VLDB Journal, 3, 479-516 (1994), Ralf Hartmut Güting, Editor

Qualitative Representation of Spatial Knowledge in Two-Dimensional Space

Dimitris Papadias and Timos Sellis

Received July 5, 1993; revised version received April 6, 1994; accepted May 20, 1994.

Abstract. Various relation-based systems, concerned with the qualitative representation and processing of spatial knowledge, have been developed in numerous application domains. In this article, we identify the common concepts underlying qualitative spatial knowledge representation, we compare the representational properties of the different systems, and we outline the computational tasks involved in relation-based spatial information processing. We also describe *symbolic spatial indexes,* relation-based structures that combine several ideas in spatial knowledge representation. A symbolic spatial index is an array that preserves only a set of spatial relations among distinct objects in an image, called the modeling space; the index array discards information, such as shape and size of objects, and irrelevant spatial relations. The construction of a symbolic spatial index from an input image can be thought of as a transformation that keeps only a set of representative points needed to define the relations of the modeling space. By keeping the relative arrangements of the representative points in symbolic spatial indexes and discarding all other points, we maintain enough information to answer queries regarding the spatial relations of the modeling space without the need to access the initial image or an object database. Symbolic spatial indexes can be used to solve problems involving route planning, composition of spatial relations, and update operations.

Key Words. Spatial data models, spatial query languages, representation of direction and topological relations, qualitative spatial information processing.

1. Introduction

The term *spatial knowledge* refers to configurations among distinct spatial entities (i.e., spatial representations preserve location in space without incorporating information such as shape, size, texture, or color of objects; Glasgow and Papadias, 1992). As an

Dimitris Papadias, M.Sc., is a Ph.D. candidate, and Timos Sellis, Ph.D., is Associate Professor, Computer Science Division, Department of Electrical and Computer Engineering, National Technical University of Athens, Greece 15780.

example of the use of spatial knowledge representations in everyday life, consider subway maps (e.g., the London Underground map). These maps do not display the shapes and sizes of the stations or quantitative distances between stations, but they contain line and point data and preserve spatial relations (e.g., one station is north of another station) and intersects (one line intersects another line). In particular, this article concentrates on qualitative spatial representations. Qualitative knowledge representation does not require an intermediate domain in which scale is defined, but comparisons are performed directly in the represented domain; in the domain of spatial knowledge, object locations are compared through spatial relations (Freksa, 1992). Spatial relations have been classified (Pullar and Egenhofer, 1988) into several types, including *direction relations* that describe order in space (e.g., north, north_east), *topological relations* that describe neighborhood and incidence (e.g., inside, overlap), *ordinal relations* that describe inclusion (ordinal relations are a subset of topological relations), and *distance relations* (e.g., near, far). These types of spatial relations have been studied independently or in association with each other. Egenhofer and Herring (1990), for instance, provided a mathematical framework for the definition of topological relations, while Papadias and Sellis (1993) defined direction relations using representative points. Kainz et al. (1993) modeled ordinal relations using partially ordered sets, and Frank (1992) proposed a method for qualitative reasoning that combines direction with distance relations.

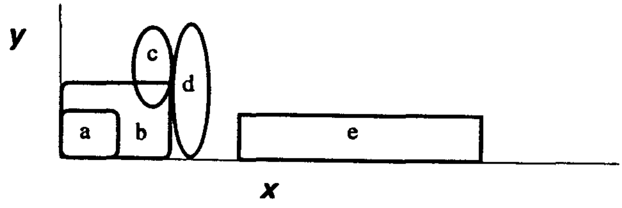
There are several reasons for following a qualitative approach to spatial knowledge representation:

1. the precision of quantitative representations is not always desirable
2. the input and the output of spatial processes is often qualitative rather than quantitative
3. qualitative knowledge is usually cheaper.

In Computer Vision research, semantic networks and graph representations have been used to represent spatial relations among image components. Levine (1978) developed a semantic network where the nodes denote objects and the arcs encode spatial relations such as left, above, or behind. In Artificial Intelligence several representational formalisms, usually based on logic (e.g., Randell et al., 1992), have been developed to represent and reason with spatial relations. The qualitative representation and processing of spatial knowledge has been proposed for Image Databases and Geographic Information Systems (Sistla et al., 1994; Papadias and Sellis, 1992). Other possible applications include Route Planning (Holmes and Jungert, 1992), Image Similarity Retrieval (Lee et al., 1992), and Spatial Pattern Matching—matching that depends on the spatial relations among distinct objects, and not on geometric properties (Glasgow et al., 1992).

We use the term *relation-based representations* for qualitative representational systems that deal with spatial relations, but exclude object characteristics or quantitative metric information. In this article, we look at existing work on qualitative spatial knowledge representation and we describe symbolic spatial indexes, a new

Figure 1. Original image



relation-based representation. Section 2 outlines several systems concerned with the representation of spatial relations, and links the various approaches under one framework of study. Section 3 introduces symbolic spatial indexes, and Section 4 describes how they can be used to capture direction relations in different levels of resolution. Section 5 enhances the expressive power of symbolic spatial indexes by incorporating topological relations. Section 6 is concerned with qualitative information processing using symbolic spatial indexes, and Section 7 concludes with comments and a discussion about future work.

2. Overview of Qualitative Spatial Knowledge Representation

Several relation-based systems have been proposed to represent spatial relations in various scientific areas. Depending on the particular viewpoint, the goals have been:

- explanatory and predictive power, in the case of computational models of Vision and Imagery
- expressive power and inferential adequacy, in the case of Artificial Intelligence representation schemes
- efficient manipulation of large amounts of geographic and geometric data, in the case of Spatial Databases.

In the following section, we identify the concepts underlying qualitative spatial knowledge representation, we survey previous relation-based systems, and we link the different perspectives under one framework.

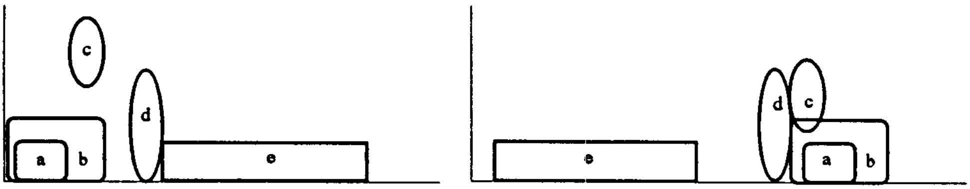
2.1 Properties of Relation-Based Representations

The image representations that we use are 2-D projections of 3-D objects (Figure 1). Each object in the image occupies a set of pixels and has an interior, a boundary, and a complement with respect to the embedding space. Furthermore, we assume a pair of viewer-independent orthogonal axes x and y.

The image conveys information about object characteristics (e.g., shape, size, and color), as well as spatial knowledge about the location of objects in space. Relation-based representations of the image discard object characteristics and irrelevant

Figure 2. Images equivalent to the original image

a. With respect to direction relations **b. With respect to topological relations**



spatial relations, and preserve only a set of spatial relations, which is called the modeling space M. Buisson (1989) argues that the spaces of interest in spatial reasoning are *topological spaces* which include only concepts of connectedness and continuity, *vector spaces* which deal with vectorial dimensions and directions, *metric spaces* which deal with the concept of distance, and *Euclidean spaces* which admit notions of scalar products, orthogonality, angle, and norm. In this article, we deal with the relation-based representation of modeling spaces consisting of binary direction and topological relations among rigid objects.

Let S^M be the relation-based representation that preserves the spatial relations of M that exist in an image s. Each spatial relation $r \in M$ between two object representations p and q in s is mapped onto a relation R between two symbolic object representations P and Q in S^M. In the rest of the article, we use small letters to denote objects in the original images and capital letters to denote symbolic object representations in relation-based structures. We adopt the notation $s \Vdash^M r$ (p, q) to denote that image s implies the spatial relation $r(r \in M)$ between objects p and q. Two images s and t are said to be equivalent with respect to a modeling space $M(s \equiv^M t)$ iff they imply the same subset of M for every pair of objects, that is: $s \Vdash^M r(p, q) \Leftrightarrow t \Vdash^M r(p, q)$. The image in Figure 2*a* is equivalent to the image of Figure 1 with respect to direction relations in 2-D space (e.g., north or east), while the image in Figure 2*b* is equivalent to the image of Figure 1 with respect to topological relations (e.g., meet, disjoint, overlap). Image equivalence depends on the resolution of the assumed modeling space (Section 4). Depending on their definitions, direction and topological relations may be related; topological properties, for instance, may be inferred from direction relations.

According to Hernàndez (1993), the relation-based representation of spatial knowledge avoids the falsifying effects of exact geometric representations by not committing to all aspects of the situation being presented (Hernàndez uses the term *relative representation based on comparative relations*); in this sense a relation-based representation is underdetermined since it may correspond to many situations. Using our terminology, we can say that each relation-based representation does not correspond to a single image, but to a class of equivalent images with respect to a modeling space; that is, relation-based representation systems are *ambiguous*.[1] This feature is of great importance in cases such as scene matching where equivalent scenes should have the same representations. If we did not use relation-based representations, then we would have to search for a transformation chain that would transform one scene into the other to determine equivalence between two visual scenes.[2] The combinatorial search for transformation chains connecting the two scenes can be avoided by computing the corresponding relation-based representations and comparing them for identity. Notice that to have this feature, the representation system must be *unique*; only one output relation-based representation should be generated from an input image representation. In this case, we have:

$$s \equiv^M \Leftrightarrow S^M = T^M.$$

We do not claim that relation-based representation systems are adequate for all applications involving spatial knowledge. For instance, they can not be used in problems that involve visualization or quantitative reasoning. Nevertheless, there are several potential application domains where relation-based representations can be used independently (e.g., qualitative reasoning) or in conjunction with other representation systems (e.g., spatial databases). In the rest of this section, we describe various representational systems that have been developed for different computational tasks, and we provide a framework to facilitate the study of qualitative spatial knowledge representation.

2.2 Previous Relation-Based Systems

Most of the work in Artificial Intelligence concerned with qualitative spatial reasoning has focused on logic-based representations. Such representational systems permit the description of real-world knowledge in predicates and rules of inference. Randell et al. (1992) developed a theory for topological reasoning in 2-D space expressed in a many-sorted logic. Sistla et al. (1994) proposed a set of rules for inference of direction and ordinal relations in 3-D images, and proved soundness and completeness. A variety of approaches to qualitative spatial reasoning has been based on Allen's (1983) temporal reasoning approach; extensions of Allen's interval algebra to higher-dimension spaces can be found in Güsgen (1989) and Mukerjee and Joe (1990). Related research has been carried out in the area of *spatial constraint networks.* A spatial constraint network is a graph-based description of a scene, where the nodes represent objects, and the arcs correspond to disjunctions of possible spatial relations between them. Inserting a new relation between two objects in the network

1. An extensive discussion about *ambiguity* and *uniqueness* of representations for rigid objects can be found in Requicha (1980).

2. To determine equivalence between two images, or spatial entities in general, we assume that there is a pre-defined modeling space of binary relations that are the only relations of interest between any pair of objects.

