Journal of Visual Languages and Computing, Vol. 8, No. 4, pp. 403-424, 1997.

Query Processing in Spatial-Query-by-Sketch*

Max J. Egenhofer
Department of Spatial Information Science and Engineering
Department of Computer Science
and
National Center for Geographic Information and Analysis
University of Maine
Orono, ME 04469-5711, USA
max@spatial.maine.edu
http://www.spatial.maine.edu/~max/max.html

Abstract

Spatial-Query-by-Sketch is the design of a query language for geographic information systems. It allows a user to formulate a spatial query by drawing the desired configuration with a pen on a touch-sensitive computer screen and translates this sketch into a symbolic representation that can the processed against a geographic database. Since the configurations queried usually do not match exactly the sketch, it is necessary to relax the spatial constraints drawn. This paper describes the representation of a sketch and outlines the design of the constraint relaxation methods used during query processing.

1. Introduction

Traditional methods for spatial querying are tedious [17]. The difficulties of communicating a user's request to a spatial database through conventional spatial query languages becomes most apparent when several users have to work together. Fundamental to this problem is the fact that verbal descriptions of spatial situations are frequently ambiguous and may easily lead to misinterpretations, particularly in multi-language groups. The use of traditional spatial query languages has serious limitations, because geographic concepts are often vague, imprecise, little understood, or not standardized. As an example, take the notion of the spatial predicate *cross* whose semantics may vary depending on the context in which it is used, the meaning of the objects to which the predicate relates, and the topology and the metric of the particular configuration [39]. These drawbacks make most current spatial query languages error-prone and difficult to use. Graphical user interfaces provide only little improvement for such query languages, because they use the same type of syntax and grammar as the typed languages, and their primary advantage is that they release users from remembering the particular syntax.

We attempt to overcome the limitations of conventional spatial query languages by considering alternative interaction methods between users and geographic data in a geographic information system. With the advent of pen-based user interfaces, a more intuitive style of interaction with spatial data is made possible than typing a query or composing it from menus. Pen-based user interfaces are expected to become more important in the future with an increasing demand for multi-media systems in most any application domain [31]. Particularly for the interaction with geographic data, pen-based user interfaces provide a series of advantages over current interaction

---

* This work was partially supported by the National Science Foundation under grant numbers SBR-8810917 (for the National Center for Geographic Information and Analysis), IRI-9309230, IRI-9613646, and SBR-9600465; by Rome Laboratory under grant number F30602-95-1-0042; and by a Massive Digital Data Systems contract sponsored by the Advanced Research and Development Committee of the Community Management Staff and administered by the Office of Research and Development.

techniques. By their very nature, geographic data are spatial and it is most appealing to refer to them in terms of explicit spatial concepts. Rather than expressing a spatial query in lexical terms, users may prefer to *sketch* a spatial query. Sketching is an interaction mode that more directly supports human spatial thinking than interactions through a verbal spatial query language alone, because users frequently have an image-like representation in their minds when they query about spatial configurations. It also provides immediate graphical feedback and, therefore, is an inherently more natural process to formulate many spatial constraints than a textual language. *In lieu* of forcing users to express a spatial configuration in some (semi)-formal or natural language, it is a major step towards the successful use of spatial information systems if users are allowed to draw a picture of the image they have in their minds, in order to retrieve the spatial data of interest. Such a spatial query language is *Spatial-Query-by-Sketch* [10], which will allow users to express spatial queries closer to the way they think about many spatial problems and incorporates powerful reasoning mechanisms to infer geometric variations in the sketch. Spatial-Query-by-Sketch is a design and a prototype implementation is currently under development. An area of particular interest is the access to digital image libraries [22, 42] through a language like Spatial-Query-by-Sketch, where users may want to retrieve, for instance, remotely-sensed images on which features match a particular geometric configuration drawn.

Besides many interesting considerations about interaction by sketching, Spatial-Query-by-Sketch poses challenging questions with respect to the processing of sketched queries. If the database were to retrieve only those configurations that provide an exact match with the geometry of the drawing, standard methods used in image matching and image retrieval could be applied. In a geographic context, however, it may be necessary to relax some of the constraints of the sketch, because trying to retrieve a situation that fits exactly the geometry of the sketch would only rarely result in a match. There is an important conceptual difference, however, between finding a picture that matches a sketch vs. finding a geographic configuration that matches a sketched query. In pictorial queries, the shape of the objects, their relative sizes, and their proportions are considered to be known [26]. The match between the picture and the sketch—the outline of some features that must appear on the image of interest—could be established through modest variations of the metric. The processing task is then to match the outlines with the boundaries on the pictures. Deviations between the image and the sketch occur due to inevitable inaccuracies in the user's drawing. To compensate for them, methods like epsilon bands around the boundaries, within which valid matches would be found, are acceptable solutions. In queries about geographic data, however, this is not the case, because such spatial properties as the orientation of the objects may be immaterial for the query or relative distances among the objects may be highly distorted.

To decide which constraints might be relaxed and which constraints should be maintained, it is necessary to base the query processing on a computational model for similarity of spatial relations. For this goal, we use a powerful computational model to represent spatial relations and extend this model where necessary to account for various degrees of similarity. This approach enables us to retrieve not only those situations that provide a perfect match with the sketch, but also those that capture the essence of the sketch; therefore, Spatial-Query-by-Sketch enables *spatial similarity retrieval* [26]. Experiments in psychology and cartography showed that topology is among the most critical information people refer to when they assess spatial relationships in geographic space [38, 49, 52], while metrical changes are frequently considered to be of lesser importance. To reflect such human behavior, Spatial-Query-by-Sketch is based on the premise *topology matters, metrical refines* [19].

The remainder of this paper first reviews previous approaches to spatial querying, focusing on traditional spatial query languages, visual spatial query languages, and sketching. Section 3 introduces the principles of Spatial-Query-by-Sketch and gives a guided tour through some fundamental interactions in Spatial-Query-by-Sketch. Section 4 focuses on the internal representation of a sketched query in the form of a semantic network with spatial objects and their spatial relationships. Query processing of spatial relations, relaxation of spatial constraints, prioritization of query results are described in Sections 5, 6, and 7, respectively. Conclusions and future work are discussed in section 8.

2. Spatial Querying

Spatial-Query-by-Sketch builds on state-of-the-art knowledge in spatial query languages, particularly visual spatial query languages, and extends the sketching paradigm. This section reviews relevant approaches in these fields.

2.1. Spatial Query Languages

Query languages for geographic databases and geographic information systems are either complex macro languages or extensions of SQL. There exists a large variety of Spatial SQL dialects [7, 28, 32, 47]. Such SQL extensions are relevant to Spatial-Query-by-Sketch, because they provide the means for accessing geographic databases and retrieving data from a database. Most critical is the support for spatial relations. Many SQL dialects include some notions of spatial relations, however, the semantics of the operations provide varying levels of detail and differ quite dramatically. Spatial extensions to SQL are currently being addressed by the SQL3 Multimedia working group.

Similar to SQL extensions, there are several spatial query languages that are derivatives of Query-by-Example. Query-by-Pictorial-Example [5] and Picquery [33] are examples for the Query-by-Example approach of inserting example values in tables, without exploiting the 2-dimensional characteristics of the language for spatial (2-dimensional) querying.

2.2. Visual Spatial Query Languages

More advanced user interfaces and spatial query languages include concepts similar to Spatial-Query-by-Sketch. The query language Cigales, for example, allows users to draw a query [4]. Unlike Spatial-Query-by-Sketch, Cigales requires the users, prior to drawing the sketch, to select the type of spatial relation they are addressing [37]. For instance, to specify that the road enters the park, the user would have to select the “intersect” operation, and then draw the particular configuration [1]. This leads to moded interfaces, which are tedious to use.

In a similar attempt, Lee and Chin [36] designed an iconic query language in which users compose a query by selecting spatial relations from a predefined set represented as icons. They only consider a small subset of topological relations, so that a user can select them from a set of icons.

A visual spatial query language that is based on a comprehensive algebra is Query-by-Visual-Example [43], an extension of Query-by-Example. Users of Query-by-Visual-Example construct templates of scenes in an array-like framework, describing primarily cardinal directions. While this approach comes closer to the way people think about space and its objects, it has its limitations through the equal resolution of the space. The grid also favors the specification of direction relations, but makes it more difficult to state approximate distances and topological relations independent of directions.

All of these visual spatial query languages lack a method to cope with the fact that an acceptable answer—even the best fit—may actually differ from the geometry in the query configuration.

2.3. Sketching

Sketching was used in the past primarily in CAD for design. Sketchpad [51] and ThingLab [2] were initial approaches to formulate constraints graphically. Pizano *et al.* [46] used spatial constraints for describing consistency in spatial databases; however, unlike describing situations that should match the configuration of interest, they focused on constructing those situations that would establish unacceptable database states. Although their language was iconic rather than sketch-based, it shares much similarity with the principles of sketching.

Sketching for querying was used in Query by Visual Example [29, 30, 34, 35] and Query by Image Content [22], which are targeted for content-based image retrieval. While the interaction mode of these query languages is similar to the basics of Spatial-Query-by-Sketch [10]—in both cases users draw an approximate spatial configuration of what to retrieve—scope and sketch interpretation are considerably different. Sketches for content-based image retrieval assume that the user draws something that matches quite closely the target and that all relations are intended as drawn. Their query processors accommodate primarily metrical variations and they are very sensitive to variations in sizes, orientations, and shapes. On the other hand, Spatial-Query-by-Sketch assumes that the user's sketch and the targets may vary considerably, as long as they match in the most important criteria.

