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
max@spatial.maine.edu

Abstract

Today's methods for interacting with geographic information systems (GISs) and geographic databases are primarily aspatial, as they require users to deal with geographic data primarily through alphanumeric command languages. Spatial querying by typing a command in some spatial query language or by selecting the same syntax from pull-down menus is a tedious process, because it often requires extensive training in the use of the particular query language, and forces users to translate a spatial image they may have in their mind into a non-spatial language. To overcome this conceptual gap, we propose Spatial-Query-by-Sketch, a sketch-based GIS user interface that focuses on specifying spatial relations by drawing them. This query style supports more directly human spatial thinking, which is critical, because users frequently have an image-like representation in their minds when they query about spatial configurations. This paper introduces the fundamental concepts of Spatial-Query-by-Sketch, provides examples of typical interactions, and discusses query-processing strategies by relaxing the constraints drawn in terms of a qualitative model.

1. Introduction

Traditional methods for spatial querying are tedious. The problems with communicating a user's request to a spatial database through conventional spatial query languages becomes most apparent when several users have to work together and have to understand their intentions. Verbal descriptions of spatial situations are frequently ambiguous and may easily lead to misinterpretations, particularly in multi-language groups. The use of spatial query languages has serious limitations when geographic concepts are used that are 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 the predicate relates to, and the topology and the metric of the particular configuration [21]. These drawbacks make 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 advanced interaction methods between users and geographic data represented 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. Rather than expressing a spatial query in lexical terms, users can *sketch* a spatial query.

Sketching more directly supports human spatial thinking than does interaction through a spatial query language, because users frequently have an image-like representation in their minds when they query about spatial configurations. Rather than 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 mind, in order to retrieve the spatial data of interest. This spatial query language, called *Spatial-Query-by-Sketch* allows users to express spatial queries closer to the way they think about many spatial problems and it incorporates powerful

---

* This work was partially supported by Rome Laboratories under grant number F30602-95-1-0042 and by the National Science Foundation under grant numbers SBR-8810917 (for the National Center for Geographic Information and Analysis) and IRI-9309230. Max Egenhofer's work is further supported by grants from the Scientific and Environmental Affairs Division of the North Atlantic Treaty Organization, Intergraph Corporation, and Environmental Systems Research Institute; 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.

M. Egenhofer (1996) Spatial-Query-by-Sketch, in: M. Burnett and W. Citrin (eds.), VL'96: IEEE Symposium on Visual Languages, Boulder, CO, IEEE Press, pp. 60-67.

reasoning mechanisms to infer geometric variations in the sketch. Sketching provides immediate graphical feedback and, therefore, is an inherently more natural process to formulate many spatial constraints than a textual language.

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 [15]. Pen-based user interfaces provide a series of advantages over current interaction techniques, particularly for the interaction with geographic data. By their very nature, geographic data are spatial, and it is most appealing to refer to them in terms of explicit spatial concepts. An area of particular interest is the access to digital image libraries [10, 23] through a language like Spatial-Query-by-Sketch, where users may want to retrieve remotely-sensed images on which features match a particular geometric configuration drawn. The use of Spatial-Query-by-Sketch in such demanding tasks as querying about physical terrain, e.g., to express line of sight between two places, will provide considerably easier use with instant feedback than current query languages do.

The remainder of this paper continues with a review of previous approaches to spatial querying, focusing on traditional spatial query languages, visual spatial query languages, and sketching. Section 3 discusses the difference between a sketch—a graphical representation of a spatial situation—and a symbolic representation. Section 4 introduces the principles of Spatial-Query-by-Sketch, and Section 5 gives a guided tour with a series of user interface snapshots. Section 6 provides background information to the query processors and Section 7 discusses the relaxation of sketched constraints. Conclusions 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 is a large variety of Spatial SQL dialects, see Egenhofer [5] for an overview. 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 covered by the SQL3 Multi-Media working group.

Similar to SQL extensions, there are several spatial query languages that are derivatives of Query-by-Example. Query-by-Pictorial-Example [4] and Picquery [16] 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 [3]. Unlike Spatial-Query-by-Sketch, Cigales requires the users to select the type of spatial relation they are addressing prior to drawing [20]. 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 [19] 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.

The only visual spatial query language that is based on a comprehensive algebra is Query-by-Visual-Example [24], 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. Sutherland's [26] Sketchpad and Borning's [2] ThingLab were initial approaches to formulate constraints graphically. Pizano *et al.* [25] 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 [13, 14, 17, 18] and Query by Image Content [10], 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—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 metric 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 (Del Bimbo *et al.* 1994). 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 a query language proposed in [22], however, Spatial-Query-by-Sketch is founded on a solid mathematical model of spatial relations and their relaxation.