Figure 3. Intersection matrices

$$In = \begin{vmatrix} \partial P \cap \partial Q & \partial P \cap Q^{o} \\ P^{o} \cap \partial Q & P^{o} \cap Q^{o} \end{vmatrix}$$

$$I(A,B)=\begin{vmatrix} \neg\varnothing & \neg\varnothing \\ \varnothing & \neg\varnothing \end{vmatrix} \quad I(B,C)=\begin{vmatrix} \neg\varnothing & \neg\varnothing \\ \neg\varnothing & \neg\varnothing \end{vmatrix} \quad I(B,D)=\begin{vmatrix} \neg\varnothing & \varnothing \\ \varnothing & \varnothing \end{vmatrix} \quad I(B,E)=\begin{vmatrix} \varnothing & \varnothing \\ \varnothing & \varnothing \end{vmatrix}$$

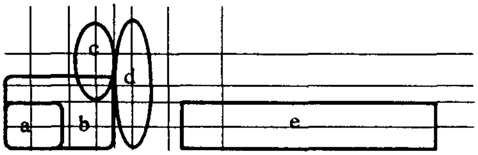
affects not only the two objects, but the insertion might yield additional constraints between other objects (*constraint propagation*). Studies of constraint propagation and consistency checking in networks of topological relations can be found in Smith and Park (1992) and Egenhofer and Sharma (1993). Hernàndez (1993) studied constraint networks of direction and topological relations.

Egenhofer and Herring (1990) developed a system that deals with a modeling space, consisting of the binary topological relations *disjoint, meet, equal, overlap, contains* (and its converse relation *inside*), and *covers* (and its converse *covered_by*). In their notation, each object P is represented in 2-D space as a point set which has an interior (P^{O}) and a boundary (∂P). The topological relation between any two objects (point sets) P and Q is described by the four intersections of the boundary and interior of P with the boundary and the interior of Q. Egenhofer (1991) extended the system by introducing the nine-intersection matrices that also include objects' exteriors. The matrix *In* (Figure 3), represents the four intersections between the two point sets P and Q. For instance, if the intersection of the boundaries of P and Q is non-empty, the element $In(1,1)$ is $\sim\oslash$, otherwise it is $\oslash$. The following four matrices of Figure 3 show how the formalism represents spatial knowledge about Figure 1. The information preserved in the matrices is *a covered_by b, b overlaps c, b meets d,* and *b disjoint e.* Notice that all equivalent images with respect to the assumed modeling space (Figures 1 and 2*b*) generate the same set of matrices. The formalism uses an inference mechanism to infer the spatial relation between two objects when their spatial relation with a third object is known (the composition relation). From *a covered_by b* and *b disjoint e,* it can be inferred that *a disjoint e.* (If multiple relations can be inferred, a disjunction of the possible relations is generated.)

Chang et al. (1987) developed the *2-D string* representation for encoding *symbolic images* (i.e., images where distinct objects are denoted by different symbols). A 2-D string is a pair of 1-D strings (u, v) where u represents the symbolic projections of the objects on the x axis, and v represents the projections on the y axis. The modeling space for 2-D strings includes direction relations in 2-D space. Although topological

Figure 4. Construction of a 2D-G string

a. Differences in object projections **b. Corresponding 2D-G string**



u: AB | B | BC | BCD | D | es | E
v: ABDE | ABCDE | BCD | CD

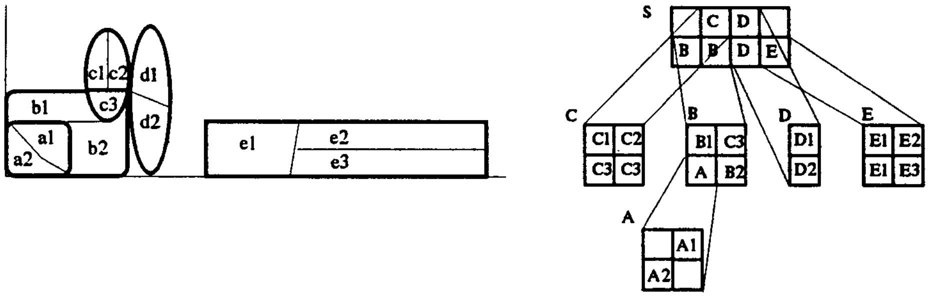
relations are not explicitly represented, topological information can sometimes be extracted from 2-D strings. Several variations, such as the *2D-H strings* and the *2D-G strings* (Chang et al., 1989), have been introduced to extend the expressive power of the original 2-D strings. Figure 4 illustrates the construction of a 2D-G string through a *cutting function* that detects and records differences in object projections on the x and y axes. Figure 4*a* illustrates the instances where the construction process has detected a change in object projections. On the x axes, for example, there are projections of objects a and b in the beginning; after the end of object a there is only object b, then object c starts, and so on. Figure 4*b* contains the strings u and v, which represent the order of object projections (*es* denotes empty space).

A hierarchical variation of symbolic images, called *symbolic arrays,* was used as the spatial working memory representation in the knowledge representation scheme for computational imagery (Papadias and Glasgow, 1991). Each array represents spatial relations (symbolic arrays have been used primarily for direction relations) among the distinct parts of a complex spatial entity. A part may be decomposed in an array of simpler parts at the immediately lower level (aggregation hierarchies). For instance, if we assume that the regions of Figure 1 consist of sub-regions, as illustrated in Figure 5*a*, then we can use the array structure of Figure 5*b* to represent the ordinal relations between regions and sub-regions, and to represent the direction relations between the sub-regions of each region (e.g., A is in region B, while it is south of sub-region $B1$). Direction relations between parts that exist in different regions (e.g., the relation west between $A1$ and $E1$) are not explicitly represented but they can sometimes be retrieved using appropriate inference mechanisms; Papadias et al., 1994*a*). Symbolic arrays were implemented in a functional language, called NIAL, based on a formal theory of nested arrays. Several functions that operate on symbolic arrays have been developed. These functions can be used to create symbolic arrays from other representations that store spatial knowledge (e.g., a frame database of complex objects), to modify symbolic arrays (e.g., rotate an array or move an object within an array) or to extract information found in the array (Glasgow and Papadias, 1992).

Depending on the modeling space to be preserved for a specific application domain, several of the previous systems could be used. Although the different systems can represent the same information about a given domain (they can be

Figure 5. Symbolic array example

a. Sub-regions of the original objects b. Corresponding symbolic array



made *informationally equivalent*[3]), they are not computationally equivalent because the efficiency of the retrieval mechanisms is not the same. Identical tasks may involve different algorithmic solutions and consequently have different complexities in distinct representational systems. Consider the goal of finding the objects that exist west of a given object in different systems:

1. In a representation system based on first order logic, this goal involves considering stored predicates plus recursive calls to rules of the form: $west(P, Q) \wedge west(Q, R) \Rightarrow west(P, R)$.
2. In a 2-D string representational system, the processing goal becomes a problem of string subsequence matching. In a symbolic array, the problem becomes one of searching array elements.

Furthermore, there is a basic difference between the inference mechanisms involved in logic and in the other spatial representation schemes. Inference in logic involves a proof procedure (e.g., resolution) built into the concept of logic. However, there can be spatial knowledge representation systems that make inferences without the use of explicit rules of deduction, and with a constraint satisfaction mechanism built into the processes that construct and access them. Lindsay (1988) used the following example: consider a simple case consisting of a discrete grid, each cell of which could be occupied by a single object labeled by a name. Now consider the case where object *b* is one grid point to the right of *a, c* is directly above *b,* and *d* is one grid point to the left of *c.* From this information we may conclude that *d* is directly above *a.* This inference could be supported by a logic-based system with appropriate

3. A discussion about *informational* and *computational equivalence* of spatial representations can be found in Larkin and Simon (1987).

rules of deduction; such a system would be *deductive*. Alternatively, this inference could be supported by a system of construction and retrieval processes that placed object names on the described grid and read off relations by scanning the grid; such a system would be *non-deductive*. The non-deductive system requires no separate computational inference-making stage; the operation of the construction process entails the making of inferences, and is similar to the well known techniques of query modification and materialization of views in relational database management systems (Ullman, 1988).

2.3 Qualitative Spatial Knowledge Representation

The representational systems of the previous subsection are relation-based because they use spatial relations among symbolically represented objects, rather than absolute coordinates. We can increase the amount of spatial knowledge represented by increasing the size and complexity of the representations. In any case, the preserved spatial knowledge will be a subset of the knowledge found in the initial image (as in Figure 1) in the sense that spatial relations can also be retrieved from the image using appropriate retrieval processes that operate on pixels. What we gain by using relation-based representations is a reduction of storage size, and an increase of computational efficiency in spatial knowledge retrieval since irrelevant information is discarded. The extraction of spatial relations from relation-based structures involves symbolic, and not numerical, computation and avoids the usual problems of geometric representations such as finite resolution and geometric consistency (although some of these problems may arise during the construction of relation-based structures from input images).

Let M be the modeling space, s be the set of input image representations, and S be the set of relation-based representation structures; then a spatial relation-based representation system can be defined as a function: $(s,M) \longrightarrow S$. In procedural terms, the function that maps an input image representation s ($s \in s$) to an output relation-based structure S^M ($S^M \in S$), which preserves the spatial relations of M, is achieved through a construction process that :

1. scans the input image, detecting the relation r ($r \in M$) between each pair of pixel object representations p and q,
2. maps r onto a relation R between the corresponding symbolic object representations in the output relation-based structure.

The construction process can be defined procedurally as:

```
Construction Process (s,M)
create S^M
for each object p in s do
for each object q in s do
    for each relation   r (r ∈ M)do
     if s ||-^M r(p, q) then incorporate R(P,Q) in S^M
    endfor
```

```
endfor
endfor
return (S^M)
```

In the case of logic-based representations, the previous construction process adds a predicate of the form $R(P,Q)$ to the list of predicates representing the image s when the spatial relation r between objects p and q is detected in s. For Egenhofer's system, the construction process adds an intersection matrix representing the topological relation between P and Q. In the cases of 2-D strings and symbolic arrays, however, information is not just added; the position of the new item of information in the output representation is determined by a constraint satisfaction mechanism built into the construction process. Thus, according to Lindsay's (1988) definitions, 2-D strings and symbolic arrays are non-deductive representations, while logic-based representations and intersection matrices are deductive.

The systems of Section 2.2 follow different approaches to the specification of construction processes. Chang et al. (1989) used *symbolic projections* to generate 2-D strings from raster input images. In Figure 4, we illustrated a projection-based method called the cutting function, which creates 2D-G string representations from 2-D input images. For symbolic arrays, the construction process is domain dependent; in Molecular Scene Analysis applications (Glasgow et al., 1992), a construction process creates symbolic arrays representing proteins from crystallographic data. On the other hand, logic-based systems do not usually include a construction process as a part of the system, but they take an initial set of relation predicates describing aspects of an image as given and perform some reasoning task (e.g., *find the deductive closure of the relations in the system*). The meaning of spatial predicates is encoded in the set of axioms that constitute the theory of logic-based systems.

We assume that the construction process can detect all the binary relations of the modeling space between distinct objects in the input image, and then map them onto relations among symbolic object representations, independently of the implementation. We use the notation $S^M \vdash R(P,Q)$ to denote that the relation R between object representations P and Q is retrieved through the representation S^M. A relation-based representation is complete iff whenever a relation $r (r \in M)$ between object representations p and q holds in an image s, then the corresponding relation R between P and Q can be retrieved through S^M, that is, $s \Vdash^M r(p,q) \Rightarrow S^M \vdash R(P,Q)$. According to the previous definition, the representational systems of Section 2.2, except for symbolic arrays, are complete, because no information is lost due to the structure of the relation-based representations.

