### Title
Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees (94-8)




### Authors
Papadias, Dimitris
Theodoridis, Yannis
Sellis, Timos
et al.



### Publication Date
1994-12-01

# NCGIA

## National Center for Geographic Information and Analysis

# Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees

by

Dimitris Papadias
University of California, San Diego

Yannis Theodoridis
National Technical University of Athens

Timos Sellis
National Technical University of Athens

Max J. Egenhofer
University of Maine, Orono

Technical Report 94-8

December 1994

**Simonett Center for Spatial Analysis**
**University of California**
35 10 Phelps Hall
Santa Barbara, CA 93106-4060
Office (805) 893-8224
Fax (805) 893-8617
ncgia@ncgia.ucsb.edu

**State University of New York**
301 Wilkeson Quad, Box 610023
Buffalo NY 14261-0001
Office (716) 645-2545
Fax (716) 645-5957
ncgia@ubvms.cc.buffalo.edu

**University of Maine**
348 Boardman Hall
Orono ME 04469-5711
Office (207) 581-2149
Fax (207) 581-2206
ncgia@spatial.maine.edu

# Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees

Dimitris Papadias[+]
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, California, USA 92093-0114
dimitris@cs.ucsd.edu

Yannis Theodoridis[*]
Department of Electrical and Computer Engineering
National Technical University of Athens
Zographou, Athens, GREECE 15773
theodor@theseas.ntua.gr

Timos Sellis[*]
Department of Electrical and Computer Engineering
National Technical University of Athens
Zographou, Athens, GREECE 15773
timos@theseas.ntua.gr

Max J. Egenhofer[**]
National Center for Geographic Information and Analysis
University of Maine
Orono, Maine, USA 04469-5711
max@mecan1.maine.edu

**Abstract** Recent developments in spatial relations have led to their use in numerous applications involving spatial databases. This paper is concerned with the retrieval of topological relations in Minimum Bounding Rectangle-based data structures. We study the topological information that Minimum Bounding Rectangles convey about the actual objects they enclose, using the concept of projections. Then we apply the results in R-trees and their variations, $R^+$-trees and $R^*$-trees in order to minimise disk accesses for queries involving topological relations. We also investigate queries that involve complex spatial conditions in the form of disjunctions and conjunctions and we discuss possible extensions.

## 1. INTRODUCTION

The representation and processing of spatial relations has recently gained much attention in Spatial Query Languages (Egenhofer, 1994, Papadias and Sellis, 1994b), Image and Multimedia Databases (Papadias et al., 1994a, Sistla et al., 1994), Geographic Applications (Frank, 1994), Spatial Reasoning (Glasgow, 1994) and Cognitive Science (Glasgow and Papadias, 1992). Several kinds of spatial relations have been defined and used. Egenhofer and Franzosa (1991), for instance, provided a mathematical framework for the definition of topological relations (e.g., *inside, overlap*), while Papadias and Sellis (1993) defined direction relations (e.g., *north, north_east*) using representative points. Kainz et al. (1993) modelled order relations

[+] Work partially supported by NSF - IRI 9221276. Part of this work was done while the author was with the Department of Geoinformation, Technical University of Vienna.
[*] Work partially supported by the Department of Research and Technology of Greece (PENED 91).
[**] Work partially supported by the National Science Foundation under grant numbers IRI-9309230 and SBR-8810917 (for the National Center for Geographic Information and Analysis), Intergraph Corporation, Environmental Systems Research Institute, the Scientific Division of the North Atlantic Treaty Organization, and the Maine Mathematics and Science Alliance.

(e.g., *part_of*) using partially ordered sets and Frank (1992) proposed a method for qualitative reasoning that combines direction with distance relations.

Despite the attention that spatial relations have attracted in other application domains, they have not been extensively applied in spatial data structures. So far most of the work on spatial access methods has concentrated on the retrieval of distance information and on the relations *disjoint* and *not_disjoint*. This is not because other spatial relations are unimportant for practical applications, but mostly because of the lack, until recently, of definitions for spatial relations.