3. Spatial-Query-by-Sketch

Spatial-Query-by-Sketch uses a touch-sensitive input device—ideally a touch screen with a pen, such as Apple's Newton. Simulations may be obtained with mice or trackballs, but sketching with these devices is more cumbersome and therefore less effective. A sketched query consists of five steps, ranging from the drawing of a spatial query to its execution against a database management system.

Step 1: The user draws with a pen a prototypical geometric configuration that matches closely the spatial situation(s) he or she expects to retrieve from the geographic database.

Step 2: The user annotates the sketch to describe desired properties of the sketched objects. These annotations may include the specification of the objects' classes (e.g., "lake," "forest," or "road"), object attributes (e.g., the name of a road), and metric constraints (e.g., distances or areas).

Step 3: Spatial-Query-by-Sketch parses the sketch and translates it from a pixel representation into a topological vector data model.

Step 4: Spatial-Query-by-Sketch develops a query processing plan. If necessary, ambiguities are resolved through a visual interview process with the user.

Step 5: The query processor executes the query against the spatial database and retrieves the scenes that match the query asked in a prioritized order, such that scenes with the best match to the query are retrieved first.

While these processes are conceptualized as steps, they need not occur in a linear fashion. For example, after drawing the first object (Step 1), the user may immediately annotate the geometry with its semantics (Step 2), before going back to a Step 1–operation and drawing the next object. The parser (Step 3) will act as soon as the user starts drawing and will not wait until the sketch has been completed (Step 1) and fully annotated (Step 2). Likewise, after analyzing the query, the system may interview the user and request additional clarifying information, which in turn may prompt the user to edit the sketch (by re-initiating additional tasks under Step 1 and 2). In a similar way, the examination of a query result (Step 5) may prompt a user to modify a query (Step 1 and 2).

4. A Guided Tour through Spatial-Query-by-Sketch

The following scenario provides a rough outline of the interaction a user may perform when sketching a query. This user interface is organized into three major interaction areas (Figure 1): (a) the sketch region in which the user draws the configuration of interest; (b) the overview area which displays the sketch in its entirety and allows users to pan and zoom; and (c) 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.

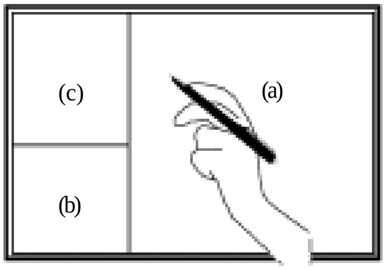

Figure 1. The layout of a Spatial-Query-by-Example user interface: (a) the sketching area, (b) the pan-and-zoom area, and (c) the location of the control panel.

The user employs a pen to sketch an example of what he or she wants 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 2).

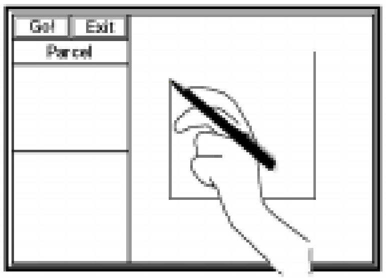

Figure 2. The user draws the geometry of a land parcel.

Then she describes the location of the forest by drawing part of the forest's boundary (Figure 3).

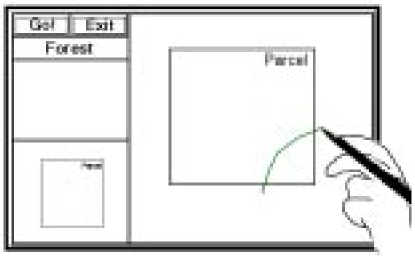

Figure 3. The user adds the boundary of a forest.

Since it is unclear on which side of the line the forest is located, the user fills the interior of the forest (Figure 4).

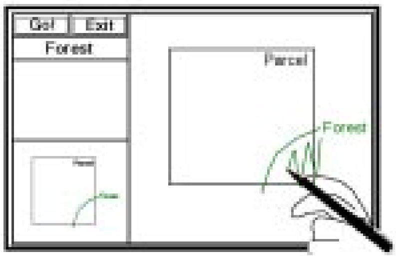

Figure 4. To determine the location of the forest, the user fills the forest's interior.

Finally the user draws a river such that it crosses the land parcel (Figure 5).