Hierarchical spatial representations (e.g., symbolic arrays) are, in general, non-complete because we may lose information about the relations of the assumed modeling space. For instance, using the symbolic arrays of Figure 5*b* we cannot answer whether *A is east of C* or *A is west of C*. The hierarchical representation of space reduces storage requirements in applications where spatial knowledge is organized in hierarchies that correlate in certain ways. On the other hand, it results

Figure 6. A symbolic image

	C	D		
B	B	D		E
A	B	D		E

in information loss regarding the relations between objects that exist in different arrays although some of these relations can be retrieved. An extensive discussion about efficiency and information loss in geographic applications of symbolic arrays can be found in (Papadias et al., 1994*a*).

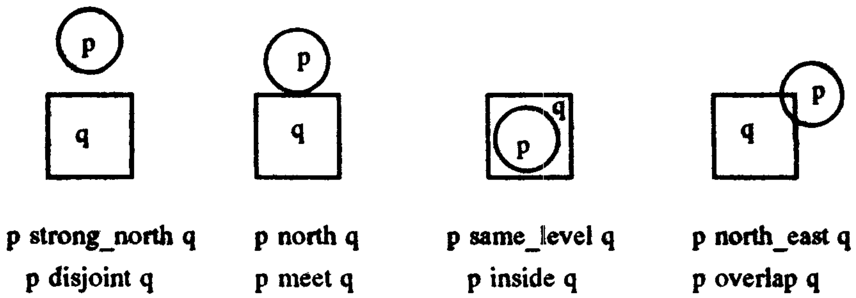
In addition to the representational properties of the system, the processing tasks also determine the choice of the representation system when various options are available. Graph-based representations have been used in pattern matching because graph matching is an area which has been extensively studied and several efficient algorithms have been developed. Logic-based representations are used in qualitative spatial reasoning because they provide a natural and flexible way to represent spatial knowledge, usually well understood semantics, and inference rules in terms of which proof procedures can be defined. The ordered structure of information and the compactness of non-deductive representations, such as 2-D string and symbolic arrays, facilitates the retrieval of spatial relations in applications involving large image databases and geographic information systems (GIS). In the rest of the article, we apply the concepts presented in this section to the development of another relation-based representation, called the *symbolic spatial index.*

3. Symbolic Spatial Indexes

Symbolic spatial indexes were motivated by previous work on symbolic images and arrays. Consider, for example, the symbolic image in Figure 6. Although the symbolic image obviously preserves some spatial relations about the image of Figure 1, several questions arise regarding the symbolic image, such as:

- Which spatial relations are preserved?
- Why was this and not another symbolic image generated from the input image?
- How can we incorporate more detailed direction and topological information in the symbolic image?
- How can we use symbolic images to infer information not explicitly stored (e.g., image overlay)?

Figure 7. Direction and topological relations



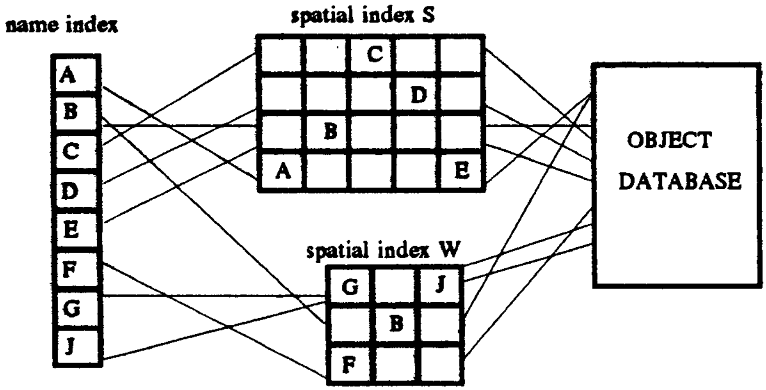
Symbolic spatial indexes were developed to answer the above questions. In the rest of this article, we discuss the representational properties of symbolic spatial indexes, and we show how they relate to the systems of the previous section. In addition, we demonstrate construction processes that can be used to construct indexes from input image representations, we describe alternative ways to represent direction and topological information at different levels of resolution, and we show how spatial indexes can be used in qualitative spatial information processing.

To generate a spatial index array from an input image, we assume a construction process that detects a set of special points in the image, called *representative points*. Every spatial relation in the modeling space can be defined using only the representative points. There are two kinds of representative points in the context of this article: *direction representative points,* which are used to define direction relations, and *topological* (*intersection*) *representative points,* which are used to define topological relations. Figure 7 illustrates several spatial relations whose formal definitions using representative points are given later in this article. The first line denotes the direction relation between objects p and q, while the second line denotes the topological relation between the two objects.

Spatial indexes, like symbolic images, are arrays of cells that store symbolic object representations. The construction of a symbolic spatial index from an input image can be thought of as a *transformation that keeps only the representative points needed to define the relations of the modeling space* and discards all other points of the image. By filling the array cells with representative points, we maintain adequate expressive power to answer queries regarding the spatial relations of the modeling space, without the need to access the initial image or an object database.

We use subscripts to denote individual cells; S_{ij} denotes the cell at row i and column j. S_{ij} and S_{kl} refer to the same cell iff $i=k$ and $j=l$ and S_{11} is the lower left cell of the array. Each cell of an index can be empty or it can be occupied by one (or more) symbol(s) denoting one (or more) representative point(s). We start with indexes that represent each object, using one representative point. We gradually

Figure 8. Possible implementation of symbolic spatial indexes



increase the number of points (symbols) per object. Direction representative points are indicated by a capital letter corresponding to the object to which the point belongs, and a subscript that corresponds to the points function.[4] For example, P_c denotes the center of object p. PQ denotes a topological representative point that belongs to the intersection of objects p and q. The predicate $S(sym,i,j)$ denotes that the cell S_{ij} contains the symbol sym; $S(P_c,i,j)$ denotes that S_{ij} contains P_c, while $S(PQ,i,j)$ denotes that S_{ij} contains the symbol PQ.

Where conventional indexes are used in database systems to facilitate information retrieval using attribute values, symbolic spatial indexes can be used to facilitate retrieval using relations in space. Consider a database of cities, from which we would like answers to questions involving the relative positions of cities. In a spatial index-based implementation, each index corresponds to one map (Figure 8), and each city symbol exists in all the indexes that represent maps in which the city appears. With this scheme, we can efficiently retrieve the cities that satisfy spatial conditions in one or more maps.

Consider, for example, the query "is there a city with a population greater than 1 million *north_east* of city B in map S?" Using the name index we locate city symbol B in the spatial index S, and then select the symbols denoting cities *north_east* of B (cities C and D). Using the pointers from these symbols to the object database, we retrieve the additional information concerning the query (population of the cities). Furthermore, we can answer more complicated queries such as "retrieve all cities that in map S that exist *north_east* of some city that is *south_east* of city G in map

4. The subscripts are not necessary for the definition of spatial relations, and we sometimes omit them in the illustrations.

W." (Details about expressing such queries in symbolic spatial indexes are given in Section 6.) In the next two sections, we demonstrate how the construction process detects representative points and generates symbolic spatial indexes that preserve the spatial relations of the modeling space.

4. Representation of Direction Relations in Spatial Indexes

For the representation system of this section, we assume that the modeling spaces consist of direction relations such as *north, east, north_east, same_level,* etc. Notice that the meaning of these relations is not obvious. Most people will agree that England is north of Portugal, but what about the relation between Spain and Portugal? There are parts of Spain that are directly north of parts of Portugal, but is it correct to state that Spain is north of Portugal?[5] These concepts are directly applicable to geographic applications where the formalization of spatial relations is crucial for user interfaces and query optimisation strategies. In addition, the importance of direction relations has been pointed out by several researchers in areas including spatial data structures (Peuquet, 1986), spatial reasoning (Dutta, 1989), cognitive science (Jackendoff, 1983) and linguistics (Herskovits, 1986). In the rest of the section, we specify the meaning of the direction relations using representative points, and we show how to map direction relations onto relations among representative points stored in symbolic spatial indexes. We use two sets of definitions for direction relations: the first one defines the direction relation between two objects by using one representative point per object, while the second set of definitions uses two representative points per object.

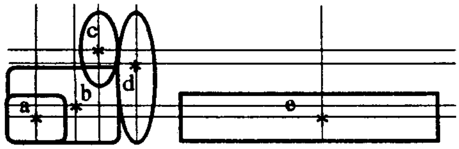
4.1 Representation of Direction Relations Using One Point per Object

Most of the work on direction relations has concentrated on point objects. According to this approach, each object is abstracted as one point which may be, for example, the center of mass or the center of symmetry. Consider *cartographic generalization* and *geometric abstraction* (Bruegger and Muller, 1992). At some high level of resolution (state level) cities are represented as regions, while at a lower level (country or continent level) cities are denoted by points. The symbol * in Figure 9*a* denotes the centers of the objects in the initial image, and the projection lines correspond to the instances at which the construction process has detected a representative point (a center). All the rows and columns that do not contain a representative point are deleted from the input image, and the result is moved to the spatial index $S^{M_{D1}}$ (Figure 9*b*). Using this construction process, the order of the representative points on the x and y axes is preserved in the output index but metric and topological

5. A survey and experimental study regarding the use of direction relations in cognitive spatial reasoning at geographic scales can be found in Mark (1992).

Figure 9. Generation of symbolic spatial index $S^{M_{D1}}$

a. Detection of representative points



b. Symbolic spatial index $S^{M_{D1}}$

		C_c		
			D_c	
	B_c			
A_c				E_c

information is lost.

M_{D1} denotes the set of primitive direction relations for point objects that we can define using the previous construction process. Primitive relations have the following properties:

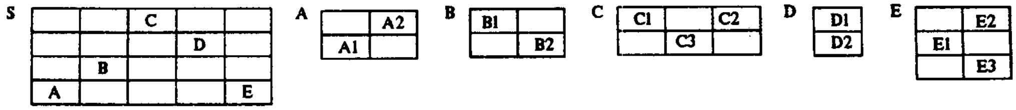
- they are mutually exclusive,
- they provide a complete coverage,
- they correspond to the highest resolution given a set of representative points.

A discussion of primitive direction relations can be found in Section 4.3. We define the mapping function that maps direction relations among objects in the initial image to relations among the object centers in $S^{M_{D1}}$ as follows:

$$
\begin{aligned}
&p \text{ north\_west } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i > k \wedge j < l\,]\\
&p \text{ restricted\_north } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i > k \wedge j = l\,]\\
&p \text{ north\_east } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i > k \wedge j > l\,]\\
&p \text{ restricted\_west } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i = k \wedge j < l\,]\\
&p \text{ same\_position } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i = k \wedge j = l\,]\\
&p \text{ restricted\_east } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i = k \wedge j > l\,]\\
&p \text{ south\_west } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i < k \wedge j < l\,]\\
&p \text{ restricted\_south } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i < k \wedge j = l\,]\\
&p \text{ south\_east } q &&\equiv \exists\, i,j,k,l\ [S(P,i,j) \wedge S(Q,k,l) \wedge i < k \wedge j > l\,]
\end{aligned}
$$

Exactly one of the previous relations holds true between any pair of point objects. The primitive relations are transitive and *same_position* is also symmetric. The rest form four pairs of converse relations (e.g., *p north_east q* $\Leftrightarrow$ *q south_west p*). Additional direction relations can be defined using disjunctions of the primitive relations. For instance, we can define the relation: *p same_level q* $\equiv$ *p restricted_east q* $\vee$ *p restricted_west q* $\vee$ *p same_position q*.