Spatial relations have been considered as a secondary criterion in an image retrieval system that focuses on shape similarity [6]. The measures for shape are quantitative and thus expensive to process in a spatial database, and the spatial relations considered use rough approximations based on minimum-bounding rectangles. In contrast, Spatial-Query-by-Sketch prefers qualitative measures, starting with the spatial relations among the objects drawn, and resorts to quantitative methods only to prioritize hits.

The concepts of Spatial-Query-by-Sketch come closest to the Electronic Cocktail Napkin [25], which uses free-hand drawings to interact with architectural images, and a query language for sketch-based querying of geographic databases [40]. While their interactions modes and intention for similarity retrieval closely match with Spatial-Query-by-Sketch, their models used for representing sketches and processing them use an *ad-hoc* collection of spatial relations, which distinguishes Spatial-Query-by-Sketch as it is founded on a solid mathematical model of spatial relations and their relaxations.

3. Spatial-Query-by-Sketch

Spatial-Query-by-Sketch is designed to use a touch-sensitive input device—ideally a touch screen with a pen, such as Apple's Newton. Simulations may be obtained with a mouse or a trackball, but sketching with these devices is more cumbersome and therefore less effective. Users draw with a pen a geometric configuration that matches closely the spatial situation(s) they expect to retrieve from the geographic database. While composing the sketch, they may annotate the sketch to describe desired properties of the sketched objects. Spatial-Query-by-Sketch parses the sketch and translates it into a topological vector data model [27]. Subsequently, Spatial-Query-by-Sketch develops a query processing plan and executes the query against the spatial database. If several scenes match the query, the results are prioritized such that scenes with the best match to the query are presented first.

The following scenario provides a cursory outline of the envisioned interaction a user may perform when sketching a query. This user interface is organized into three major interaction areas: the sketch region in which the user draws the configuration of interest; the overview area which displays the sketch in its entirety and allows users to pan and zoom; and the control panel from which the user selects database commands, the type of feature he or she is drawing, and the confidence level for the placement of a feature. Users employ a pen to sketch an example of what they want to find in the database. In this particular case, the user is interested in all land parcels that have a wooded area and a river crossing the parcel. The user first sketches the parcel by selecting the class of the object (in this case a **Parcel**), and drawing its boundary (Figure 1a). Then she describes the location of the forest by drawing part of the forest's boundary (Figure 1b). Since it is unclear on which side of the line the forest is located, the user fills the interior of the forest (Figure 1c). Finally the user draws a river such that it crosses the land parcel (Figure 1d). Since the user is satisfied with the drawing, she requests that all configurations that match the sketch be retrieved from the database by pressing the **Go!** button on the control panel.

Journal of Visual Languages and Computing, Vol. 8, No. 4, pp. 403-424, 1997.

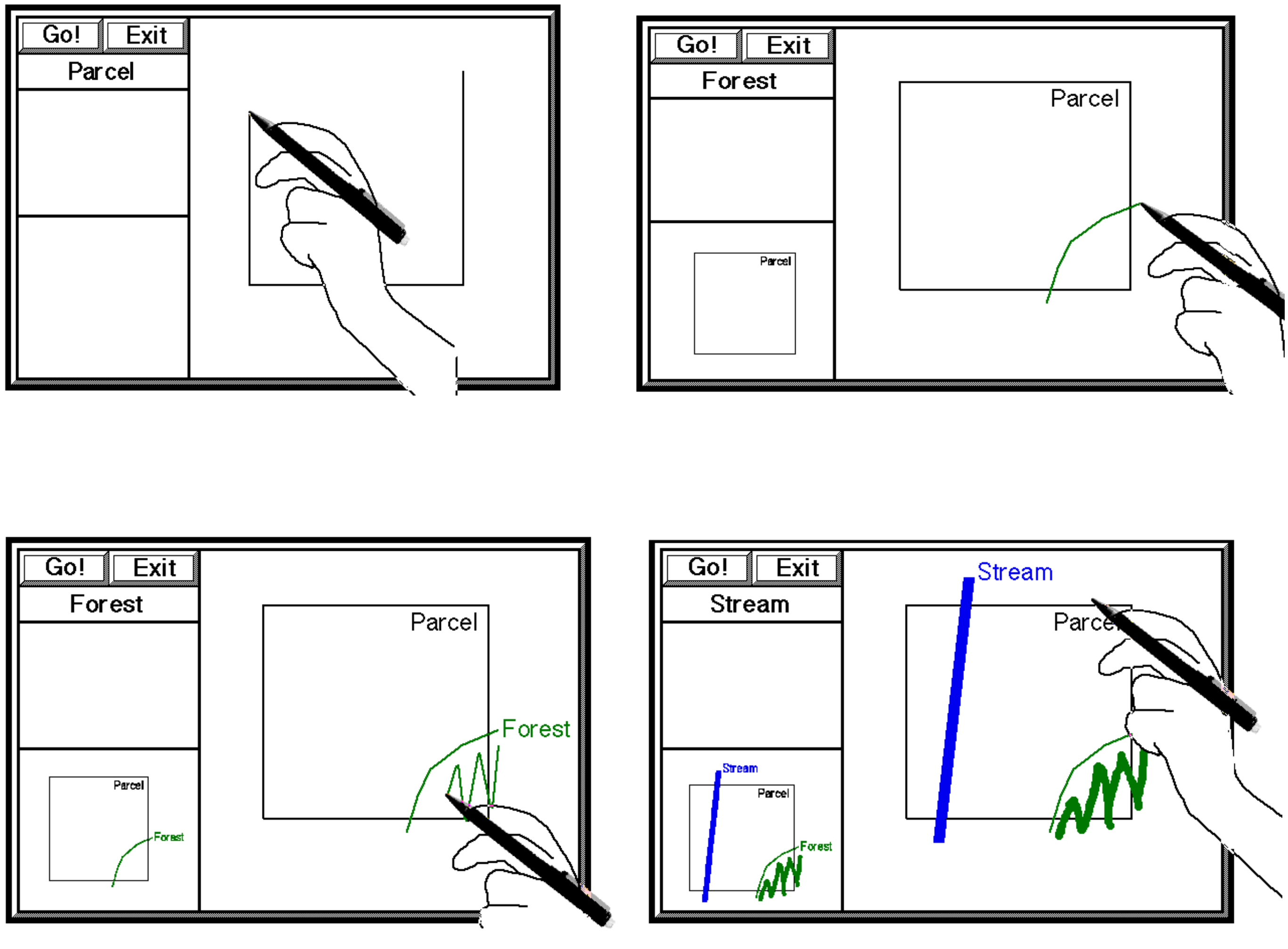


Figure 1: (a) The user draws the geometry of a land parcel; (b) the user adds the boundary of a forest; (c) to determine the location of the forest, the user fills the forest's interior; and (d) the user adds the location of a stream such that it crosses the land parcel, but does not intersect with the forest.

4. Symbolic Representation of a Sketch

While a bitmap representation would provide an accurate snapshot of such a sketch, it would be difficult to interpret it and match it against elements in other datasets whose relations, sizes, and shapes are distorted or not to scale with the sketch or whose orientations among elements differ to some degree. Instead, we select an *object representation* for the sketch, which allows us to abstract away some details of the sketch while it emphasizes its salient parts. This representation stresses objects, their spatial and non-spatial properties, and the spatial relations among the objects drawn. The latter are of particular importance for processing a query in Spatial-Query-by-Sketch as they capture the essence of a spatial scene.

We represent the sketch internally as a semantic network of spatial objects and their binary spatial relations. In this network, each object drawn corresponds to a node whose values are given by the semantics assigned in the sketch. They may include the class of an object, a name, other attribute values, or such metrical constraints as the size of the area or length of an object. Directed edges between nodes stand for binary spatial relations between the spatial objects. For this purpose, we distinguish five different types of spatial relations: coarse binary topological relations, detailed binary topological relations, metrical refinements, coarse cardinal directions, and detailed cardinal directions. With these five types of binary spatial relations, a qualitative model of a sketch is built in the form of a multi-resolution semantic network, called a *scene network*. Such a network serves as a symbolic, qualitative representation of the sketch. Its elements translate into predicates in spatial queries. See [44] for a discussion about the completeness of the approach of using binary relations for spatial queries. The scene network may be constructed at different levels of detail, for instance only at a coarse level of detail with topological and direction relations, or only as a topological representation with coarse and detailed topological relations. For the most detailed analysis, a complete scene network would be derived with all five types of spatial relations. Such a representation translates easily into database queries in the form of first-order predicates or extended-SQL statements. Depending on the configuration, fewer binary relations may be sufficient to describe the scene completely if they allow to drive uniquely the eliminated relations through compositions of elementary or inferred relations [20]. There are additional dependencies among the different types of binary relations that could further reduce the smallest number of relations required to fully specify a scene. For example, detailed cardinal directions imply their corresponding coarse cardinal directions. The actual number of spatial relations to be considered for processing a particular query is an issue of spatial query optimization [8]. In the following, we discuss the models used for the five types of spatial relations.

4.1 Coarse Topological Relations

We base the analysis of topological relations on the 9-intersection, a comprehensive model for binary topological relations that applies to objects of type area, line, and point [13, 15]. It characterizes the topological relation between two point sets, A and B, by the set intersections of A's interior, boundary, and exterior with the interior, boundary, and exterior of B, called the *9-intersection*. With each of these nine intersections being empty or non-empty, the model has 512 possible topological relations between two point sets, some of which cannot be realized. For two simple regions without holes embedded in R^2, the categorization shows eight distinct topological relations. They have been called *disjoint*, *meet*, *equal*, *overlap*, *inside*, *contains*, *covers*, and *coveredBy* (Figure 2). For two simple lines (non-branching, no self-intersections) embedded in R^2, 33 different topological relations can be realized with the 9-intersection, and for a line and a region, 19 different situations are found [16].