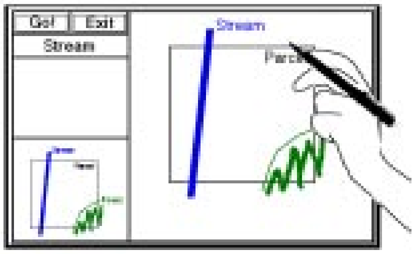

Figure 5. The user adds the location of a stream such that it crosses the land parcel, but does not intersect with the forest.

The user is satisfied with the drawing and requests that all configurations that match the sketch be retrieved from the database by pressing the **Go!** button on the control panel.

5. Processing Sketched Queries

Processing a sketched query is a non-trivial task, because the sketch needs interpretation as to which aspects the user drew intentionally, and which geometric constraints are accidental. This differs considerably from the processing of other spatial queries, where spatial relations are given explicitly and the semantics of the spatial relations are fixed. In Spatial-Query-by-Sketch, the query processor has to determine what to follow strictly and what may be relaxed.

Traditional spatial query languages make use of symbolic representations of spatial configurations and spatial constraints. A symbolic representation of a spatial configuration is like a verbal description of a spatial scene, and does not make use of any pictorial elements. Users who have an image in their mind, have to convert it into a symbolic form in order to describe their intentions. Feedback is based on reasoning about the spatial concepts expressed, but no direct match between the image in the user's mind and the query formulated is supplied.

Symbolic representations have the advantage that when used in a query language, a reference to a symbolic representation allows a user to focus exclusively on the particular constraints of interest, and abstract away other spatial aspects that may not be of concern for the particular query. For example, with a conventional SQL-like query language one can formulate a query to retrieve all lakes included in the state of Maine without having to specify the extent, orientation, or size of neither the lakes, nor the State. On the other hand, a user must make a correct translation from the image in his or her mind into the verbal expression. This process may give rise to some ambiguities because the mappings from pictorial representations to verbal descriptions are highly subjective.

A second source of imprecision arises when such a spatial query is processed against a spatial database. Current geographic databases record the geometry either in a vector- or a raster data model; therefore, a query processor has to understand the semantics of the spatial terms in order to know what to retrieve from the database. While recent research in the area of the semantics of natural-language spatial relations has produced initial promising results, this field of research is far from providing a complete coverage of the formalization of all spatial relations. A major impediment is the fact that spatial terms are often context-dependent such that their semantics may change from one situation to another. To date, only little knowledge exists about the use of such terminology in different situations [12, 21, 27].

Unlike the symbolic representation of a verbal query, any graphical representation is overdetermined, because it is embedded in a Euclidean space with all its properties associated. For example, if one draws a sketch for the configuration that a road crosses a park, this sketch does not only contain the topological relation between the two objects, but also additional constraints about relative sizes of the objects, their shapes, their orientations, and the proportions by which the road divides the park into separate pieces. From looking at the sketch it is unclear what constraints the user intended to specify, and what constraints just occurred by chance.

If the database were to retrieve only those configurations that provide an exact match with the drawing, standard methods in image matching and image retrieval could be applied. For geographic image processing, however, it may be necessary to relax some of the constraints of the sketch, because the user may have drawn certain parts of the scene that were accidental rather than intended. 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.

6. Modeling 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 [6, 8]. It characterizes the topological relation t 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, the categorization shows 8 distinct topological relations. They have been called *disjoint*, *meet*, *equal*, *overlap*, *inside*, *contains*, *covers*, and *coveredBy* (Figure 6). 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 [9].

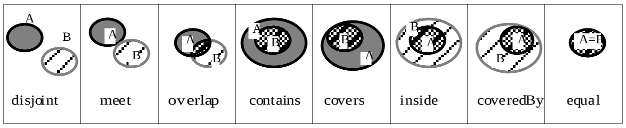

Figure 6. The eight topological relations that can be realized between two spatial regions.

6.1. Conceptual Neighborhoods

Similarity among topological relations is described in terms of the *conceptual neighborhood graph*, which links those relations that are most similar 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.

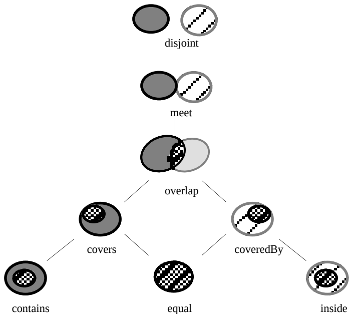

Figure 7. Conceptual neighborhood graph of the eight region-region relations.