Figure 10. Representation of ordinal relations



Like symbolic arrays, spatial indexes that preserve direction relations can capture aggregation hierarchies. In contrast to symbolic arrays, where the concept of inclusion is embedded in the implementation language, symbolic spatial indexes can represent ordinal relations by permitting objects that appear as parts of regions in an index to be names of other indexes. For instance, the arrays of Figure 10 could be used to represent the direction and inclusion relations of Figure 5*a*. In such applications, spatial indexes, in addition to direction relations, preserve the relation *in*: p in q $\equiv$ *i,j Q(P,i,j)* and its converse relation *contains*: p contains $q \equiv q$ in p. Object p is *in* q, if symbol P exists in the array Q representing object q.

When spatial relations are defined using representative points stored in arrays, information retrieval becomes a straightforward procedure that searches parts of the arrays for representative points. Modeling space M_{D1} is adequate for point abstractions of objects, but we want to answer queries of the form "are there parts of c which are *restricted_south* of some parts of b?" The "centers approach" is not sufficient for such queries, and we need abstractions that preserve two or more representative points for each object.

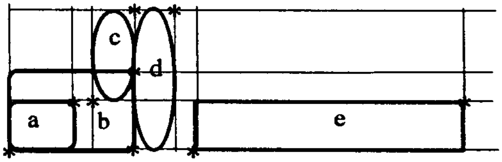
4.2 Representation of Direction Relations Using Two Points per Object

Depending on the application domain, there are several options for choosing multiple points to define direction relations. In this section, we describe a second set of definitions, which uses the lower left point (P_f) and the upper right point (P_s) of the minimum bounding rectangle that covers p. Figure 11*a* contains the edge points of the bounding rectangles, and Figure 11*b* illustrates the corresponding spatial index $S^{M_{D2}}$.

Although the image of Figure 2*a* will generate the spatial index of image 9*b* using the centers as representative points, it will not generate the index of Figure 11*b* using two representative points per object. It can been seen that *image equivalence depends on the choice of representative points.* When we use two points instead of one we can represent more detailed spatial knowledge. For instance, we can define several refinements of the *north* relation:

Figure 11. Generation of symbolic spatial index $S^{M_{D2}}$

a. Detection of representative points **b. Symbolic spatial index $S^{M_{D2}}$**



			C_s	D_s		
			B_s			
	A_s	C_f				E_s
B_f,A_f			D_f		E_f	

$$\begin{aligned}
p \text{ strong_north } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge \\
& \forall\, i,j,k,l\ [(S\ (P,i,j) \wedge S\ (Q,k,l)) \Rightarrow i > k]
\end{aligned}$$

$$\begin{aligned}
p \text{ strong_bounded_north } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge \\
& \forall\, i,j\ \forall\, k,l\ [(S\ (P,i,j) \wedge S\ (Q,k,l)) \Rightarrow i > k] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge j > l)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge j < l)]
\end{aligned}$$

$$\begin{aligned}
p \text{ strong_northeast } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge \\
& \forall\, i,j\, \forall\, k,l\ [(S\ (P,i,j) \wedge S\ (Q,k,l)) \Rightarrow i > k \wedge j > l]
\end{aligned}$$

$$\begin{aligned}
p \text{ just_north } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge i = k] \wedge \\
& \forall\, k,l\ [S\ (Q,k,l) \Rightarrow \exists\, i,j\ (S\ (P,i,j) \wedge i > k)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge i > k)] \wedge
\end{aligned}$$

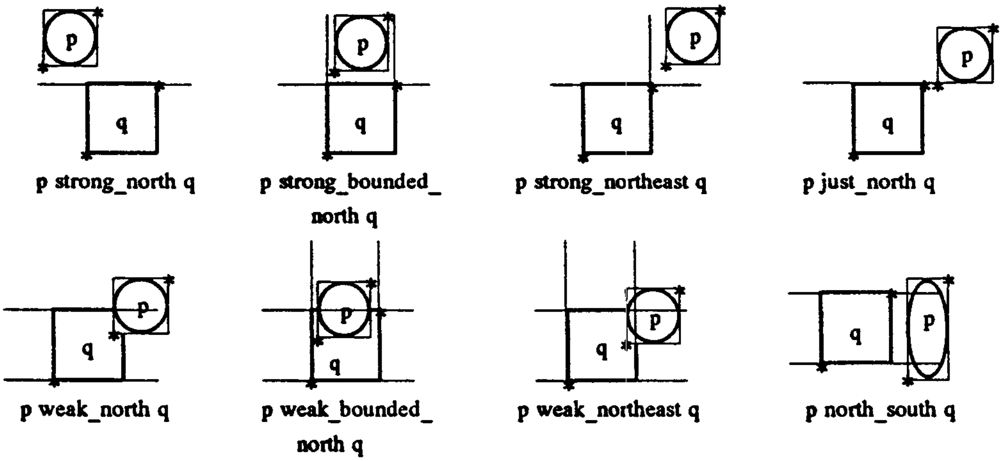
$$\begin{aligned}
p \text{ weak_north } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge i < k] \wedge \\
& \forall\, k,l\ [S\ (Q,k,l) \Rightarrow \exists\, i,j\ (S\ (P,i,j) \wedge i > k)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge i > k)]
\end{aligned}$$

$$\begin{aligned}
p \text{ weak_bounded_north } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge i < k] \wedge \\
& \forall\, k,l\ [S\ (Q,k,l) \Rightarrow \exists\, i,j\ (S\ (P,i,j) \wedge i > k)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge i > k)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge j > l)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge i < l]
\end{aligned}$$

$$\begin{aligned}
p \text{ weak_northeast } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [(S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge i < k] \wedge \\
& \forall\, k,l\ [S\ (Q,k,l) \Rightarrow \exists\, i,j\ (S\ (P,i,j) \wedge i > k \wedge j > l)] \wedge \\
& \forall\, i,j\ [S\ (P,i,j) \Rightarrow \exists\, k,l\ (S\ (Q,k,l) \wedge i > k \wedge j > l)]
\end{aligned}$$

$$\begin{aligned}
p \text{ north_south } q \equiv\ & \exists\, i,j\, \exists\, k,l\ [S\ (P,i,j) \wedge S\ (Q,k,l)] \wedge \\
& \forall\, k,l\ [S\ (Q,k,l) \Rightarrow \exists\, i,j\ (S\ (P,i,j) \wedge i > k)] \wedge \\
& \forall\, k,l\ [S\ (Q,k,l) \Rightarrow \exists\, i,j\ (S\ (P,i,j) \wedge i < k)]
\end{aligned}$$

Figure 12. Refinements of the north relation



These relations are illustrated in Figure 12. An implementation using R-trees can be found in Papadias et al. (1994*b*). According to the previous definitions, Spain is *north_south* of Portugal while England is *strong_north* of Portugal. Similar refinements may be necessary for many applications, and cannot be achieved if we use one point per object. Notice that the previous relations do not provide a complete coverage and they are not necessarily mutually exclusive (a discussion about the primitive relations of M_{D2} can be found in the following subsection). Although few of the possible primitive relations of M_{D2} are given specific names, they are preserved in $S^{M_{D2}}$ and they can be retrieved by using a proper query language.

By projecting the contents of a symbolic spatial index on the x and y axes we can generate pairs of 1-D encodings. Figure 13*a* illustrates the strings x_s and y_s, representing array $S^{M_{D2}}$; x_s contains the symbolic projections on the x axis and y_s on the y axis. If we remove the subscripts from the object symbols and eliminate the identical substrings, we will generate the strings of Figure 13*b*. Although these strings resemble the 2D-G string of Figure 4*b*, they are not identical, due to the differences of the construction process that create the symbolic spatial index, and the cutting function that creates the 2D-G strings. The correspondence of spatial indexes and 2-D strings depends on the choice of representative points.

If m is the number of objects, and k is the number of points per object, then the maximum size of the index array is $(km)^2$, when there is exactly one symbol in each row and each column. On the other hand, the size of each of the 1-D encodings that can be used to represent the array is km; thus 1-D representations are more efficient for information storage. Chang and his colleagues (1987) developed algorithms for the generation of 2-D strings from symbolic images, and for the reconstruction of images from their 2-D string representations. Similar algorithms can be applied to produce 1-D encodings of symbolic spatial indexes and vice-versa.

Figure 13. One-dimensional encodings of $S^{M_{D2}}$

a. Strings representing $S^{M_{D2}}$	b. Elimination of subscripts
x_s: $A_fB_f \mid A_s \mid C_f \mid B_sC_sD_f \mid D_s \mid E_f \mid E_s$	AB | A | C | BCD | D | E
y_s: $A_fB_fD_fE_f \mid A_sC_fE_s \mid B_s \mid C_sD_s$	ABDE | ACE | B | CD

4.3 Representation of Direction Relations Using Representative Points

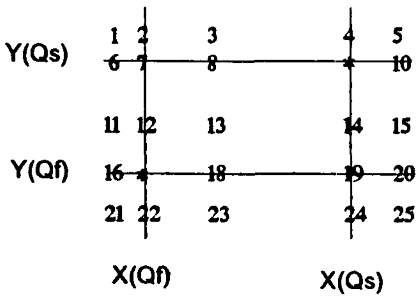
The term *primary object* denotes the object to be located, and the term *reference object* refers to the object in relation to which the primary object is located. When one point is used for the representation of the reference object, the plane is divided into nine partitions. The symbol * in Figure 14*a* denotes the representative point of the reference object, and X and Y are functions that return the x and y coordinates of a point. The numbers correspond to the possible positions of the representative point of the primary object with respect to the reference object (i.e., the primitive relations of M_{D1}; Figure 14*b*). Figure 14*c* illustrates how direction relations among points on the plane are mapped onto relations among point symbols in a symbolic spatial index. The symbol Q_c in Figure 14*c* denotes the reference point symbol, and the numbers refer to the direction relation, depending on the position of the primary point symbol in the index array. The relation between two point symbols does not change if we add or remove either from the index array lines, or from columns that do not contain the point symbols.

When we use two points to represent the reference object, the plane is divided into 25 partitions; thus there are 25 possible ways to place a point of the primary object in the plane. The symbol * in Figure 15*a* denotes the direction representative points of the reference object, and the numbers correspond to the partitions of the plane. In general, if k is the number of points used to represent the reference object, then the plane is divided into $(2k+1)^2$ partitions; k^2 of the partitions are points, $2k(k+1)$ are open line segments and $(k+1)^2$ are open regions. The previous numbers refer to the case when the points do not have any common x or y coordinates.

