and *same-latitude* are special cases. This formal model of neighborhood enables us to specify all legal combinations of relative spatial positions that could be specified by SVIQUEL as well as incremental transitions between them. Thus, each SVIQUEL query corresponds to a set of neighboring SPRIMs and the interface allows to move from one SPRIM to another if and only if they form a neighborhood set.

5 APIQ: An Active Picture for Qualitative Querying

5.1 Direct Manipulation in APIQ

The APIQ environment is used to specify a qualitative relative position spatial query. In our system, the user selects and de-selects the primitives which are a part of the query using mouse clicks. For example, given the APIQ depicted in Fig. 2, if the user wishes to specify the primitive *disjoint east* as part of the query, then a mouse click on that primitive results in including that primitive as part of the query. The resulting APIQ and S-sliders, consistent with the integrated neighborhood model, are shown in Fig. 3. The tight coupling between the S-sliders and APIQ achieved by our mapping functions between the two gives the new setting of the S-sliders as a result of the direct manipulation of APIQ [13].

[7] The configuration *same-longitude* is the neighbor of *overlap-north* and *overlap-south* and also to all neighbors of these two configurations. In the case of *same-latitude*, its neighbors are *overlap-east*, *overlap-west* and all the neighbors of these configurations.

After the user has specified the qualitative query using the APIQ, the S-sliders could be used to set precise numeric values and thus specify the relative position query quantitatively.

5.2 APIQ: Qualitative Query Representation

Formal user studies of temporal queries in MMVIS have shown that a graphical depiction of a slider-based query for query disambiguation results in better performance and easier comprehension for the users [8]. Borrowing this idea of qualitative query disambiguation of MMVIS [6], the Active Picture for Querying (APIQ) also provides visual *qualitative* representations of the query to the user. APIQ represents a valuable aid for confirming the specified query to the users in a visually intuitive manner.

APIQ is the natural compliment to the S-slider query model as it corresponds to the union of all neighboring SPRIMs specified by the user in the SVIQUEL query. In other words, APIQ shows all the primitive positions B could occupy with respect to a reference object A - each position graphically depicted by one SPRIM primitive.

This should increase the speed and the certainty with which the users modify and construct complex queries. Another advantage of APIQ is that it increases the utility of the SVIQUEL interface for spatial browsing of the data with no particular query in mind. That is, both APIQ and the visualization of the results (SVIZ) are dynamically updated as users manipulate the sliders, thereby enabling the users to simply slide the sliders front and back until an interesting result appears and then see from APIQ what query was specified.

6 Conclusions

In this paper, we have presented a framework for applying the direct manipulation paradigm to spatial data. In our work, we have defined a Spatial Visual Query and Exploration Language (SVIQUEL) for specifying relative spatial position queries. SVIQUEL enables users to specify any primitive spatial relation, with both direction and topology combined, as well as conceptual neighborhood combinations of them. Our SVIQUEL interface provides users with a visual paradigm for browsing the spatial data through direct manipulation and in a spatially continuous manner. SVIQUEL introduces a qualitative active picture for querying (APIQ) which allows the user to qualitatively specify spatial queries. We have a working implementation of the complete system in Java using JDK1.1.3 at the following URL ("http://davis.wpi.edu/dsrg/MMVIS/svql/svql.html").

Acknowledgments: Special thanks goes to Stacie Hibino, who had developed the initial MMVIS framework including TVQL, for providing us with TVQL source code on which we based our implementation. We would also like to thank the Database Systems Research Group (DSRG) at WPI whose feedback was very valuable to us at every stage of our research.

References

1. C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. *CHI'94*, pages 313–317, Apr 1994.
2. R. Berman and M. Stonebraker. Geo-Quel, a system for the manipulation and display of geometric data. *ACM Computer Graphics*, 11(2):186–191, 1977.
3. M. Egenhofer. Spatial SQL: A query and presentation language. *IEEE Trans. on Knowledge and Data Eng.*, 6:86–95, Feb 1994.
4. M. Egenhofer and D. Mark. Modeling conceptual neighborhoods of topological line region relations. *International Journal of Geographic Information Systems*, 9:555–565, 1995.
5. D. Hernandez. Qualitative representation of spatial knowledge. In *Lecture Notes in Artificial Intelligence*, volume 804. Springer Verlag, Feb 1994.
6. S. Hibino and E. Rundensteiner. MMVIS: Design and implementation of a multimedia VIS environment. *ACM Multimedia*, pages 75–86, 1996.
7. S. Hibino and E. A. Rundensteiner. A visual multimedia query language for temporal analysis of video data. In *MultiMedia Database Systems*, pages 123–159. Kluwer Aca, 1996.
8. S. Hibino and E. A. Rundensteiner. User study evaluation of direct-manipulation temporal interfaces. *ACM Multimedia*, pages 99–109, Nov 1997.
9. S. Hibino and E. A. Rundensteiner. Comparing MMVIS to a timeline for temporal trend analysis of video data. *Advanced Visual Interfaces*, May 1998.
10. S. Hibino and E. A. Rundensteiner. Processing incremental multidim. range queries in a direct manipulation paradigms. *IEEE Int. Conf. on Data Eng.*, pages 458–465, Feb 1998.
11. Z. John, M. Tamer, and D. Szarfon. Spatial reasoning rules in multimedia management systems. Multimedia Technical Report TR96-05, Department of Computer Science, University of Alberta, Mar 1996.
12. S. Kaushik. Direct-manipulation spatial exploration using SVIQUEL. Master's thesis, Computer Science Department, Worcester Polytechnic Institute, May 1998.
13. S. Kaushik and E. A. Rundensteiner. SVIQUEL: A spatial visual query and exploration language. Technical Report CS-TR-97-10, Comp. Science Dept, Worcester Polytechnic Institute, 1997.
14. S. Kaushik and E. A. Rundensteiner. Direct-manipulation spatial exploration using SVIQUEL. In *Visual Database Systems Conference'98*. Short paper and software system demonstration, Acquia, Italy, Springer Verlag, May 1998.
15. S. Kaushik and E. A. Rundensteiner. SEE: A spatial exploration environment based on a direct-manipulation paradigm. Technical Report CS-TR-98-7, Computer Science Department, Worcester Polytechnic Institute, May 1998.
16. D. Papadias, Y. Theodoridis, T. Sellis, and M. Egenhofer. Topological relations in the world of minimum bounding rectangles: A study with R-trees. *Proceedings of SIGMOD*, 24, June 1995.
17. A. B. Wansek, D. Calcinelli, B. Languou, C. Lecocq, and M. Mainguenaud. CIGALES: A visual query language for GIS: The user interface. *International Journal of Visual Languages and Computing*, 5:113–132, 1994.
18. C. Williamson and B. Shneiderman. The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system. Technical Report CS-TR2819, Dept. of Computer Science,Univ. of Maryland, Jan 1992.