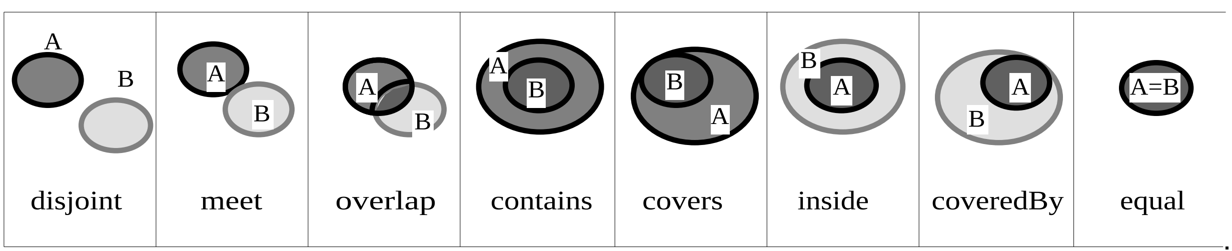


Figure 2: The eight topological relations that can be realized between two spatial regions embedded in R^2.

4.2 Detailed Topological Relations

More detailed distinctions about topological relations are possible if further criteria are employed to evaluate the non-empty intersections. In order to establish topological-relation equivalence between two regions (i.e., to decide whether or not two pairs of objects have the same topological relations), it is sufficient to describe topological invariants for the components (or separations) of the boundary-boundary intersection [14] and the approach generalizes to line-line and line-region relations. The necessary invariants to consider for region-region relations are:

Journal of Visual Languages and Computing, Vol. 8, No. 4, pp. 403-424, 1997.

- the *sequence* of components counted in a consistent orientation of the plane along the boundaries of the regions (Figure 3a);
- the *dimension* of each component—0-dimensional for boundary-boundary intersections in a single point and 1-dimensional for boundary-boundary intersections that form a common line (Figure 3b);
- the *type* of boundary-boundary component intersection—*touching* if the boundary enters and leaves the component intersection from the same part, or *crossing* if the boundary enters from a different part than it leaves (Figure 3c);
- the *crossing direction* of boundary-boundary components—*into* and *out of* the interior (Figure 3d);
- the *boundedness*, i.e., whether a 1-dimensional boundary-boundary component is inside or along the border of the union of the two objects (Figure 3e); and
- the *complement relationship*, i.e., whether a component is a next to an open or closed exterior (Figure 3f).

Journal of Visual Languages and Computing, Vol. 8, No. 4, pp. 403-424, 1997.

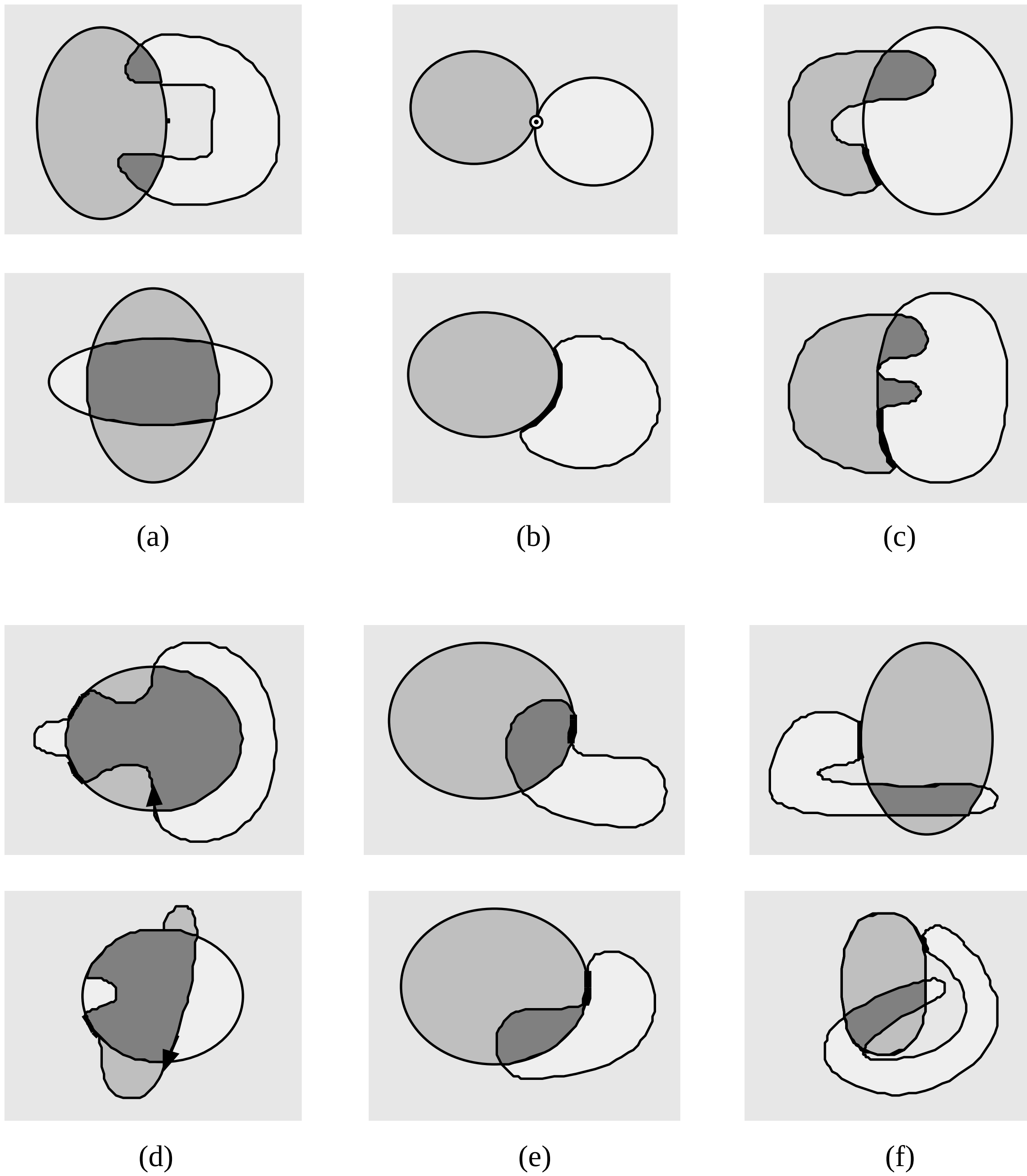


Figure 3: Six pairs of relations each of which distinguishes by different detailed topological relations: (a) component sequences, (b) component dimensions, (c) types of boundary-boundary component intersections, (d) crossing directions, (e) boundedness, and (f) complement relationships.

Detailed topological relations between two regions are expressed by the *component invariant table* for non-empty boundary-boundary sequences, which lists the sequence of boundary-boundary components and each component's dimension, type, crossing direction, boundedness, and complement relationship [12, 14].

4.3 Metrical Refinements

Occasionally, topology *per se* is insufficient to characterize the essence of spatial relations. For instance, in order to capture the semantics of the spatial relation between Interstate I-95 and the state of New Hampshire requires the consideration of some metrical properties in addition to topological concerns—I-95 divides New Hampshire into a very small area on one side of I-95 and a larger piece on the other side. To describe metrical details, we apply measures about areas and lengths as refinements of the topological properties [48]. These measures are normalized values with respect to the areas or lengths of interiors and boundaries and, therefore, scale-independent. These measures are defined as refinements of the 9-intersection, adding length and area measures to quantify non-empty intersections. The same concepts apply to the 9-intersections of line-region and line-line relations, although the number of measures that are applicable may vary with the geometric types of the objects involved. Figure 4 shows the eight measures that apply to region-region relations—six splitting ratios that capture how object *A*'s parts separate object *B*, and two closeness measures that describe relative distances from *A*'s boundary to *B*'s boundary. The same types of measures apply as ratios of *A*'s metrical properties over *B*'s.