In this paper we focus on the retrieval of *topological* relations, relations that stay invariant under topological transformations such as translation, rotation and scaling. In particular we will deal with topological relations between *contiguous region objects*[1] without holes as defined by the 9-intersection model (Egenhofer, 1991). According to this model, each object p is represented in 2D space as a point set which has an interior, a boundary and an exterior. The topological relation between any two objects (point sets) p and q is described by the nine intersections of p's interior, boundary and exterior, with the interior, the boundary and the exterior of q (based on the concepts of point-set topology). Out of 512 different relations that can be distinguished by the model, only the following eight are meaningful for region objects: *disjoint, meet, equal, overlap, contains* (and the inverse relation *inside*) and *covers* (and the converse *covered_by*)(see Figure 1).

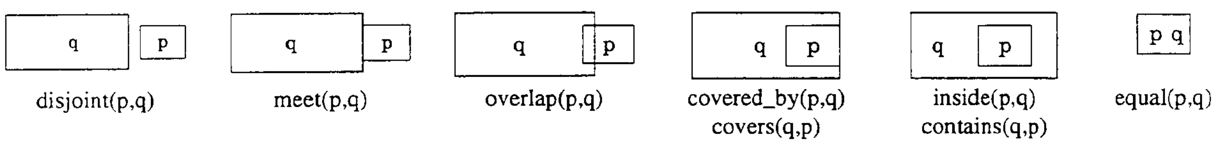



**Fig. 1** Topological relations

We call the previous set of topological relations $m_{t2}$ in order to distinguish them from the set $m_{t1}$={*disjoint*, *not_disjoint*[2]}. The meaning of *disjoint* is the same in both sets (it implies that two objects have no common points), while all the other relations of $m_{t2}$ are refinements of *not_disjoint*. The relations of $m_{t2}$ are pairwise disjoint, and they provide a complete coverage. Randell et al. (1992) reached the same set of pairwise disjoint topological relations following an axiomization based on mereology and expressed in a many-sorted logic. Related work can be found also in (Vieu, 1993).

Tests with human subjects have shown evidence that the 9-intersection model has potential for defining cognitively meaningful spatial predicates, a fact that renders the above relations a good candidate for commercial systems (Mark and Egenhofer, 1994). In fact, the 9-intersection model has been implemented in Geographical Information Systems (GIS); Hadzilacos and Tryfona (1992), for instance, used it to express geographical constraints, and Mark and Xia (1994) to determine spatial relations in ARC/INFO. In addition there are implementations in commercial systems such as Intergraph (MGE, 1993) and Oracle

[1] The term refers to homogeneously 2-dimensional, connected objects with connected boundaries.
[2] Sometimes in spatial data structures bibliography, the term *overlap* (instead of *not_disjoint*) is used to denote any configuration in which the objects are not *disjoint*.

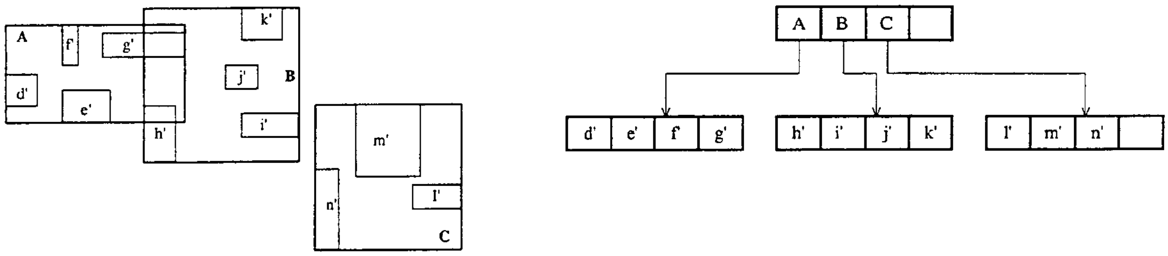
(Keighan, 1993). Furthermore, the relations of $m_{t2}$ have been influential in assessing the consistency of topological information in spatial databases (Egenhofer and Sharma, 1993), query optimisation strategies (Clementini et al., 1993) and Spatial Reasoning (Sharma et al., 1994).