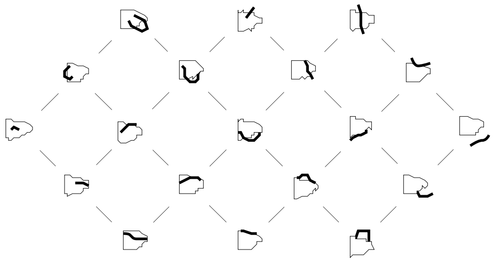

Figure 8. Conceptual neighborhood graph of the 19 line-region relations.

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. Figure 7 and 8 respectively show the conceptual neighborhoods for the 8 region-region relations (Egenhofer and Al-Taha 1992) and the 19 line-region relations (Egenhofer and Mark 1995a).

6.2. Relaxation of Topological Relations

The strategy of constraint relaxation is applied to the topological relation itself. *Relaxation of a topological relation* corresponds to changing from one topological relation to its conceptual neighborhoods. For example, if a user drew a scenario such that a line was fully included in the interior of a region, then the relaxation would consider not only those configurations that match exactly its topological relation, but also those that match its conceptual neighbors. Higher degrees of relaxation 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.

6.3. Query Processing

Spatial-Query-by-Sketch parses the sketch and extracts from the sketch's vector data model the topological relations as drawn. For n objects in a sketch, there are n^2 binary topological relations. Due to the equivalence of each object with itself, and implied converse relations, only $(n^2–n)/2$ binary relations have to be considered to describe the scene, though depending on the configuration even less relations may be sufficient (Egenhofer and Sharma 1993).

For the sketch in Figure 5, we obtain three significant relations: the parcel *overlaps* with the forest, the stream *crosses* the parcel, and the forest is *disjoint* from the stream. If necessary, these relations may be relaxed by also considering their topological neighbors as candidates (Figure 9). This relaxation provides flexibility to account for sketching inaccuracies. Such scene descriptions of binary topological relations translate immediately into database queries (Egenhofer 1994).

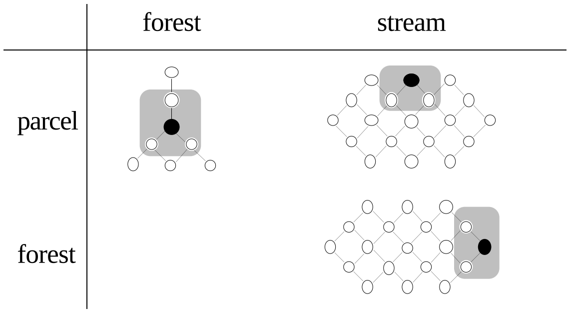

Figure 9. The binary topological relations (black) for the sketch in Figure 5, and their conceptual neighbors (gray).

6.4. Component Invariants for Topological Details

More detailed distinctions are possible if further criteria are employed to evaluate the non-empty boundary-boundary 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 such invariants for the components (or separations) of the boundary-boundary intersection only, since the other intersections can be inferred from them [7]. The necessary invariants to consider are the *sequence* of components counted along the boundaries; the *dimension* of each component; the *type* of boundary-boundary component intersection (*touch* or *cross*), where *crossing* may be further refined depending on whether the component *crosses into* or *crosses out* of the other region's interior; and the *complement relationship*, i.e., whether a component is a next to a *bounded* or *unbounded* exterior. Similar to the 9-intersection relations, topological details can be relaxed through changes in the component invariants. Unlike the conceptual neighborhood, however, which does not include a notion of complexity, the component invariants capture complexity (Table 1) if they are ordered (Egenhofer *et al.* 1994).

low complexity	→	high complexity
few boundary-boundary components		multiple boundary-boundary components
0-dimensional component		1-dimensional component
crossing intersections		touching intersections
unbounded complement		bounded complement

Table 1: Complexity of a spatial relation expressed in terms of component invariants.

6.5. Quantitative Refinements of Topological Relations

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 New Hampshire requires the consideration of some metric properties in addition to topological concern—I-95 divides New Hampshire into a very small area to the East of I-95 and a larger piece to the West. We apply such measures about areas, lengths, and directions as refinements of the topological properties. For line-region relations we use the following set of metric properties: the ratio by which a line's interior divides a region's interior; how much of a line's interior is inside a region's interior; how much of a line's interior coincides with a region's boundary; the ratio between the distance from a line to and region's boundary, and the line's total length; the ratio between the area made up by an equidistant enlargement (reduction) of the region (also known as a *buffer zone*) and the actual area; and the cardinal direction between the objects, measured in a qualitative orientation scheme such as orthogonal half planes with a neutral zone. Similarity schemes based on conceptual neighborhoods may be employed to relax direction or distance relations (Bruns and Egenhofer 1996) much like relaxing topological relations.