For 1-D space, the number of partitions is $2k+1$; k of the partitions are points, and $k+1$ are line segments. When we have two points for the reference object, for instance, the number of partitions is 5. If the primary object P is also represented by two points P_f and P_s ordered on the x axis ($X(P_f) < X(P_s)$) then the number of primitive relations between the two objects in 1-D space is 13. These 13 relations correspond to the relations between time intervals introduced by Allen (1983), and applied to 1-D spatial reasoning by Freksa (1991) and Pullar and Egenhofer (1988). In 2-D space and for the case of region objects, the constraint for the first and the second points of the bounding rectangle is: $X(P_f) < X(P_s) \wedge Y(P_f) <$

Figure 14. Direction relations using one point per object

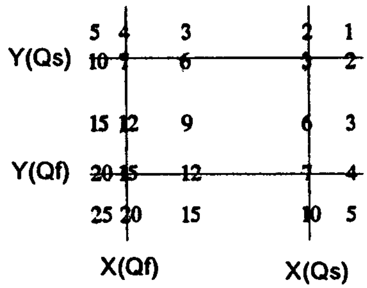
Figure 15. Plane partitions using two points



$Y(P_s)$, and the number of possible relations is 169 (the square of the number of relations in 1-D space). The 169 relations constitute the primitive relations of M_{D2} in the case of region objects, because they are mutually exclusive, they provide a complete coverage, and they correspond to the higher resolution using two points per object. An illustration of these relations for minimum bounding rectangles and a classification with respect to topological information can be found in Lee et al. (1992).

If we relax the constraint for the representative points of the primary object to: $X(P_f) \leq X(P_s) \wedge Y(P_f) \leq Y(P_s) \wedge \neg (X(P_f) = X(P_s) \wedge Y(P_f) = Y(P_s))$, that is, we also allow line primary objects (potentially parallel to the coordinate axes), thus the number of permitted relations is 221. Figure 16 illustrates how the placement of the first point in one partition constrains the possible partitions for the second. If, for instance, we place P_f in partition 4, then the only acceptable partitions for P_s are 4 and 5 (on the other hand, if p were a region, the only permitted position for P_s would be partition 5). The sum of all numbers is 221 (i.e., the set of primitive direction relations between a region reference object and a line primary object). If we relax the constraint to: $X(P_f) \leq X(P_s) \wedge Y(P_f) \leq Y(P_s)$, allowing $X(P_f) =$

Figure 16. Primitive relations between region and line object



$X(P_s) \wedge Y(P_f) = Y(P_s)$ (the primary object can be a region, a line, or a point), then the number of primitive relations is 225, because there are four more relations for the four point partitions. Similarly, we can calculate the number of primitive direction relations between any combination of region, line, and point objects.

The previous numbers refer to primitive relations. The relations *strong_north, weak_north, just_north, weak_northeast* and *north_south* are not primitive because they constrain each point to a number of possible partitions. For example, object p is *weak_north* of object q if P_f is in one of the partitions 11-15 and P_s is in the partitions 1-5 with respect to q. The rest of the relations are primitive because they constrain each point to exactly one partition; for instance, according to *weak_bounded_north,* P_f must be in partition 13, and P_s must be in partition 3 with respect to q. Relations of lower resolution can be defined by allowing points to be in various partitions (i.e., using disjunctions of more "restrictive" relations). For instance, we can define the relation *north* as: p *north* $q \equiv p$ *strong_north* $q \vee p$ *weak_north* $q \vee p$ *just_north* q. *Strong_north* and *weak_north* can be further decomposed to disjunctions of other relations until we reach primitive relations. In case of *strong_north,* the definition consists of 13 primitive relations (two of which are *strong_northeast* and *strong_bounded_north*). Out of the large number of possible direction relations, only a few may be needed for an application domain. For example, it is improbable that a geographic extension of SQL would include 169 different expressions for primitive directions between region objects. Although we have defined a few of these relations, a number of additional ones can be defined and used in practical applications.

Our work extends previous approaches to direction relations (Dutta, 1989; Freksa, 1992) by dealing with extended objects instead of point objects. Peuquet and Ci-Xiang (1987) designed an algorithm to determine direction relations between arbitrary polygons represented by minimum bounding rectangles. Their method involves only four direction relations (*north, east, west and south*) and is based on the cone-shaped concept of direction (i.e., direction relations are defined using angular regions between the reference and the destination object or primary object). Our method is based on the concept of projections. An extensive study of the two approaches for point objects can be found in Frank (1994).

5. Incorporation of Topological Relations in Spatial Indexes

In Figure 1, the actual objects intersect when there are intersecting minimum bounding rectangles. But this is not always true for any pair of objects (this is a well known fact in spatial data structures where there is a refinement step to retrieve the relation between the actual objects in the case of intersecting bounding rectangles; Papadias et al., 1994*b*). As a consequence, indexes that preserve direction relations are inadequate for several practical applications that also involve neighborhood, inclusion, overlap, or other topological queries. In this section, we show how to incorporate topological relations in symbolic spatial indexes containing direction information.

5.1 Representation of Topological Relations Using One Symbol per Object

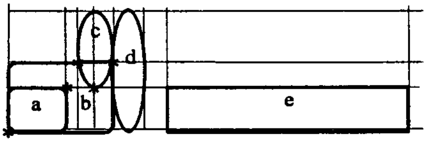
The notion of resolution in the representation of direction relations can be extended for topological relations; but instead of the number of points per object, it is the type of symbols per object that determines the number of represented relations. For instance, we can use different symbols to distinguish a point that belongs to the boundary from a point that belongs to the interior of an object. We start with the representation of topological relations using one symbol per object. According to this approach, the primitive relations that we represent are $p \cap q = \oslash$, which corresponds to the *disjoint* relation, and $p \cap q \neq \oslash$, which corresponds to all the other topological relations of Figure 7.

Unlike direction relations, where we can assume the existence of unique representative points ("centers" or "edge" points), uniqueness is not trivial for symbolic spatial indexes preserving topological relations. Multiple options arise in cases where two or more points satisfy the conditions to be (intersection) representative points for a topological relation between two objects. For instance, any point of object *a* can be chosen as a representative point for the intersection with object *b*. To achieve uniqueness when there are two or more symbolic spatial indexes corresponding to the same equivalence class, the construction process is responsible for creating the "correct" index by choosing one point in a deterministic way.

The construction process that creates the index $S^{M_{D2T1}}$, incorporating both direction and topological information about *s*, starts by detecting the direction representative points. The first intersection points that are incorporated in $S^{M_{D2T1}}$ are the ones that coincide with direction representative points. Intersection points *AB* coincide with points A_f and A_s, while intersection points *BC, BD,* and *CD* coincide with B_s. Such points do not increase the size of the output index. Then

Figure 17. Generation of symbolic spatial index $S^{M_{D2T1}}$

a. Detection of representative points **b. Symbolic spatial index $S^{M_{D2T1}}$**



				C_s	D_s		
		BC		B_s ,BC,BD,CD			
	A_s ,AB	C_f	BC				E_s
A_f ,B_f ,AB				D_f		E_f	

the process detects object intersections that exist on the sides of the bounding rectangles. We follow this approach because intersections on the sides of the rectangles generate at most one extra row or column in the output index, whereas an arbitrary intersection may produce an extra row and an extra column. Finally, the construction process finds intersections for the pairs of objects that do not intersect on a direction representative point or on their bounding rectangles.

In general, the process chooses the intersection points in a deterministic way so that uniqueness is preserved, and the output index remains as small as possible (for a detailed description of the algorithm see Papadias, 1994). Figure 17 illustrates the generation of symbolic spatial index $S^{M_{D2T1}}$. The symbol * in Figure 17*a* denotes the instances where the construction process has detected an intersection (the direction representative points are not included because they are illustrated in Figure 11*a*). Figure 17*b* illustrates the generated index $S^{M_{D2T1}}$ (the order of symbols in topological representative points is not important; *AB* is the same as *BA*).

Using $S^{M_{D2T1}}$, in addition to the direction relations of M_{D2}, we can define the topological relations:

p disjoint q $\equiv \neg\exists\ i,j\ S(PQ,i,j)$

p not_disjoint q $\equiv \exists\ i,j\ S(PQ,i,j)$

Notice that $S^{M_{D2T1}} \vdash R(P,Q)$ does not necessarily imply that $S^{M_{D2}} \vdash R(P,Q)$ or $S^{M_{T1}} \vdash R(P,Q)$. $S^{M_{D2T1}}$ cannot be represented using $S^{M_{D2}}$ and a set of intersection predicates, because it also contains the relative positions of intersections with respect to the direction representative points. This property allows the definition of relations that belong to M_{D2T1} but not to M_{T1} or M_{D2}. Such relations are:

p north_touches q $\equiv \exists\ i,j[S(PQ,i,j) \wedge \forall\ k,l(S(P,k,l) \Rightarrow k \geq i)]$

p east_touches q $\equiv \exists\ i,j[S(PQ,i,j) \wedge \forall\ k,l(S(P,k,l) \Rightarrow l \geq j)]$

p south_touches q $\equiv \exists\ i,j[S(PQ,i,j) \wedge \forall\ k,l(S(P,k,l) \Rightarrow i \geq k)]$

p west_touches q $\equiv \exists\ i,j[S(PQ,i,j) \wedge \forall\ k,l(S(P,k,l) \Rightarrow j \geq l)]$

Objects a and c *north_touch* object b in Figure 1 because there is an intersection point in the southmost part of their boundary.[6] The previous construction process records the relative positions of the intersections with respect to the bounding rectangles, and permits the definition of relations such as *north_touches*. If such relations are not needed for an application domain, then a simpler construction process that marks just one intersection for each pair of objects can be implemented.

Although we assumed two distinct modeling subspaces M_{D2} and M_{T1}, we do not argue that direction relations are independent of topological relations. In the case where objects are the same as their minimum bounding rectangles, the direction relation also conveys the topological relation between the objects. In the case of arbitrary objects, where none of the points of the primary object is in the bounding rectangle of the reference object, we also have the topological relation disjoint between the objects. Furthermore, it is not possible to have certain direction relations in conjunction with some topological relations among the same objects. For example, according to the previous definitions it is not possible to have *p strong_north q* and *p touch q* in the same array. A discussion about the topological information that bounding rectangles convey about the actual objects that they enclose can be found in Clementini et al. (1994).

5.2 Representation of Topological Relations Using Representative Points

As in the case of direction relations, the topological resolution of symbolic spatial indexes can be increased or decreased to match the representation and processing goals of a given application domain. If we make the distinction between the boundary of an object p, denoted by ∂p, and the interior denoted by p^o, then we can define the topological relation between two objects using the following intersections (intersection matrices in Figure 3):

$$\partial p \cap \partial q = \emptyset,\ \partial p \cap \partial q = \neg\emptyset$$
$$p^o \cap \partial q = \emptyset,\ p^o \cap \partial q = \neg\emptyset$$
$$\partial p \cap q^o = \emptyset,\ \partial p \cap q^o = \neg\emptyset$$
$$p^o \cap q^o = \emptyset,\ p^o \cap q^o = \neg\emptyset$$

The relations in each line are mutually exclusive; one is true, while the other is false for any pair of objects. By combining the intersections between boundaries and interiors, we can create 16 mutually exclusive topological relations between objects. However, not all of these relations are valid due to the constraints imposed by the properties of the object boundaries and interiors. For instance, for all pairs of objects the following constraints must always hold:

6. The relations that combine direction and topological information should not be confused with conjunctions of direction and topological relations (e.g., p north_touches $q \neq p$ north $q \wedge p$ not_disjoint q. Object a north_touches b but it is not the case that a north b).