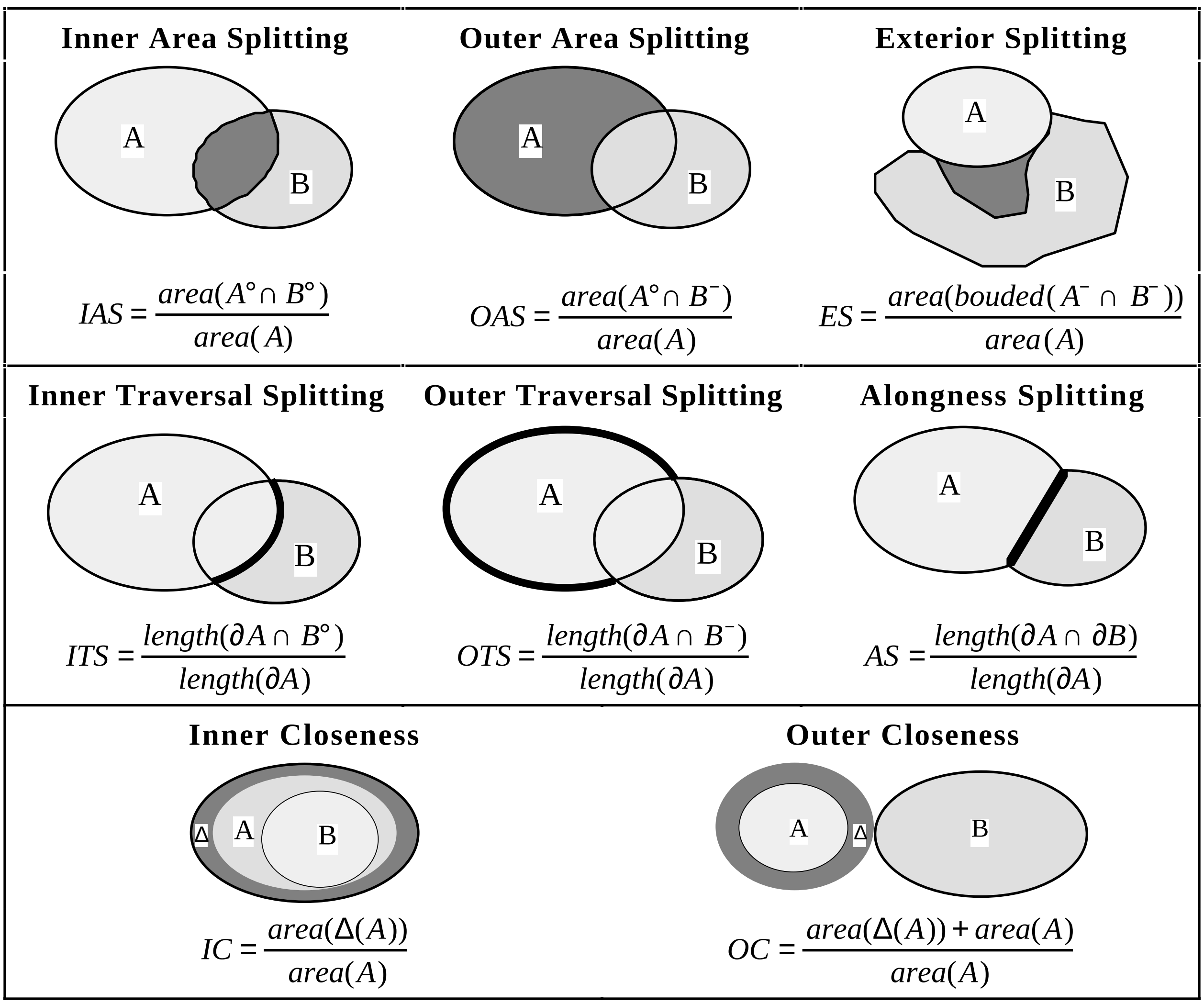


Figure 4: Metrical refinements of topological relations.

4.4 Cardinal Directions

Analogous to the role metrical properties may play in the interpretation of a scene, direction relations may provide a basis for certain decisions about matching and similarity. Direction relations are well understood for point objects; however, for extended spatial objects, such as linear or areal features, no generally accepted models exist and a variety of semantically different approaches have been proposed [45]. In this case, we adapt the projection-based method [23, 43] around a the minimum bounding rectangle of an object, partitioning space into nine regions for an areal object. These partitions are named north (N), northeast (NE), east (E), southeast (SE), south (S), southwest (SW), west (W), northwest (NW), and at the same location (0). The cardinal direction from an object to a target direction is described by recording the partitions into which at least some parts of the target object fall (Figure 5). Further refinements would be possible to describe if a target's outline coincides with the boundaries between partitions [24].

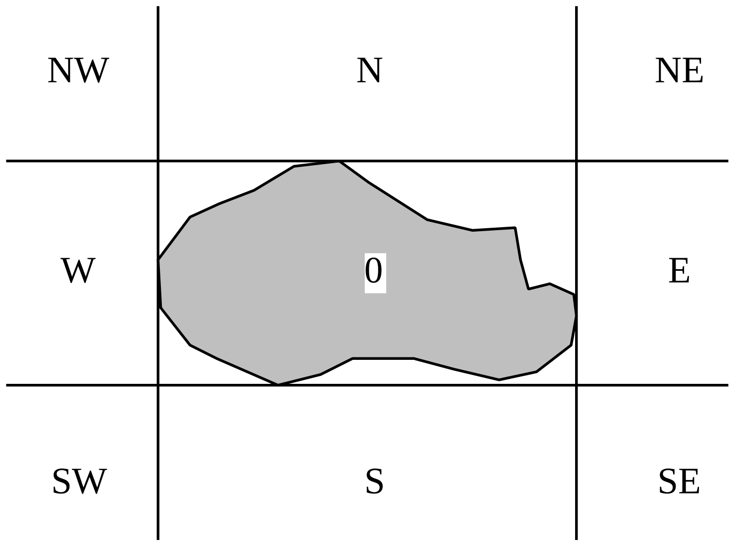


Figure 5: Projection-based cardinal directions for extended spatial objects.

4.5 Detailed Cardinal Directions

Cardinal directions are often a coarse approximation such that an interpretation of the mere fact that an object falls within some direction partition(s) of another object may be misleading or inappropriate. To provide more detail about directions among objects, we extend the cardinal-direction method, recording for each object that falls into more than one direction partition the percentage of the common intersection between a partition and the object (Figure 6). The range of each detailed cardinal direction x is $0 < x < 1.0$. The sum of all percentages for an object with respect to the partitions of another object must be 1.0. The refinement measure does not apply to empty partitions nor would it provide any additional information if the entire object falls into a single partition.

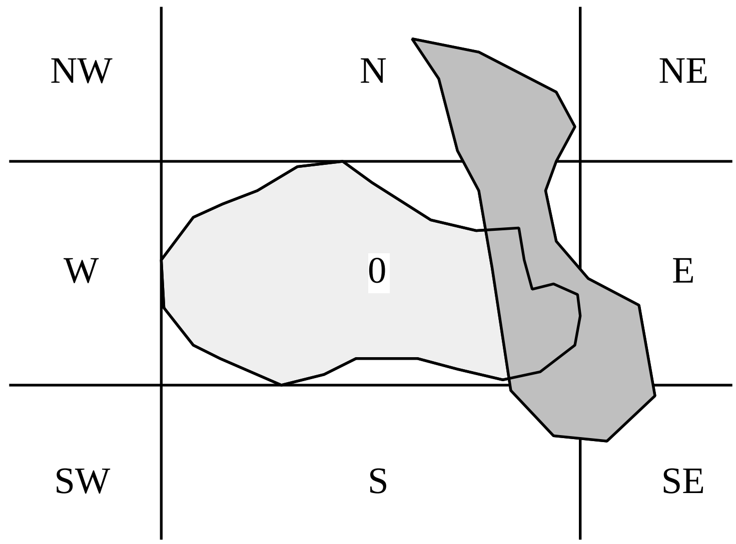
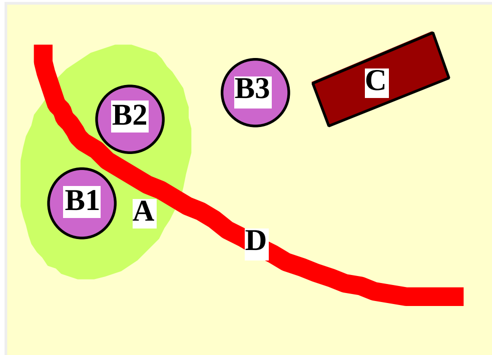


Figure 6: The cardinal directions are described by the partitions in which object *B* fall (N, E, SE, S, 0) and the detailed cardinal directions are described by the percentages that *B* extends over the different partitions (N=28%, 0=42%, E=16%, SE= 8%, S=6%).

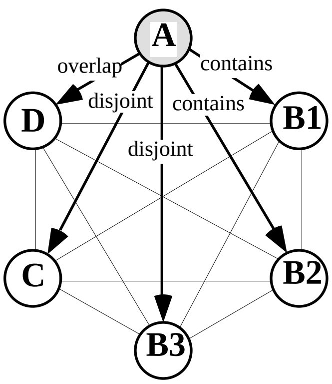
4.6 An Example

Figure 7 illustrates the use of the five types of spatial relations for the construction of a scene network. The sketch (Figure 7a) shows six areal objects, which received unique identifiers that also reflect the objects' classes. These identifiers are used consistently throughout the scene network. Figures 7b-f show subsets of the scene network, focusing on the spatial relations with respect to object **A**. Corresponding representations exist for each other object in the scene network. Figure 7b depicts for object **A** the binary topological relations that were derived from the 9-intersection. Details about these topological relations are described in Figure 7c through the component invariant table for those relations where detailed topological descriptions are possible (in this case for the relation *overlap* between objects **A** and **D**). The sketch's splitting and closeness ratios for metrical details are shown in Figure 7d. For the two *disjoint* relations with objects **B3** and **C**, only the outer closeness (OC), i.e., the relative distance to **A**, applies. The two *inside* relations with objects **B1** and **B2** are specified by the relative sizes of their areas with respect to **A**'s area—inner area splitting (IAS) and outer area splitting (OAS)—and how far **A**'s boundary is from the enclosed objects—the inner closeness (IC). The greatest detail is provided for the *overlap* relation between **A** and **D**: the inner area splitting (IAS), which describes how much of **D**'s interior coincides with **A**'s interior; the outer area splitting (OAS) for the portion of **A** that is outside of **D**; the inner traversal splitting (ITS), which captures the length of **A**'s perimeter that runs through **D**'s interior; the length of **A**'s perimeter that is outside of **D**—the outer traversal splitting (OTS); and the length of the common boundaries, called the alongness splitting (AS). Figures 7e and 7f describe the direction relations at a coarse and a more detailed level. Coarse cardinal directions are determined through a grid that aligns with **A**'s minimal bounding rectangle. Three objects fall completely into a single partition (**B1**, **B2**, and **B3**), while **C** and **D** span respectively over two and three partitions. Details about the distribution over multiple partitions are captured in Figure 7f, recording by how much an object extends over multiple partitions.