This paper is concerned with the retrieval of the topological relations of $m_{t2}$ using spatial data structures based on Minimum Bounding Rectangles (MBRs). In particular we concentrate on R-trees and their variations. Section 2 describes briefly MBR-based spatial data structures. Section 3 illustrates the possible relations between MBRs and describes the corresponding topological relations. Section 4 investigates the topological information that MBRs convey about the actual objects they enclose with respect to the 9-intersection model. Section 5 applies the results in R-tree-based data structures and compares the retrieval times. Section 6 is concerned with queries that involve complex spatial conditions. Section 7 discusses extensions that deal with imprecision and Section 8 concludes with comments on future work.

## 2. SPATIAL DATA STRUCTURES BASED ON MINIMUM BOUNDING RECTANGLES

It is a common strategy in spatial access methods to store object approximations and use these approximations to index the data space in order to efficiently retrieve the potential objects that satisfy the result of a query. Depending on the application domain there are several options in choosing object approximations. Brinkhoff et al. (1993) compare the use of *rotated minimum bounding rectangles*, *convex hulls* and *minimum bounding n-corner convexes* in spatial access methods. In this paper we examine methods based on the traditional approximation of Minimum Bounding Rectangles. MBRs have been used extensively to approximate objects in Spatial Data Structures and Spatial Reasoning because they need only two points for their representation; in particular, each object $q$ is represented as an ordered pair $(q'_l, q'_u)$ of points that correspond to the lower left and the upper right point of the MBR $q'$ that covers q ($q'_l$ stands for the lower and $q'_u$ for the upper point of the MBR). While MBRs demonstrate some disadvantages when approximating non convex or diagonal objects, they are the most commonly used approximations in spatial applications.

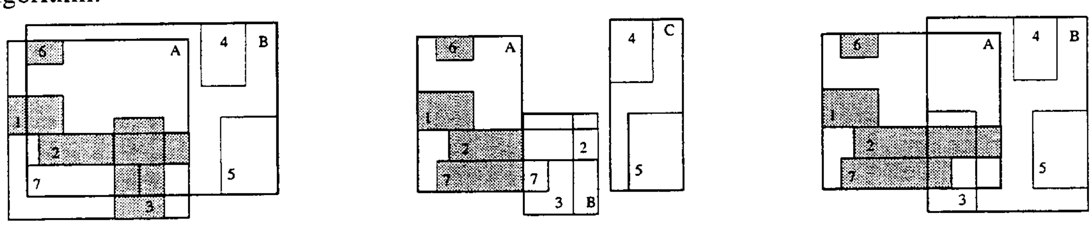
The most promising group of spatial data structures based on MBRs includes R-trees (Guttman, 1984) and their variations. The R-tree data structure is a height-balanced tree, which consists of intermediate and leaf nodes (R-trees are direct extensions of B-trees in k-dimensions). The MBRs of the actual data objects are assumed to be stored in the leaf nodes of the tree. Intermediate nodes are built by grouping rectangles at the lower level. An intermediate node is associated with some rectangle which encloses all rectangles that correspond to lower level nodes. Each pair of nodes may satisfy any of the topological relations of $m_{t2}$. Figure 2 shows an example set of data rectangles and the corresponding R-tree built on these rectangles. MBRs are denoted by accented small letters and intermediate rectangles by capitals (in this example we assume a branching factor of 4, i.e., each intermediate node contains at most four entries).




MBRs and intermediate rectangles R-tree

**Fig. 2** Some MBRs grouped into intermediate rectangles and the corresponding R-tree

The fact that R-trees permit overlap among node entries sometimes leads to unsuccessful hits on the tree structure. The R$^+$-tree (Sellis et al., 1987) and the R$^*$-tree (Beckmann et al., 1990) methods have been proposed to address the problem of performance degradation caused by the overlapping regions or excessive dead-space respectively. To avoid this problem, R$^+$-tree achieves zero overlap among intermediate node entries by allowing partitions to split nodes. The trade-off is that more space is required because of the duplicate entries and thus the height of the tree may be greater than the original R-tree. On the other hand, the R$^*$-tree permits overlap among nodes, but tries to minimise it by organising rectangles into nodes using a more complex algorithm than the one of the original R-tree. Figure 3 illustrates how each method would organise a set of rectangles into intermediate nodes. The original R-tree would create two intermediate nodes A and B (the grey MBRs belong to node A). The R$^+$-tree algorithm splits rectangles that fall on some partition line (rectangles 2 and 7) achieving zero overlap between A and B. This results in duplicate entries that may increase the number of intermediate nodes (if we assume a branching factor of 4 we have an extra node C). The R$^*$-tree algorithm groups rectangles in a different way in order to achieve smaller overlap and coverage area of the intermediate nodes compared to the original R-tree algorithm.