$(p^o \cap q^o = \neg\oslash) \Rightarrow (\partial p \cap \partial q = \neg\oslash \vee \partial p \cap q^o = \neg\oslash \vee p^o \cap \partial q = \neg\oslash)$
$(p^o \cap q^o = \oslash \wedge \partial p \cap \partial q = \oslash) \Rightarrow (p^o \cap \partial q = \oslash \wedge \partial p \cap q^o = \oslash)$.

Of the 16 possible relations that can be defined using the previous intersections, only the following eight defined by Egenhofer and Herring (1990) are valid. These are the primitive relations[7] of M_{T2} (some of these relations are illustrated in Figure 7):

p disjoint $q \equiv (\partial p \cap \partial q = \oslash) \wedge (p^o \cap \partial q = \oslash) \wedge (\partial p \cap q^o = \oslash) \wedge (p^o \cap q^o = \oslash)$
p meet $q \equiv (\partial p \cap \partial q \neq \oslash) \wedge (p^o \cap \partial q = \oslash) \wedge (\partial p \cap q^o = \oslash) \wedge (p^o \cap q^o = \oslash)$
p equal $q \equiv (\partial p \cap \partial q \neq \oslash) \wedge (p^o \cap \partial q = \oslash) \wedge (\partial p \cap q^o = \oslash) \wedge (p^o \cap q^o \neq \oslash)$
p contains $q \equiv (\partial p \cap \partial q = \oslash) \wedge (p^o \cap \partial q \neq \oslash) \wedge (\partial p \cap q^o = \oslash) \wedge (p^o \cap q^o \neq \oslash)$
p inside $q \equiv (\partial p \cap \partial q = \oslash) \wedge (p^o \cap \partial q = \oslash) \wedge (\partial p \cap q^o \neq \oslash) \wedge (p^o \cap q^o \neq \oslash)$
p covers $q \equiv (\partial p \cap \partial q \neq \oslash) \wedge (p^o \cap \partial q \neq \oslash) \wedge (\partial p \cap q^o = \oslash) \wedge (p^o \cap q^o \neq \oslash)$
p covered_by $q \equiv (\partial p \cap \partial q \neq \oslash) \wedge (p^o \cap \partial q = \oslash) \wedge (\partial p \cap q^o \neq \oslash) \wedge$
$(p^o \cap q^o \neq \oslash)$
p overlap $q \equiv (\partial p \cap \partial q \neq \oslash) \wedge (p^o \cap \partial q \neq \oslash) \wedge (\partial p \cap q^o \neq \oslash) \wedge (p^o \cap q^o \neq \oslash)$

The construction process that generates a symbolic spatial index, preserving the relations of M_{T2}, must detect four kinds of intersections for each pair of objects; $\partial p \cap \partial q$, $p^o \cap \partial q$, $\partial p \cap q^o$ and $p^o \cap q^o$. For each non-empty pair, the corresponding symbol is marked (i.e., if $\partial p \cap \partial q$ is found non-empty, the symbol $\partial P \partial Q$ is marked in the output index). To transform the previous definitions of topological relations into definitions using representative point symbols stored in a symbolic spatial index, we have to replace each equation of the form: $p \cap q \neq \oslash$ with a formula of the form: $\exists\ i,j\ S(PQ,i,j)$ and each equation of the form $p \cap q = \oslash$ with a formula of the form $\neg\exists i,j\ S(PQ,i,j)$. Thus, the definitions of the topological relations between objects represented by two symbols per object (the boundary and the interior) are transformed to:

p disjoint $q \equiv \neg\exists\ i,j\ S(\partial P \partial Q,i,j) \wedge \neg\exists\ i,j\ S(P^o \partial Q,i,j) \wedge \neg\exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\neg\exists\ i,j\ S(P^o Q^o,i,j)$
p meet $q \equiv \exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \neg\exists\ i,j\ S(P^o \partial Q,i,j) \wedge \neg\exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\neg\exists\ i,j\ S(P^o Q^o,i,j)$
p equal $q \equiv \exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \neg\exists\ i,j\ S(P^o \partial Q,i,j) \wedge \neg\exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\exists\ i,j\ S(P^o Q^o,i,j)$
p contains $q \equiv \neg\exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \exists\ i,j\ S(P^o \partial Q,i,j) \wedge \neg\exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\exists\ i,j\ S(P^o Q^o,i,j)$
p inside $q \equiv \neg\exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \neg\exists\ i,j\ S(P^o \partial Q,i,j) \wedge \exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\exists\ i,j\ S(P^o Q^o,i,j)$

7. Randell et al. (1992) presented an interval logic for topological reasoning that includes the same set of primitive topological relations. Related work also can be found in Vieu (1993).

p covers $q \equiv \exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \exists\ i,j\ S(P^o \partial Q,i,j) \wedge \neg \exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\exists\ i,j\ S(P^o Q^o,i,j)$
p covered_by $q \equiv \exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \neg \exists\ i,j\ S(P^o \partial Q,i,j) \wedge \exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\exists\ i,j\ S(P^o Q^o,i,j)$
p overlaps $q \equiv \exists\ i,j[S(\partial P \partial Q,i,j)] \wedge \exists\ i,j\ S(P^o \partial Q,i,j) \wedge \exists\ i,j\ S(\partial P Q^o,i,j) \wedge$
$\exists\ i,j\ S(P^o Q^o,i,j)$

For an example of an array that preserves the previous topological relations using two symbols per object, see Papadias and Sellis (1993). Notice that, while the number of points per object needed for the direction relations is independent of the relations of the object with the other objects (one or two points for the first and second cases of the previous section), the number of points per object for the topological relations depends on the configuration. The maximum number of points needed to represent binary topological relations among m objects using the previous definitions is $4C(m,2) = 2m(m\text{-}1)$. When all the object pairs overlap, ($C(m,2)$ denotes the number of possible selections of 2 out of m objects. Because four intersections are needed for every pair of overlapping objects, the total number of points is four times the number of possible selections of two out of m objects.

For more refined topological information, additional symbols per object could be used. Egenhofer (1991) defined topological relations by using definitions that involve the interiors, the boundaries, and the exteriors of objects. If we want a further extension of the expressive power to include the dimensions of the intersections, we could, for example, add symbols to distinguish line from point data.

6. Qualitative Information Processing Using Spatial Indexes

While previous sections were concerned with knowledge representation issues, Section 6 describes how symbolic spatial indexes can facilitate qualitative information processing. In particular, we show how spatial indexes can deal with the problems of information retrieval, composition of spatial relations, route planning, and the problem of update operations. Since each of these problems itself requires extensive analysis, we deal with them briefly, without providing the reader with details in the algorithmic level.

6.1 Information Retrieval

Consider the task of finding all objects that exist *strong_north* of object B in index $S^{M_{D2T1}}$. Using a Pictorial SQL (Roussopoulos et al., 1988) we could express the query in the following form:

```
select P
from S^{M_D2T1}
where strong_north(P,B)
```

An SQL-like format, however, is restrictive because not all of the relations of the

Figure 18. Spatial indexes representing maps of cities

a. Index $S^{M_{D1}}$

		C		
			D	
	B			
A				E

b. Index $W^{M_{D1}}$

G		J
	B	
F		

modeling space are given specific names. In this section, we describe how to retrieve information stored in symbolic spatial indexes using a pictorial query-by-example (PQBE) language. In comparison to verbal (spatial) query languages, PQBE provides a more intuitive and easy to use interface because of its close correspondence with the structure of 2-D space. Unlike verbal languages, it does not presume knowledge of keywords for spatial relations on behalf of the user, nor familiarity with database languages; complex spatial conditions, however, are expressed in a straightforward manner. Similar to the original QBE, PQBE generalizes from the example given by the user to compute the answer to the query. In this case, instead of having skeleton tables showing the relation scheme, we have skeleton arrays corresponding to symbolic spatial indexes. Domain variables denoting objects and images are preceded by the character "_", while constants appear without qualification. The character D before a variable causes its value to be displayed. For the following examples, we use the index arrays $S^{M_{D1}}$ and $W^{M_{D1}}$ (illustrated in Figure 18 and in Figure 8). We assume that the indexes represent maps of cities.

The query in the skeleton array of Figure 19*a* retrieves all cities P (primary object variable) where P is *north_east* with respect to city B (reference object constant) in map S. The query results in a search of the subpart of the array illustrated in Figure 19*b*. The output (the set of cities that satisfy the spatial conditions), is illustrated in Figure 19*c*.

Similarly, the query in the skeleton array of Figure 20*a* will retrieve all cities P such that P is *north_east* of some city Q (reference object variable) in map S. The result of this query (B, C, D) is a superset of the previous result, because cities *north_east* of B are also included. We can also have queries with more than one primary, or reference, object. The query in the skeleton array of Figure 20*b*, will retrieve all cities *north_east* of B and *north_west* of some city Q, where Q is *north_east*

Figure 19. Query involving primary object retrieval

a. Skeleton array **b. Subarray to be searched** **c. Results**

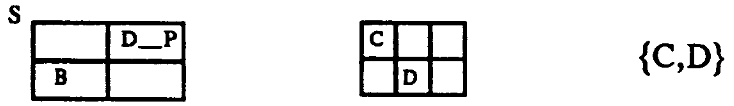
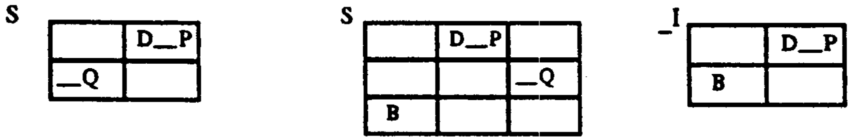


Figure 20. Additional queries involving primary object retrieval

a. Reference object variable **b. Multiple reference objects** **c. Image variable**

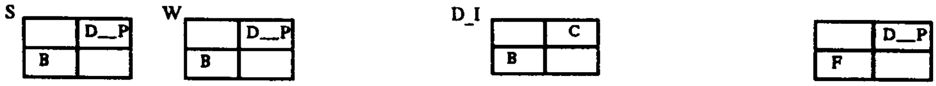


of B in S. The result (city C) is a subset of the cities that are retrieved through the query of Figure 19*a* because the primary objects in this case must satisfy more specialized spatial conditions (all binary constraints among the domain variables and the constants that appear in a skeleton array must be taken into account when processing the query). Unlike the previous queries which refer to map constants, the query in the skeleton array of Figure 20*c* will retrieve all cities *north_east* of B in some map I (map variable). This query, in addition to cities C and D, will also retrieve city J because it is *north_east* of B in map W.

We can use multiple skeleton arrays to express union, intersection, and join. The query in Figure 21*a*, will retrieve all cities *north_east* of B in map S or in map W. PQBE also allows map retrieval. The query in Figure 21*b*, for instance, will retrieve all maps in which city C is *north_east* of B (map S). The previous queries retrieve spatial knowledge explicitly stored in one or more symbolic spatial indexes. PQBE also allows the retrieval of spatial knowledge regarding cities that exist in different maps. This information is not explicitly stored but it can be inferred using appropriate rules of inference encoded in a composition table (composition of spatial relations using symbolic spatial indexes is discussed in the next subsection). The only difference is that in queries involving composition, the skeleton arrays do not correspond to particular indexes and therefore skeleton arrays do not have superscripts denoting specific maps. For instance, the result of the query of Figure 21*c* is the set of all cities in the database that are *north_east* of city F. This query,

Figure 21. Additional features of PQBE

a. Union **b. Image retrieval** **c. Inference**



unlike the previous ones, retrieves not only the cities that are *north_east* of F in some map (B and J are *north_east* of F in map W), but also the cities that exist in different maps and satisfy the spatial condition according to the composition table. Cities C and D are also *north_east* of F because they are *north_east* of B in map S and B is *north_east* of F in map W (transitivity).