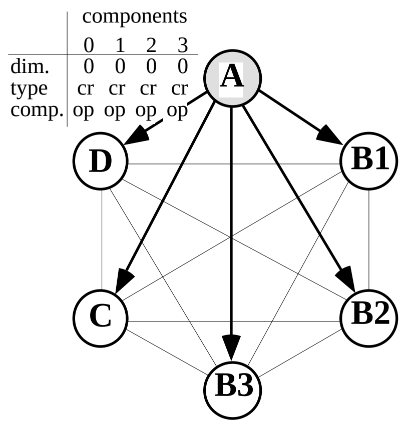
Journal of Visual Languages and Computing, Vol. 8, No. 4, pp. 403-424, 1997.



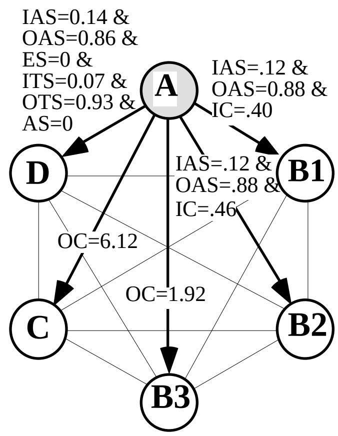
(a)



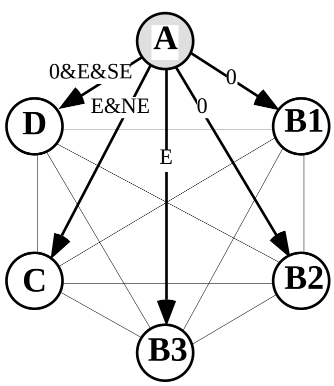
(b) topological relations



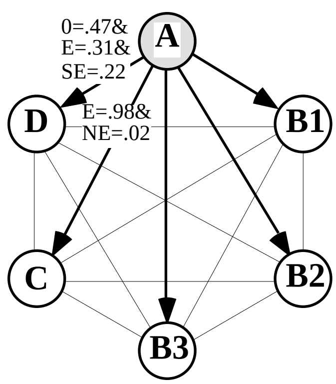
(c) detailed topological relations



(d) metric refinements



(e) cardinal directions



(f) detailed cardinal directions

Figure 7: A sketch (a) and the scene networks of (b) topological relations, (c) detailed topological relations, (c) metrical refinements, (d) cardinal directions, and (e) detailed cardinal directions (only the relations for object **A** are shown).

5. Processing Sketched Queries

The scene network forms the basis for processing a sketched query and for presenting the query results in a prioritized order to the user. The spatial relations captured in the scene network relate to different query processing stages, because the relations are of different levels of importance for capturing the semantics of a spatial scene; therefore, different strategies may be pursued, such as multiple querying of a geographic database using each time a different part of the scene description.

We use the 9-intersection relations as the key for pre-processing spatial relations sketched and querying, because they describe topological relations at a coarse level and, therefore, group sketches into classes of similar relations. By mapping the sketched relations onto 9-intersection relations, we capture the most salient features of a sketch in a form that is independent of orientations and sizes. This abstraction is critical for the translation of a sketched configuration into a database query.

The component invariants are the key for analyzing the intentional complexity of the spatial relations sketched, because the component invariants capture complexity of topological relations. A greater number of component intersections indicates more complexity [12]. If a user draws a sketch with a high level of complexity, then we assume that this complexity was intended and that it provides the lower bound of what should be retrieved; therefore, a configuration in a spatial database with the same 9-intersection relation, but lower-rated component invariants, would not qualify as a match. On the other hand, a sketch of a low-complexity spatial relation may indicate that more complex configurations under the same 9-intersection category should be considered as well.

With the metrical refinements of the 9-intersection relations we formalize detailed geometric constraints about sketched spatial relations. In Spatial-Query-by-Sketch, metrical details play two roles. First, they are critical to decide whether the query processor should also search for configurations that deviate from the topology sketched. For instance, a particularly short closeness measure for two disjoint regions may indicate that the user also would accept as an answer a configuration in which the two objects meet topologically. Second, the metrical properties are the key to prioritizing query results that have the same topology as the sketch, but differ in relative sizes of the objects, common lengths and areas, and distances between boundaries.

We exploit the cardinal directions for those queries in which the user explicitly states the importance of orientation relations, for instance, if the user drew a north arrow to give a global orientation to the sketch. Cardinal directions are also used to prioritize query results, i.e., as a tie-breaker among configurations with the same topology.

Finally, detailed cardinal directions are used to rank the query results such that the situation that matches most closely the sketch is presented first. Unlike the use of topological and detailed topological relations, the transition from coarse to detailed cardinal directions is used as a mere refinement and no intended complexity of the configuration is derived from the detailed direction relations.

A strategy that makes use of the dependencies among the different types of spatial relations is outlined below. Since the five types of spatial relations play different roles in the interpretation of a sketch, we employ a multi-step query processing strategy.

- First, the topological scene description is used to formulate a spatial database query. A topological relation is relaxed if its metrical refinements have small values indicating that alternative topological configurations may be considered as well. In addition, if the user specifies explicitly an orientation of the sketch, the cardinal directions are incorporated into the query.
- Second, for a non-empty result set of such a spatial query, each configuration is analyzed according to topological details, metrical details, and detailed cardinal directions, eliminating false hits and prioritizing the remaining configurations.
- If the query result is an empty set (i.e., no configuration was found that matches the relations specified), the initial constraints may be relaxed, from which a revised spatial query gets formulated.

6. Relaxing Spatial Relations

The comparison of the sketched spatial relations with the spatial relations recorded in a geographic database may not necessarily provide an exact match. The sketch may, for instance, be distorted leading to different relations than intended or the user may be satisfied with a configuration that is an approximation of what he or she drew. For this purpose it is necessary to consider not only exact matches, but also similar matches [3, 41]. The challenging aspect of determining similar configurations is that spatial relations represent discrete concepts that are usually thought of as being on a nominal scale. In order to assess similarity among elements, however, it is necessary to introduce some non-arbitrary order over the elements. Spatial-Query-by-Sketch establishes similarity over different types of spatial relations through a formal model, including a metric, that assesses deviations of a spatial relation from a target relation.

An important basis for the similarity assessment is Stevens's categorization of scales of measurements, which distinguishes nominal, ordinal, interval, and ratio type data [50]. Topological relations are discrete values on a nominal scale, therefore, no linear order can be established among them. Similarity among topological relations is described in terms of the conceptual neighborhood graph, which links most similar relations to each other. It is based on the computational model of determining for each relation those relations with the least number of differences in the 9-intersection matrices. For instance, *disjoint* and *meet* are conceptually closer to each other than *disjoint* and *overlap*, because *disjoint* and *meet* differ in one entry in their 9-intersections—they have different boundary-boundary intersections—while *disjoint* and *overlap* differ in four entries. Conceptual neighborhood graphs have been derived for the eight region-region relations [11] (Figure 8), line-line relations [21], and line-region relations [18].

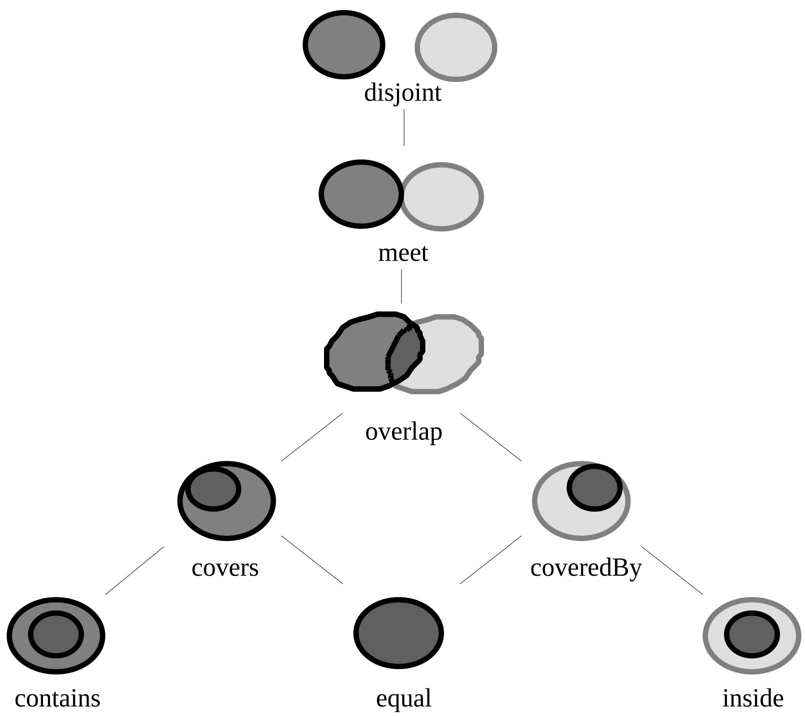


Figure 8: Conceptual neighborhood graph of the eight region-region relations.

Relaxing a topological relation corresponds to changing a constraint from a topological relation to include its conceptual neighbors. For example, if a user drew a scenario in which a region was fully included in the interior of another region, then a relaxation would consider not only those configurations that match exactly its topological relation, but also those that match the relation's conceptual neighbors [3]. Figure 9 shows the relaxation of all topological constraints for the sketch in Figure 7a. Higher degrees of dissimilarity can be achieved by recursively moving from the conceptual neighbors to their conceptual neighbors (without moving back). The more the topological relation gets relaxed, the less similar a relation becomes to its target.