7. Relaxing Spatial Constraints Sketched

Relaxation of spatial constraints is necessary to process the sketch as a query, because trying to retrieve a situation that fits exactly the geometry of the sketch would only rarely result in a match. 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 *similarity retrieval* [11].

A critical part of the query parser and spatial query processor is the model for the interpretation of the sketch, particularly the spatial relations among the objects considered. Different users will inevitably draw sketches with a large variation even if they have the same configuration in their minds. Likewise, the same user will introduce random variations when repeating the same drawing task. Therefore, it is necessary to select a model that will be robust enough under such variations.

There is an important difference 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 proportions are considered to be known. The match between the picture and the sketch—the outline of some features that must appear on the image of interest—can 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; however, in queries about geographic data, this is not the case because the orientation of the objects may be immaterial for the query.

Experiments in psychology and cartography showed that topology is among the most critical information people react to when assessing spatial relationships in geographic space. On the other hand, metric changes are frequently considered to be of lesser importance. For example, if someone draws a line, representing a road, such that it separates two towns, one to the left and one to the right (Figure 10a), then this very separation is the most important information, and only to a lesser degree it matters how far apart the two towns are from the road (Figure 10b), of what shape the road is (Figure 10c), and in which direction the road leads (Figure 10d); therefore, situations 10b-d could be matches with sketch 10a. Figure 10e displays a significantly different situation, although the road and one of the towns are at the same location as in the sketch. However, the two towns are on the same side of the road, which is conceptually and cognitively a more serious difference than any of the deviations in Figures 10b-d.

Spatial-Query-by-Sketch is based on the premise *topology matters, metric refines* (Egenhofer and Mark 1995b). When users draw coincidence, inclusion, disjointness, or separations of features, then these specifications are considered to be in general more critical information than directions among spatial features, relative distances, lengths, areas, or other details. Unless a non-topological relation is explicitly specified, e.g., through gesture or voice annotation (Egenhofer 1996), they are not used as constraints to limit a search to match a sketch. Thus the set of answers is a close approximation of the topology drawn, but not necessarily of the metric. Metric constraints are employed, however, to determine a prioritized list from the query answers that fulfill the topological constraints. This leads to a two-phase query processing: (1) finding matches based on topological equivalence or topological similarity, and (2) sorting query results based on metric properties.

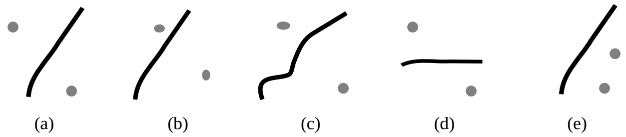

Figure 10. Four situations (a-d) with similar configurations, and one situation (e) with a significantly different configuration.

8. Conclusions

This paper presented the design principles of Spatial-Query-by-Sketch, a visual spatial query language without modes. We base its query processing on a powerful computational model that allows us to emphasize cognitively important criteria of the sketch, and to suppress aspects that may be of lesser importance. Based on the formal model for topological spatial relations, we developed a computational model for similarity of spatial relations that emphasizes topological properties, and refines them when metrical aspects are of concern. This approach is tailored for geographic similarity retrieval, where frequently the orientation, size, and shape of an object does not matter, but the relationship with respect to other objects is critical.

Spatial-Query-by-Sketch is on the opposite scale of a verbal spatial query language. While drawing spatial configurations is an intuitive way of interacting 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 (Egenhofer 1996).

9. 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, *et al.*, 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] 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. pp. 113-132, 1994.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] A. Herskovits, *Language and Spatial Cognition—An Interdisciplinary Study of the Prepositions in English,* Cambridge University Press, Cambridge, MA, 1986.

[13] 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,* A. Pirotte, C. Delobel, and G. Gottlob, eds., *Lecture Notes in Computer Science,* vol. 580, Springer-Verlag, Vienna, Austria, 1992, pp. 56-71.

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

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

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

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

[24] 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.

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

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

[27] L. Talmy, “How Language Structures Space,” *Spatial Orientation: Theory, Research, and Application,* H. Pick and L. Acredolo, eds., vol. Plenum Press, New York, NY, 1983, pp. 225-282.