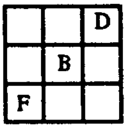
Chang and Fu (1980) developed Query-by-Pictorial Example, a relational query language for manipulating spatial and conventional relations in image databases. Query-by-Pictorial Example allows the expression of spatial queries by pointing on the display screen using special input devices. Despite the name similarity, the concept of Query-by-Pictorial Example is rather different from PQBE. Chang et al. (1987, 1988) proposed a pictorial language similar to PQBE, which is aimed at the retrieval of object configurations. PQBE extends Chang's pictorial language by including variable and constant reference objects, negation, union, intersection, join, and a large set of additional features (for details see Papadias and Sellis, 1994). Further extensions, such as incorporation of topological relations, can be developed to enhance the expressive power of PQBE.

6.2 Composition of Spatial Relations - Image Overlay

The problem of composition can be defined as "if the spatial relation between P and O, and between O and Q is known, what are the possible relations between P and Q?" Frank (1994) presented composition tables for direction relations based on the concepts of projections and cone-shaped directions. Freksa (1992) also studied composition of direction relations in 2-D space, and Egenhofer (1991) studied composition of topological relations. Frank uses the notion of Euclidean approximate to deal with uncertainty, while Freksa's and Egenhofer's systems generate a disjunction of the potential primitive relations when various cases are possible. In this section, we show how symbolic spatial indexes can deal with composition of spatial relations.

Suppose that we want to perform an image overlay using the symbolic spatial indexes of Figure 18 (i.e., we would like to generate an index that preserves information in $S^{M_{D1}}$ and in $W^{M_{D1}}$). The two indexes have an object (B in this case) in common. Let S be the set of representation structures (symbolic spatial

Figure 22. Spatial index com($S^{M_{D1}}$, $W^{M_{D1}}$, B, D, F)



indexes preserving M_{D1}) and O be the set of symbolic object representations. We can define a composition operator *com*:$SxSxOxOxO \rightarrow S$, that takes as input the two indexes to be composed, the common object with respect to which the composition is made, and two other objects each belonging to one array, whose composition relation is to be found. The operator creates one or more output arrays which contain only the three objects that preserve their relative positions. For instance, *com*($S^{M_{D1}}$,$W^{M_{D1}}$,*B,D,F*) will generate the array in Figure 22.

To compute the composition relation, the operator uses the composition table illustrated in Table 1, which shows the relation between objects *P* and *Q* when their relation with a third object *O* is known. Instead of the full name of the relation, we use abbreviations (e.g., NW instead of *north_west*). For instance, the array of Figure 22 is generated using the facts that *D* is *north_east B* in index $S^{M_{D1}}$, and that *B* is *north_east F* in index $W^{M_{D1}}$ and the entry (3,3) of the composition table.

In several cases, there may be multiple possible outputs for image overlay. In such cases the operator, such as the ones in Freksa's and Egenhofer's systems, will generate a list of output arrays corresponding to the possible relations. For instance, *com*($S^{M_{D1}}$, $W^{M_{D1}}$, *B,D,G*) will generate the arrays of Figure 23.

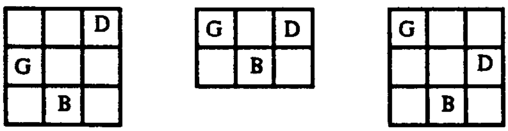
The composition relations are generated using the facts that *D* is *north_east* of *B* in index $S^{M_{D1}}$, and that *B* is *south_east* of *G* in index $W^{M_{D1}}$ and the entry (3,9) of the composition table. The arrays correspond to the three possible relations between *D* and *G* (NE∨RE∨SE) when their relative positions with respect to *B* are known. Notice that symbolic spatial indexes do not preserve metric information; otherwise only one relation could be generated as the result of composition. There is also the possibility that the composition does not produce any information about the possible relation between two objects. For instance, *com*($S^{M_{D1}}$, $W^{M_{D1}}$, *B,D,J*) will generate nine arrays that correspond to the primitive relations of M_{D1} between *D* and *J* when it is also known that *B* is *south_west* of both *D* and *J*; composition does not rule out any possible positions (the entries of the composition table that denote disjunction of nine primitive relations contain the symbol *T*).

Table 1 has been extended to capture ordinal relations for applications such as the one in Figure 10 (Papadias et al., 1994*a*). The same composition operator can handle symbolic spatial indexes that preserve modeling spaces of direction relations

Table 1. Composition Table

		1	2	3	4	5	6	7	8	9
		NW(O,Q)	RN(O,Q)	NE(O,Q)	RW(O,Q)	SP(Y,Z)	RE(O,Q)	SW(O,Q)	RS(O,Q)	SE(O,Q)
1	NW(P,O)	NW	NW	NW∨RN∨NE	NW	NW	NW∨RN∨NE	NW∨RW∨SW	NW∨RW∨SW	T
2	RN(P,O)	NW	RN	NE	NW	RN	NE	NW∨RW∨SW	RN∨SP∨RS	NE∨RE∨SE
3	NE(P,O)	NW∨RN∨NE	NE	NE	NW∨RN∨NE	NE	NE	T	NE∨RE∨SE	NE∨RE∨SE
4	RW(P,O)	NW	NW	NW∨RN∨NE	RW	RW	RW∨SP∨RE	SW	SW	SW∨RS∨SE
5	SP(P,O)	NW	RN	NE	RW	SP	RE	SW	RS	SE
6	RE(P,O)	NW∨RN∨NE	NE	NE	RW∨SP∨RE	RE	RE	SW∨RS∨SE	SE	SE
7	SW(P,O)	NW∨RW∨SW	NW∨RW∨SW	T	SW	SW	SW∨RS∨SE	SW	SW	SW∨RS∨SE
8	RS(P,O)	NW∨RW∨SW	RN∨SP∨RS	NE∨RE∨SE	SW	RS	SE	SW	RS	SE
9	SE(P,O)	T	NE∨RE∨SE	NE∨RE∨SE	SW∨RS∨SE	SE	SE	SW∨RS∨SE	SE	SE

Figure 23. Spatial indexes corresponding to com ($S^{M_{D1}}$, $W^{M_{D1}}$, B,D,G)



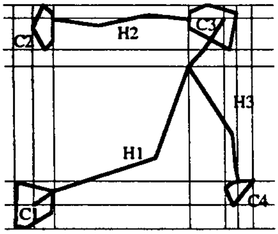
using two or more representative points per object, but the number of possible outputs grows exponentially with the number of points used to represent the objects. However, the problem of composing both direction and topological relations (e.g., arrays preserving M_{D2T1}), is more complicated and requires analysis beyond the scope of this article.

6.3 Route Planning

Route planning has been extensively studied in areas of Artificial Intelligence, such as motion planning and robot navigation. Previous systems designed to deal with this problem include TOUR (Kuipers, 1978) and SPAM (McDermott and Davis, 1984). Holmes and Jungert (1992) demonstrated how symbolic projections can be applied to knowledge-based route planning in digitized maps. In this section, we show how symbolic spatial indexes can be used to define routes connecting two objects. Consider the map of Figure 24*a*, in which the regions represent cities and the lines correspond to highways that connect them. Figure 24*b* illustrates the corresponding symbolic spatial index $T^{M_{D2T1}}$.

Figure 24. Map of cities and highways

a. Initial map t



b. Spatial index $T^{M_{D2T1}}$

7			$C2_s$			$C3_s$	
6			$H2_f$,H2C2	$H2_s$,H2C3	$H1_s$, H1C3		
5		$C2_f$		$C3_f$			
4				H1H3		$H3_s$	
3			$C1_s$	$H3_f$		C4H3	$C4_s$
2		$H1_f$,C1H1			$C4_f$		
1	$C1_f$						
	1	2	3	4	5	6	7

A direct connection between cities C_k and C_l corresponds to a highway that intersects with both cities, and can be defined as: C_k *d_connects* $C_l \equiv \exists H_m \exists i,j,g,f\ [S(C_k H_m,i,j) \wedge S(H_m C_l,g,f)]$. H_1 directly connects cities C_1 and C_3 since $S(C_1 H_1,2,2) \wedge S(H_1 C_3,6,5)$.

We use the notation H_m *reaches* H_n to denote that highway H_n can be reached from H_m using highway intersections (for all the examples we assume bi-directional highways). The relation *reaches* can be defined as: H_m *reaches* $H_n \equiv \exists i,j\ [S(H_m H_n,i,j)] \vee \exists H_k [H_m$ *reaches* $H_k \wedge H_k$ *reaches* $H_n]$. H_1 *reaches* H_3 because $S(H_1 H_3,4,4)$.

An indirect connection is achieved through a highway that intersects with the first city, and a second highway which reaches the first one (directly or indirectly) and passes from the second city. Indirect connection through highways can be defined as: C_k *i_connects* $C_l \equiv \exists H_m \exists H_n \exists i,j,g,f\ [S(C_k H_m,i,j) \wedge H_m$ *reaches* $H_n \wedge S(H_n C_l,g,f)]$. C_1 and C_4 are indirectly connected through H_1 and H_3 since $S(C_1 H_1,2,2) \wedge H_1$ *reaches* $H_3 \wedge S(H_3 C_4,3,6)$.

Additional types of connections, such as the connection of two cities through an intermediate city, can be defined to express situations involving routes from one city to another. In procedural terms, the previous definitions reduce to a search for topological representative points involving lines. When several choices are available, direction information can be used to choose a line that is in the direction of the destination. Furthermore, composition can be used in route finding when the start and the destination cities exist in different symbolic spatial indexes (provided that a common object belongs to the route).

6.4 Update Operations

Spatial knowledge representation systems deal with a dynamic environment in which a change in a single item of knowledge may have widespread effects. In logic-based systems, the assertion of a new fact may invalidate previous inferences. The problem

Figure 25. Object removal using PQBE

a. Skeleton array **b. New index**

of updating a system's representation of the state of the world to reflect the effects of actions is known as the *frame problem*. Extensions of PQBE can be used to handle update operations in symbolic spatial indexes. Object deletion from an index can be treated in a way similar to object retrieval. The character R before a variable causes the contents of the variable to be removed from the index array. The query in the skeleton array of Figure 25*a* will cause all the objects P such that P is *restricted_east* of A to be removed from $S^{M_{D1}}$. After all the symbols denoting objects that satisfy the previous conditions (object E) are deleted, the empty rows and columns are removed, and the resulting array is illustrated in Figure 25*b*.

Object insertion can also be handled by PQBE, provided that the position of the object to be added is precisely specified. Otherwise, the insertion of an object using conditions expressed in PQBE may result in several output arrays corresponding to the possible positions of the new object. Another way to insert an object in a symbolic spatial index is by explicitly specifying the cells where each point of the object is to be added. Furthermore, object movements within the array can be handled by removing the object from its old positions and inserting it into the new ones.

Related to update operations are the problems of spatial knowledge assimilation and error correction (Davis, 1986). The goal of assimilation, given an accurate representation of spatial knowledge and an accurate fact, is to augment the representation to include the new fact. The goal of error correction, given a spatial knowledge representation which is not quite accurate and a new fact which is more precise, is to improve the representation. Both of these problems could be treated by an operator similar to the composition operator, which takes two symbolic spatial indexes as arguments (one representing the initial state and one representing the new fact), and generates one or more output indexes that describe the new state.