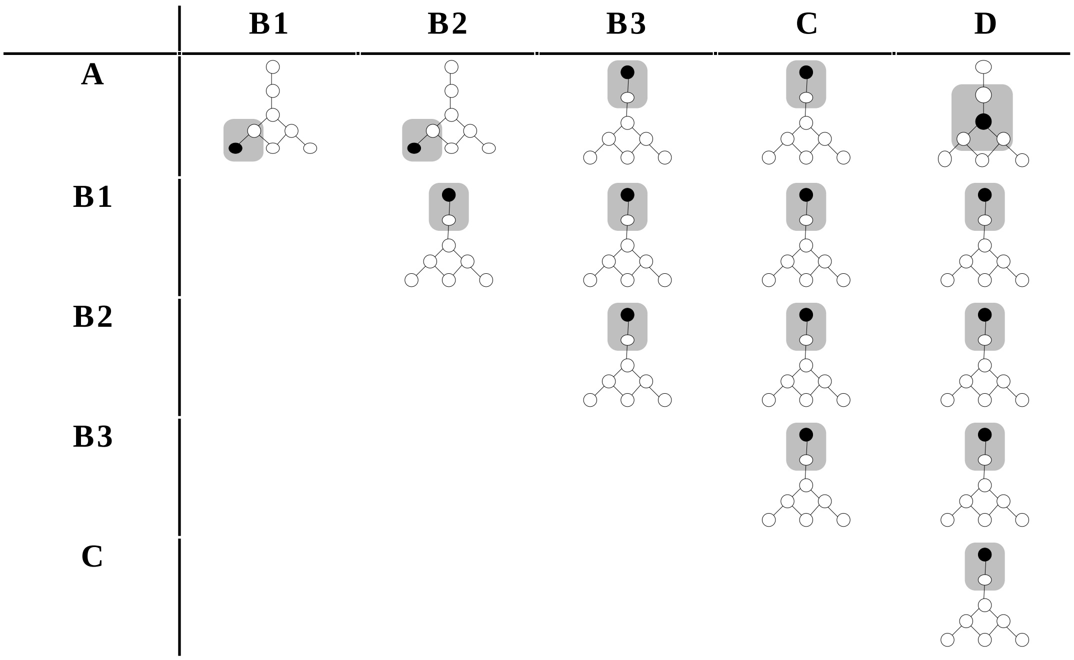


Figure 9: First-degree relaxation of all topological relations for the objects in figure 7a.

A more controlled way of relaxing topological relations exploits the metrical details as well. Metrical details are refinements of topological relations and a small value for a particularmetrical detail indicates that an alternative topological relation may be considered as well. For example, the configuration in Figure 7a shows objects **A** and **B3** being *disjoint*, but close together (the outer closeness from **A** to **B3** is 1.92), whereas **A** and **C** are *disjoint* as well, but further apart from each other (the outer closeness from **A** to **C** is 6.12). If a threshold for the outer closeness was set to 2.0, **A** *disjoint* **B3** may be relaxed into **A** *disjoint or meet* **B3**. This type of relaxation implies a direction on the conceptual neighborhood graph, i.e., not all conceptual neighbors are used. For example, an *overlap* with a small value for inner area splitting, but large value for outer area splitting, gets relaxed from *overlap* to *overlap or meet*, but not *covers* or *coveredBy*.

If cardinal directions are an explicit part of a sketch, they may be subject to relaxation during query processing in the same way topological relations are. For this purpose, it is necessary to model the conceptual neighborhoods of cardinal directions such that the most similar direction can be determined. For the projection-based model with nine direction values, a simple model arranges the relations in a 3×3 grid, reflecting the nine partitions such that conceptual neighborhoods are established both in horizontal and vertical directions (Figure 10). The conceptual neighbor of a direction relation are then its immediate horizontal and vertical neighbors in the graph. If an object extends through more than one partition, the conceptual neighbors of its cardinal direction comprise the union of the neighbors of each relation in the set.

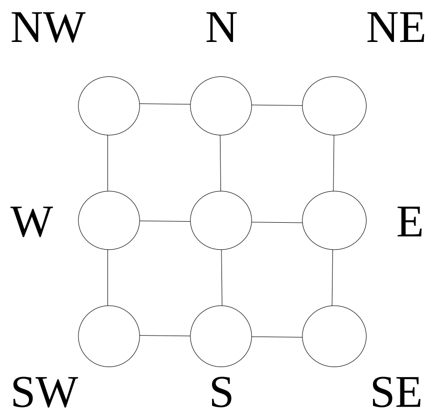


Figure 10: The conceptual neighbors of the nine projection-based cardinal directions.

Figure 11 shows the first-degree relaxation of all cardinal directions for the sketch depicted in Figure 7a. Only the direction relations between and object and itself—located in the diagonal of the table—cannot be relaxed. Unlike the topological relations, the cardinal directions do not form converse pairs and, therefore, it is necessary to consider in the relaxation of direction relations $n^2 - n$ relations among n objects.

Journal of Visual Languages and Computing, Vol. 8, No. 4, pp. 403-424, 1997.

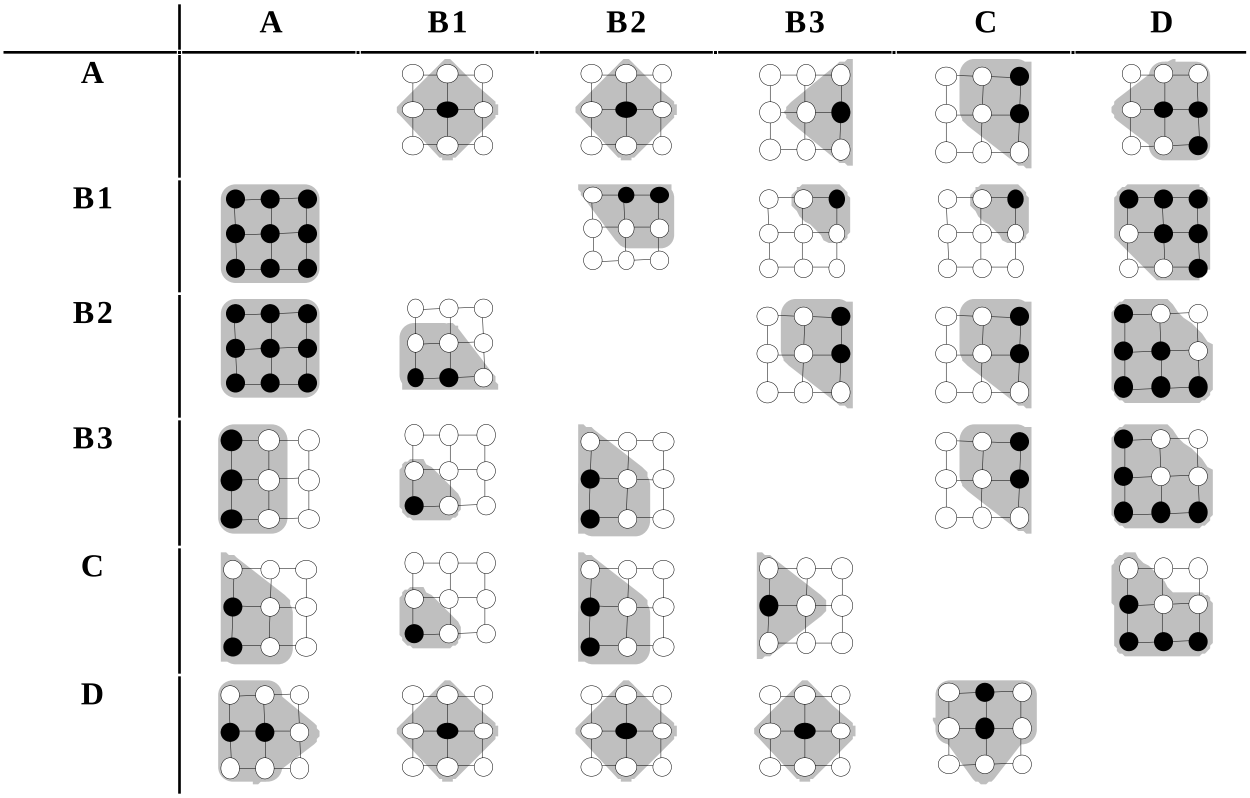
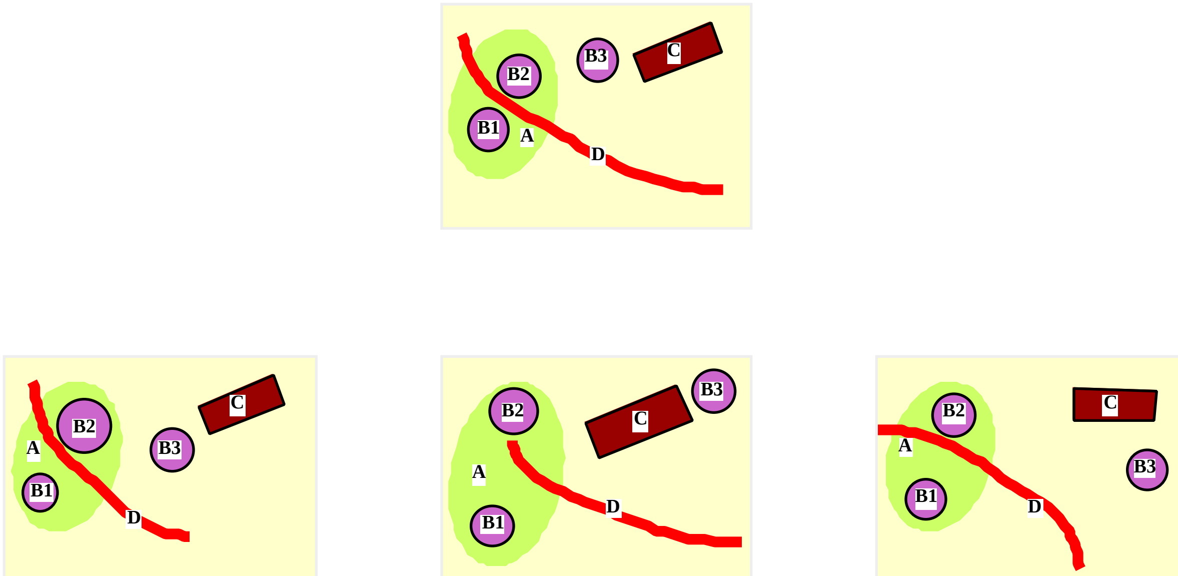


Figure 11: First-degree relaxation of all cardinal directions among the objects in Figure 7a.