7. Conclusion

This article deals with the qualitative representation of spatial knowledge and, in particular, with the representation of binary direction and topological relations in 2-D space. Various relation-based systems concerned with the representation of spatial relations have been developed in several areas with different processing tasks. Graph representations and logic-based formalisms have been used in qualitative spatial reasoning. Intersection matrices describe binary topological relations for a set of objects and compute the composition relations among them. 2-D strings represent images with 1-D encodings, and symbolic arrays preserve the spatial structure of complex entities in aggregation hierarchies.

We described a new relation-based structure, the symbolic spatial index, which applies several different ideas in qualitative spatial knowledge representation. The set of spatial relations that is explicitly represented in a symbolic spatial index, its modeling space, is determined by the choice of representative points. We have dealt with modeling spaces consisting of direction relations using one and two points per object, and topological relations using one symbol per object. Further extensions are possible for applications that require higher direction and topological resolution.

Table 2 summarizes the properties of several relation-based systems. The first two columns refer to the representation systems and the type of spatial relations that they have been used to represent. Column 3 describes construction processes that have been developed to generate relation-based representations from other forms of spatial knowledge. All the systems are ambiguous (column 4), and some of them are unique (column 5) according to the definitions in Section 2.1. For the rest of the systems uniqueness cannot be determined due to the lack of a definition for the construction process. Column 6 classifies the systems as deductive or non-deductive according to the discussion in Sections 2.2 and 2.3. Column 7 refers to completeness of representations with respect to the assumed modeling space.

Spatial information processing using relation-based representations involves symbolic and not numerical computation and avoids the usual problems of geometric representations. Although relation-based systems cannot be applied in all domains involving spatial knowledge, we believe that there is a wide scope of potential applications ranging from qualitative reasoning and spatial databases to robot navigation and computational vision and imagery. In this article, we have shown how symbolic spatial indexes can be used to handle information retrieval, image overlay, route planning, and update operations, although additional tasks (such as image similarity retrieval) are not excluded.

Topics that emerged during this work and can be considered for further investigation include:

- the combinatorial study of the number of direction and topological relations that we can represent as a function of the number and the properties of the representative points used for each object;
- the integration of direction and topological reasoning within one framework;
- the notion of image equivalence with respect to a modeling space, ambiguity, uniqueness, and completeness of spatial relation-based representations;
- comparative studies of the previous representational systems with respect to storage requirements, computational efficiency, expressive power, and inferential adequacy.

Table 2. Properties of spatial relation-based representation systems

Representation system	Modeling space	Construction process	Ambiguous	Unique	Deductive	Complex
Logic-based	direction and/or topological	No	Yes	–	Yes	Yes
Intersection matrices	topological	No	Yes	–	Yes	Yes
2D-G strings	direction (and topological)	symbolic projection	Yes	Yes	No	Yes
Symbolic arrays	direction and ordinal	application dependent	Yes	–	No	No
Spatial indexes	direction (and topological)	modeling space dependent	Yes	Yes	No	Yes

Acknowledgments

This research was supported by the ESPRIT Basic Research project 6881 (AMUSING). We thank Nora Sleumer for her helpful suggestions. Furthermore, the comments form Ralf Harmut Güting and three anonymous reviewers contributed significantly to the improvement of the article.

References

Allen, J.F. Maintaining knowledge about temporal intervals. *CACM,* 26(11):832-843, 1983.

Bruegger, B. and Muller, J.-C. Mechanisms of geometric abstraction. *Proceedings of the Fifth International Symposium on Spatial Data Handling,* Charleston, NC, 1992.

Buisson, L. Reasoning on space with object-centered knowledge representations. *Proceedings of the First Symposium on the Design and Implementation of Large Spatial Databases,* 1989.

Chang, N.S. and Fu, K.S. Query-by-pictorial example. *IEEE Transactions on Software Engineering,* 6(6):519-524, 1980.

Chang, S.K., Shi, Q.Y., and Yan C.W. Iconic indexing by 2-D string. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 9(3):413-428, 1987.

Chang, S.K., Yan, C.W., Arndt, T., and Dimitroff, D. An intelligent image database system. *IEEE Transactions on Software Engineering,* Special issue on image databases, 1988, pp 681-688.

Chang, S.K, Erland, J., and Li, Y. The design of pictorial database upon the theory of symbolic projections. *Proceedings of the First Symposium on the Design and Implementation of Large Spatial Databases,* 1989.

Clementini, E., Sharma, J., and Egenhofer, M. Modeling topological spatial relations: Strategies for query processing. *International Journal of Computer and Graphics,* to appear.

Davis, E. *Representing and Acquiring Geographic Knowledge.* Morgan Kaufmann, 1986.

Dutta, S. Qualitative spatial reasoning: A semi-qualitative approach using fuzzy logic. *Proceedings of the First Symposium on the Design and Implementation of Large Spatial Databases,* 1989.

Egenhofer, M. Reasoning about binary topological relations. *Advances in Spatial Databases,* Second Symposium, 1991.

Egenhofer, M. and Herring, J. A mathematical framework for the definitions of topological relationships. *Proceedings of the Fourth International Symposium on Spatial Data Handling,* Zurich, 1990.

Egenhofer, M. and Sharma, J. Assessing the consistency of complete and incomplete topological information. *Geographical Systems,* 1(?):47-68, 1993.

Frank, A.U. Qualitative spatial reasoning about distances and directions in geographic space. *Journal of Visual Languages and Computing,* 3(?):343-371, 1992.

Frank, A.U. Qualitative spatial reasoning: Cardinal directions as an example. To appear in the *International Journal of Geographic Information Systems,* 1994.

Freksa, C. Qualitative spatial reasoning. In: Mark, D.M. and Frank, A.U., eds. *Cognitive and Linguistic Aspects of Geographic Space.* Dordrecht, The Netherlands: Kluwer, 1991.

Freksa, C. Using orientation information for qualitative spatial reasoning. In: Frank, A.U., Campari, I., Formentini, U., eds. *Proceedings of the International GIS Conference—From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space,* Pisa, Italy, 1992.

Glasgow, J.I. and Papadias, D. Computational imagery. *Cognitive Science,* 16:355-394, 1992.

Glasgow, J.I., Fortier, S., and Allen, F.H. Molecular scene analysis: Crystal structure determination through imagery. In: Hunter, L., ed. *Artificial Intelligence and Molecular Biology.* AAAI Press, 1992.

Güsgen, H.W. Spatial reasoning based on Allen's temporal logic. TR-89-049, International Computer Science Institute, Berkeley, 1989.

Hernàndez, D. Maintaining qualitative spatial knowledge. *Proceedings of the European Conference on Spatial Information Theory,* Elba, Italy, 1993.

Herskovits, A. *Language and Spatial Cognition.* Cambridge University Press, 1986.

Holmes, P.D. and Jungert, E. Symbolic and geometric connectivity graph methods for route planning in digitized maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14(5):549-565, 1992.

Jackendoff, R. *Semantics and Cognition.* MIT Press, 1983.

Kainz, W., Egenhofer, M., and Greasley, I. Modeling spatial relations and operations with partially ordered sets. *International Journal of Geographic Information Systems,* 7(3):215-229, 1993.

Kuipers, B.J. Modeling spatial knowledge. *Cognitive Science,* 2(?):129-153, 1978.

Larkin, J. and Simon, H.A. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science,* 11(?):65-99, 1987.

Lee, S.-Y., Yang, M.-C., and Chen, J-W. Signature file as a spatial filter for iconic image database. *Journal of Visual Languages and Computing,* 3(?):373-397, 1992.

Levine, M. A knowledge-based computer vision system. In: Hanson, A. and Riseman, E., eds. *Computer Vision Systems.* New York: Academic Press, 1978, pp. 335-352.

Lindsay, R.K. Images and inference. *Cognition,* 29(?):229-250, 1988.

Mark, D. Counter-intuitive geographic "facts": Clues for spatial reasoning at geographic scales. *Proceedings of the International GIS Conference—From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space.* Pisa, Italy, 1992.

McDermott, D.V. and Davis, E. Planning routes through uncertain territory. *Artificial Intelligence,* 22(?):107-156, 1984.

Mukerjee, A. and Joe, G. A qualitative model for space. Technical Report TAMU 90-005, Texas A&M University, 1990.

Papadias, D. Relation-based representations for spatial knowledge. KDBSLAB-TR-94-05. Computer Science Division, Department of Electrical and Computer Engineering, National Technical University of Athens, 1994.

Papadias, D. and Glasgow, J.I. A knowledge representation scheme for computational imagery. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society,* Chicago, IL, 1991.

Papadias, D. and Sellis, T. Spatial reasoning using symbolic arrays. *Proceedings of the International GIS Conference—From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space,* Pisa, Italy, 1992.

Papadias, D. and Sellis, T. The semantics of relations in 2-D space using representative points: Spatial indexes. *Proceedings of the European Conference on Spatial Information Theory,* Elba, Italy, 1993.

Papadias, D. and Sellis, T. A pictorial language for the retrieval of spatial relations from image databases. *Proceedings of the Sixth International Symposium on Spatial Data Handling,* Edinburgh, UK, 1994.

Papadias, D., Frank, A.U., and Koubarakis, M. Constraint-based reasoning in geographic databases: The case of symbolic arrays. *Proceedings of the Second ICLP Workshop on Deductive Databases,* Santa Margherita Ligure, Italy, 1994*a*.

Papadias, D., Theodoridis, Y., and Sellis, T. The retrieval of direction relations using R trees. *Proceedings of the Fifth Conference on Database and Expert Systems Applications,* Athens, 1994*b*.

Peuquet, D. The use of spatial relationships to aid spatial database retrieval. *Proceedings of the Second International Symposium on Spatial Data Handling,* Seattle, WA, 1986.

Peuquet, D. and Ci-Xiang, Z. An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. *Pattern Recognition,* 20(1):65-74, 1987.

Pullar, D. and Egenhofer, M. Toward formal definitions of topological relations among spatial objects. *Proceedings of the Third International Symposium on Spatial Data Handling,* Sydney, Australia, 1988.

Randell, D.A., Cui, Z., and Cohn., A. A spatial logic based on regions and connection. *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning,* 1992.

Requicha, A. Representations for rigid solids: Theory, methods and systems. *ACM Computing Surveys,* 12(4):437-464, 1980.

Roussopoulos, N., Faloutsos, C., and Sellis, T. An efficient pictorial database system for PSQL. *IEEE Transactions on Software Engineering,* Special issue on image databases, pp. 639-650, 1988.

Sistla, P., Yu, C., and Haddad, R. Reasoning about spatial relationships in picture retrieval systems. *Proceedings of the Twentieth VLDB Conference,* Santiago, Chile, 1994.

Smith, T. and Park, K. Algebraic approach to spatial reasoning. *International Journal of Geographic Information Systems,* 6(3):117-192, 1992.

Ullman, J. *Principles of Databases and Knowledge Base Systems,* Vol. 1. Computer Science Press, 1988.

Vieu, L. A logical framework for reasoning about space. *Proceedings of the European Conference on Spatial Information Theory,* Elba, Italy, 1993.