7. Prioritizing Query Results

Independent of whether the query is relaxed or not, the result that is returned from a spatial database may contain multiple configurations, all of which fulfill the constraints of the query (Figure 12); however, in such a set of configurations there will be some that fit the original sketch better than others. Since the query acts as a filter, it is necessary to sort through the query result during the subsequent phase of *query result prioritization* and to rank the configurations retrieved according to their similarity with the sketch. This assessment requires *difference measures* for all five types of spatial relations. All measures introduced are such that lower difference values represent more similar configurations, while larger values indicate more differences. A value of 0 indicates no difference according to the type of spatial relation.



Coarse topological difference: 0
Detailed topological difference: 0
Metrical difference: 14.6
Coarse direction difference: 17
Detailed direction difference: 9.9

Coarse topological difference: 0
Detailed topological difference: 3
Metrical difference: N/A
Coarse direction difference: 43
Detailed direction difference: 20.1

Coarse topological difference: 0
Detailed topological difference: 0
Metrical difference: 62.5
Coarse direction difference: 40
Detailed direction difference: 16.0

Figure 12: The sketched configuration (top) and three configurations retrieved based on coarse topological constraints (bottom).

For coarse topological relations, the measure is the number of differences in the conceptual neighborhoods found between the sketch and the configuration in the query result. For example, if two objects are related by a *disjoint* relation in the sketch, while the corresponding objects *overlap* in the query result, then their coarse topological difference would be 2, because it takes two steps along the conceptual neighborhood graph of topological relations to get from *disjoint* to *overlap*. For the example in Figure 12, all three configurations have the same coarse topological relations between all corresponding pairs of objects as the target, therefore, the coarse topological difference is 0 for all three configurations contained in the query result.

For detailed topological relations, the difference measure is the number of elementary deformations (i.e., adding or removing an intersection) that are necessary to obtain topological equivalence between the sketch and the configuration in the query result [3]. In Figure 12, the first and third configuration have the same detailed topology as the target; however, the second configuration differs in the way objects **A** and **D** *overlap*. In order to transform one into the other, two elementary deformations are necessary; therefore, the detailed topological difference for this configuration is 2, while it is 0 for the other two configurations.

Metrical details are used only for configurations that expose the same counts for detailed topological differences. For each metrical parameter, we calculate the ratio between the result configuration and the target, and take the absolute value of the deviation from 1. If both are exactly the same their ratio is 1 and, therefore, the metrical difference 0. For each pair of scenes, the metrical difference measure is the sum of all metrical ratios. For the example in Figure 12, the metrical difference measures apply to the first and third configuration, because they have the same detailed topological difference. The third configuration turns out to be metrically less similar to the query than configuration one, because **B1** is more distant from **D** and **B3** is more remote from **A**, **B1**, and **B2**; therefore, their outer closeness ratios are much higher, while the remaining metrical ratios are relatively small, giving the third configuration a much higher metrical difference measure than the first.

As the measure for coarse direction differences we determine for each corresponding pair of objects the shortest path between their cardinal directions along their conceptual neighborhoods. For example, the shortest path from N to SW is 3; from NW to SE it is 4; from N to N it is 0, from NW and W to SW and S it is 4; and from NW and W to SW it is 3. The direction difference measure between two configurations is the sum of all shortest paths along the conceptual neighborhoods. For the example in Figure 12, the first configuration in the query result has the best match with respect to the target. The other two configurations differ more strongly from the sketch primarily due to the significantly different locations of objects **B3** and **C**, which leads to high scores for the direction differences of the relations between **B3** and **C**, but also between **B3** and **B2**.

The difference measure for detailed cardinal directions gives weights to the counts of the steps of the coarse direction differences according to the percentages of changes. For example, from the detailed direction relations E (0.98) and NE (0.02) to E (1.00), the weighted count is 0.02, while from E (0.98) and NE (0.02) to NE (1.00) would be 0.98; the count from E (0.10) and SE (0.70) and S (0.20) to S (1.00) is 2 * 0.10 + 1 * 0.70 = 0.90. The difference measure for detailed directions also applies if both directions span the same partitions, but are differently distributed. For example, the weighted count from E (0.98) and NE (0.02) to E (0.04) and NE (0.96) is 0.98 – 0.04 = 0.94. For the example in Figure 12, the detailed direction differences confirm the similarity rankings of the query results obtained from the coarse direction differences. Comparisons with the detailed direction differences, however, may lead to different results than the coarse direction differences if the individual counts are primarily small values.

8. Conclusions

This paper presented the design principles of Spatial-Query-by-Sketch, a visual spatial query language for geographic information systems. Users interact through Spatial-Query-by-Sketch by using a pen to draw an example of the configuration they are interested in. Spatial-Query-by-Sketch parses this graphical input, analyzes it, and translates it into a database query. We base its query processing mechanisms on a powerful computational model for spatial relations that allows us to emphasize cognitively important criteria of the sketch, and to suppress aspects that may be of lesser importance. Spatial-Query-by-Sketch uses five types of spatial relations: coarse topological relations, detailed topological relations, metrical details, coarse cardinal directions, and detailed cardinal directions. This approach is tailored for geographic similarity retrieval, where frequently the orientation, size, and shape of an object may not matter, but the relationship with respect to other objects is critical. Each set of spatial relations is formally defined, including computational methods to address similarity among pairs of relations of the same type. These models of spatial relations are used when translating the sketch into a database query, when relaxing spatial constraints, and when sorting query results according to highest similarity with the sketch.

A prototype of Spatial-Query-by-Sketch is under development with methods implemented for the assessment and relaxation of topological and direction relations. From experiments with the prototype we expect to gain new insights into the match between different querying strategies and people’s intuition about similarity retrieval. Such future experiments also will provide us with guidance as to whether and how a single similarity measure as a weighted combination of the five individual measures would useful. To work efficiently with a spatial database system, indexing and access methods for spatial relations are required that would support the query execution in a database system. Current spatial access methods are limited to the location of spatial objects and, therefore, only support fast retrieval based on coordinate values, such as point-in-polygon or window queries. Since Spatial-Query-by-Sketch is based on a different paradigm—spatial relations rather than location in space—either a mapping onto existing methods or the development of new methods will be necessary.

Spatial-Query-by-Sketch is on the opposite scale of a verbal spatial query language. While drawing spatial configurations is an intuitive interaction with geographic data, there are some spatial concepts, such as intentional orientation, distance, or shape, that may be difficult to express through a sketch alone. A sketch-based spatial query language may benefit from an embedding into a multi-modal interaction, where sketching may be augmented by verbal instructions [9].

9. Acknowledgments

Discussions with Doug Flewelling, Bob Franzosa, Dimitris Papadias, and Jayant Sharma helped in the development of the concepts in Spatial-Query-by-Sketch. Thanks also to the Spatial-Query-by-Sketch team, which includes Peggy Agouris, Andreas Blaser, Tom Bruns, James Craswell, Yves Dennebouy, Roop Goyal, João Paiva, Dieter Pfoser, Uwe Rupp, Rashid Shariff, and Tony Stefanidis.

10. References

[1] M.-A. Aufaure-Portier, "Definition of a Visual Language for GIS," *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems,* T. Nyerges, D. Mark, R. Laurini, and M. Egenhofer, eds., vol. Kluwer, Dordrecht, 1995, pp. 163-178.

[2] A. Borning, "Defining Constraints Graphically," *Human Factors in Computing Systems, CHI `86,* Boston, MA, 1986, pp. 137-143.

[3] T. Bruns and M. Egenhofer, "Similarity of Spatial Scenes," *Seventh International Symposium on Spatial Data Handling,* M.-J. Kraak and M. Molenaar, eds., Delft, The Netherlands, 1996, pp. 4A.31-42.

[4] D. Calcinelli and M. Mainguenaud, "Cigales, a Visual Query Language for a Geographical Information System: the User Interface," *Journal of Visual Languages and Computing,* vol. 5, no. 2, pp. 113-132, 1994.

[5] N.S. Chang and K.S. Fu, "Query-by-Pictorial-Example," *IEEE Transactions on Software Engineering,* vol. SE-6, no. 6, pp. 519-524, 1980.

[6] A. Del Bimbo, P. Pala, and S. Santini, "Visual Image Retrieval by Elastic Deformation of Object Sketches," *IEEE Symposium on Visual Languages,* IEEE Computer Society Press, St. Louis, MO, 1994, pp. 216-223.

[7] M. Egenhofer, "Why not SQL!," *International Journal of Geographical Information Systems,* vol. 6, no. 2, pp. 71-85, 1992.

[8] M. Egenhofer, "Pre-Processing Queries with Spatial Constraints," *Photogrammetric Engineering & Remote Sensing,* vol. 60, no. 6, pp. 783-790, 1994.

[9] M. Egenhofer, "Multi-Modal Spatial Querying," *Seventh International Symposium on Spatial Data Handling,* M.-J. Kraak and M. Molenaar, eds., Delft, The Netherlands, 1996, pp. 12B.11-15.

[10] M. Egenhofer, "Spatial-Query-by-Sketch," *VL '96: IEEE Symposium on Visual Languages,* IEEE Computer Society, Boulder, CO, 1996, pp. 60-67.

[11] M. Egenhofer and K. Al-Taha, "Reasoning About Gradual Changes of Topological Relationships," *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Pisa, Italy,* A. Frank, I. Campari, and U. Formentini, eds., *Lecture Notes in Computer Science*, vol. 639, Springer-Verlag, Berlin, 1992, pp. 196-219.

[12] M. Egenhofer, E. Clementini, and P. Di Felice, "Evaluating Inconsistencies Among Multiple Representations," *Sixth International Symposium on Spatial Data Handling,* T. Waugh and R. Healey, eds., Edinburgh, Scotland, 1994, pp. 901-920.

[13] M. Egenhofer and R. Franzosa, "Point-Set Topological Spatial Relations," *International Journal of Geographical Information Systems,* vol. 5, no. 2, pp. 161-174, 1991.

[14] M. Egenhofer and R. Franzosa, "On the Equivalence of Topological Relations," *International Journal of Geographical Information Systems,* vol. 9, no. 2, pp. 133-152, 1995.

[15] M. Egenhofer and J. Herring, "A Mathematical Framework for the Definition of Topological Relationships," *Fourth International Symposium on Spatial Data Handling,* K. Brassel and H. Kishimoto, eds., Zurich, Switzerland, 1990, pp. 803-813.

[16] M. Egenhofer and J. Herring, *Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases,* Department of Surveying Engineering, University of Maine, Orono, ME, 1991.

[17] M. Egenhofer and J. Herring, "Querying a Geographical Information System," *Human Factors in Geographical Information Systems*, D. Medyckyj-Scott and H. Hearnshaw, eds., vol. Belhaven Press, London, 1993, pp. 124-136.

[18] M. Egenhofer and D. Mark, "Modeling Conceptual Neighbourhoods of Topological Line-Region Relations," *International Journal of Geographical Information Systems,* vol. 9, no. 5, pp. 555-565, 1995.

[19] M. Egenhofer and D. Mark, "Naive Geography," *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT '95, Semmering, Austria,* A. Frank and W. Kuhn, eds., *Lecture Notes in Computer Science,* vol. 988, Springer-Verlag, Berlin, 1995, pp. 1-15.

[20] M. Egenhofer and J. Sharma, "Assessing the Consistency of Complete and Incomplete Topological Information," *Geographical Systems,* vol. 1, no. 1, pp. 47-68, 1993.

[21] M. Egenhofer, J. Sharma, and D. Mark, "A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis," *Autocarto 11,* R. McMaster and M. Armstrong, eds., Minneapolis, MN, 1993, pp. 1-11.

[22] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petrovic, and W. Equitz, "Efficient and Effective Querying by Image Content," *Journal of Intelligent Information Systems,* vol. 3, no. pp. 231-262, 1994.

[23] A. Frank, "Qualitative Spatial Reasoning about Cardinal Directions," *Autocarto 10,* D. Mark and D. White, eds., Baltimore, MD, 1991, pp. 148-167.

[24] C. Freksa, "Using Orientation Information for Qualitative Spatial Reasoning," *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space,* A. Frank, I. Campari, and U. Formentini, eds., *Lecture Notes in Computer Science*, vol. 639, Springer-Verlag, New York, NY, 1992, pp. 162-178.

[25] M. Gross, "Recognizing and Interpreting Diagrams in Design," *Advanced Visual Interfaces '94,* T. Catarci, M. Costabile, S. Levialdi, and G. Santucci, eds., ACM Press, New York, NY, 1994, pp. 88-94.

[26] V. Gudivada and V. Raghavan, "Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity," *ACM Transactions on Information Systems,* vol. 13, no. 2, pp. 115-144, 1995.

[27] J. Herring, "The Mathematical Modeling of Spatial and Non-Spatial Information in Geographic Information Systems," *Cognitive and Linguistic Aspects of Geographic Space,* D. Mark and A. Frank, eds., vol. Kluwer Academic Publishers, Dordrecht, 1991, pp. 313-350.

[28] J. Herring, R. Larsen, and J. Shivakumar, "Extensions to the SQL Language to Support Spatial Analysis in a Topological Data Base," *GIS/LIS '88,* San Antonio, TX, 1988, pp. 741-750.

[29] K. Hirata and T. Kato, "Query by Visual Example—Content-Based Image Retrieval," *Advances in Database Technology—EDBT '92, 3rd International Conference on Extending Database Technology, Vienna, Austria,* A. Pirotte, C. Delobel, and G. Gottlob, eds., *Lecture Notes in Computer Science*, vol. 580, Springer-Verlag, Berlin, 1992, pp. 56-71.

[30] K. Hirata and T. Kato, "Rough Sketch-Based Image Information Retrieval," *NEC Research and Development*, vol. 34, no. 2, pp. 263-273, 1993.

[31] T. Imielinski and H. Korth, "MOBIDATA Workshop Report," *MOBIDATA: An Interactive Journal of Mobile Computing,* vol. 1, no. 2, pp. http://www.cs.rutgers.edu/~badri/journal/contents12.html, 1995.

[32] K. Ingram and W. Phillips, “Geographic Information Processing Using a SQL-Based Query Language,” *AUTO-CARTO 8, Eighth International Symposium on Computer-Assisted Cartography,* N.R. Chrisman, eds., Baltimore, MD, 1987, pp. 326-335.

[33] T. Joseph and A. Cardenas, “PICQUERY: A High Level Query Language for Pictorial Database Management,” *IEEE Transactions on Software Engineering,* vol. 14, no. 5, pp. 630-638, 1988.

[34] T. Kato, T. Kurita, N. Otsu, and K. Hirata, “A Sketch Retrieval Method for Full Color Image Database—Query by Visual Example,” *11th IAPA International Conference on Pattern Recognition*, IEEE Computer Society Press, The Hague, The Netherlands, 1992, pp. 530-533.

[35] T. Kato, T. Kurita, and H. Shimogaki, “Multimedia Interaction with Image Database Systems,” *Advanced Database System Symposium ‘89,* Kyoto, Japan, 1989, pp. 271-278.

[36] Y.C. Lee and F.L. Chin, “An Iconic Query Language for Topological Relationships in GIS,” *International Journal of Geographical Information Systems,* vol. 9, no. 1, pp. 25-46, 1995.

[37] M. Mainguenaud and M.-A. Portier, “Cigales: A Graphical Query Language for Geographical Information Systems,” *Fourth International Symposium on Spatial Data Handling,* K. Brassel and H. Kishimoto, eds., Zurich, Switzerland, 1990, pp. 393-404.

[38] D. Mark, “Counter-Intuitive Geographic “Facts:” Clues for Spatial Reasoning at Geographic Scales,” *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Pisa, Italy,* A. Frank, I. Campari, and U. Formentini, eds., *Lecture Notes in Computer Science,* vol. 639, Springer-Verlag, Berlin, 1992, pp. 305-317.

[39] D. Mark, D. Comas, M. Egenhofer, S. Freundschuh, M. Gould, and J. Nunes, “Evaluating and Refining Computational Models of Spatial Relations Through Cross-Linguistic Human-Subject Testing,” *Spatial Information Theory—A Theoretical Basis for GIS, International Conference COSIT ‘95, Semmering, Austria,* A. Frank and W. Kuhn, eds., *Lecture Notes in Computer Science*, vol. 988, Springer-Verlag, Berlin, 1995, pp. 553-568.

[40] B. Meyer, “Pictorial Deduction in Spatial Information Systems,” *IEEE Symposium on Visual Languages,* A. Ambler and T. Kimura, eds., IEEE Computer Society Press, St. Louis, MO, 1994, pp. 23-39.

[41] N. Nabil, A. Ngu, and J. Shepherd, “Picture Similarity Retrieval Using the 2D Projection Interval Representation,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 8, no. 4, pp. 533-539, 1996.

[42] V. Ogle and M. Stonebraker, “Chabot: Retrieval from a Relational Database of Images,” *IEEE Computer*, vol. 28, no. 9, pp. 40-48, 1995.

[43] D. Papadias and T. Sellis, “A Pictorial Query-by-Example Language,” *Journal of Visual Languages and Computing,* vol. 6, no. 1, pp. 53-72, 1995.

[44] C. Papadimitriou, D. Suciu, and V. Vianu, “Topological Queries in Spatial Databases,” *Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS),* ACM Press, Montreal, Canada, 1995,

[45] D. Peuquet and C.-X. Zhan, “An Algorithm to Determine the Directional Relationship Between Arbitrarily-Shaped Polygons in the Plane,” *Pattern Recognition,* vol. 20, no. 1, pp. 65-74, 1987.

[46] A. Pizano, A. Klinger, and A. Cardenas, “Specification of Spatial Integrity Constraints in Pictorial Databases,” *Computer,* vol. 22, no. 12, pp. 59-71, 1989.

[47] N. Roussopoulos, C. Faloutsos, and T. Sellis, “An Efficient Pictorial Database System for PSQL,” *IEEE Transactions on Software Engineering,* vol. 14, no. 5, pp. 630-638, 1988.

[48] R. Shariff, *Natural Language Spatial Relations: Metric Refinements of Topological Properties*, Department of Spatial Information Science and Engineering, University of Maine, Orono, ME, 1996.

[49] A. Stevens and P. Coupe, “Distortions in Judged Spatial Relations,” *Cognitive Psychology,* vol. 10, no. pp. 422-437, 1978.

[50] S. Stevens, “On the Theory of Scales of Measurement,” *Science Magazine,* vol. 103, no. 2684, pp. 677-680, 1946.

[51] I. Sutherland, “SketchPad: A Man-Machine Graphical Communication System,” *AFIPS Spring Joint Computer Conference*, 1963, pp. 329-346.

[52] B. Tversky, “Distortions in Memory for Maps,” *Cognitive Psychology*, vol. 13, no. pp. 407-433, 1981.