R-tree R$^+$-tree R$^*$-tree

**Fig. 3** Some rectangles and the corresponding grouping in the R-tree variants

If the only topological relations of interest are the relations of $m_{t1}$, then when the MBRs of two objects are *disjoint* we can conclude that the objects that they represent are also *disjoint*. If the MBRs however share common points, no conclusion can be drawn about the topological relation between the objects. For this reason, spatial queries involve the following two step strategy:

1. *Filter step*: The tree is used to rapidly eliminate objects that could not possibly satisfy the query. The result of this step is a set of candidates which includes all the results and possibly some false hits.

2. *Refinement step*: Each candidate is examined (by using computational geometry techniques). False hits are detected and eliminated.

Brinkhoff et al. (1993b) extended the above strategy to include a second filter step with finer approximations than MBR (e.g. *convex hulls*) in order to exclude some false hits from the set of candidates. Furthermore, other approximations, like *maximum enclosing rectangle*, could be used in the second filter step to detect hits for which the refinement step is not necessary (Brinkhoff et al., 1994). The above techniques speed-up the retrieval of relations of $m_{t1}$ using R-trees and variations. The present work uses the original two-step strategy to explore a larger, and more detailed set of topological relations[3]. Such an investigation is important because spatial queries frequently require the kind of qualitative resolution distinguished by $m_{t2}$. We first study how MBR approximations can be used for the retrieval of topological relations of $m_{t2}$ between actual objects and then we apply the results in actual implementations.

## 3. TOPOLOGICAL RELATIONS BETWEEN MINIMUM BOUNDING RECTANGLES

The term *primary object* denotes the object to be located and the term *reference object* denotes the object in relation to which the primary object is located (in the examples, the reference objects q are grey, while the primary objects p are transparent). Let $X$ and $Y$ be functions that return the $x$ and y coordinate of a point respectively. When we use two points for the representation of the reference object and for the case of one-dimensional space, the axis is divided into 5 partitions. Three partitions are open line segments (the interior and the exteriors of line segment $q'_l q'_u$) and two partitions are the points $q'_l$ and $q'_u$ (see Figure 4). If the primary object p is also represented by two points $p'_l$ and $p'_u$ ordered on the x axis ($X(p'_l) < X(p'_u)$) then the number of pairwise disjoint relations between the two objects in 1D space is 13. These 13 relations correspond to the relations between time intervals introduced in (Allen, 1983) and applied for 1D spatial reasoning in (Pullar and Egenhofer, 1988). The symbols $q'_l$ and $q'_u$ in Figure 4 denote the edge points (lower and upper) for the reference object and the characters *l* and *u* the lower and the upper points of the primary object.

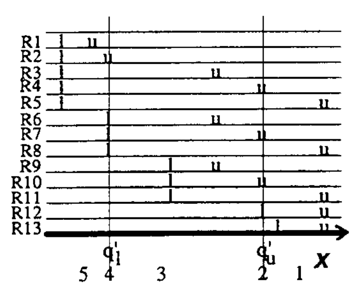



**Fig. 4** Possible relations in 1D space

[3] In a previous paper we have shown how the strategy can be applied for the retrieval of direction relations between extended objects (Papadias et al., 1994b).

In order to study the correspondence between MBRs and actual objects with respect to the relations of $m_{t2}$ we will apply the previous results in 2D space using projections on the x and y axis. In this case, the constraint for the lower and the upper points of the bounding rectangle is: $X(p'_l) < X(p'_u) \wedge Y(p'_l) < Y(p'_u)$ and the number of pairwise disjoint relations is 169 (the square of the number of relations in 1D space). These relations are illustrated in Figure 5; they correspond to the highest accuracy using two points per object, in the sense that they cannot be defined as disjunctions of more "restrictive" relations.

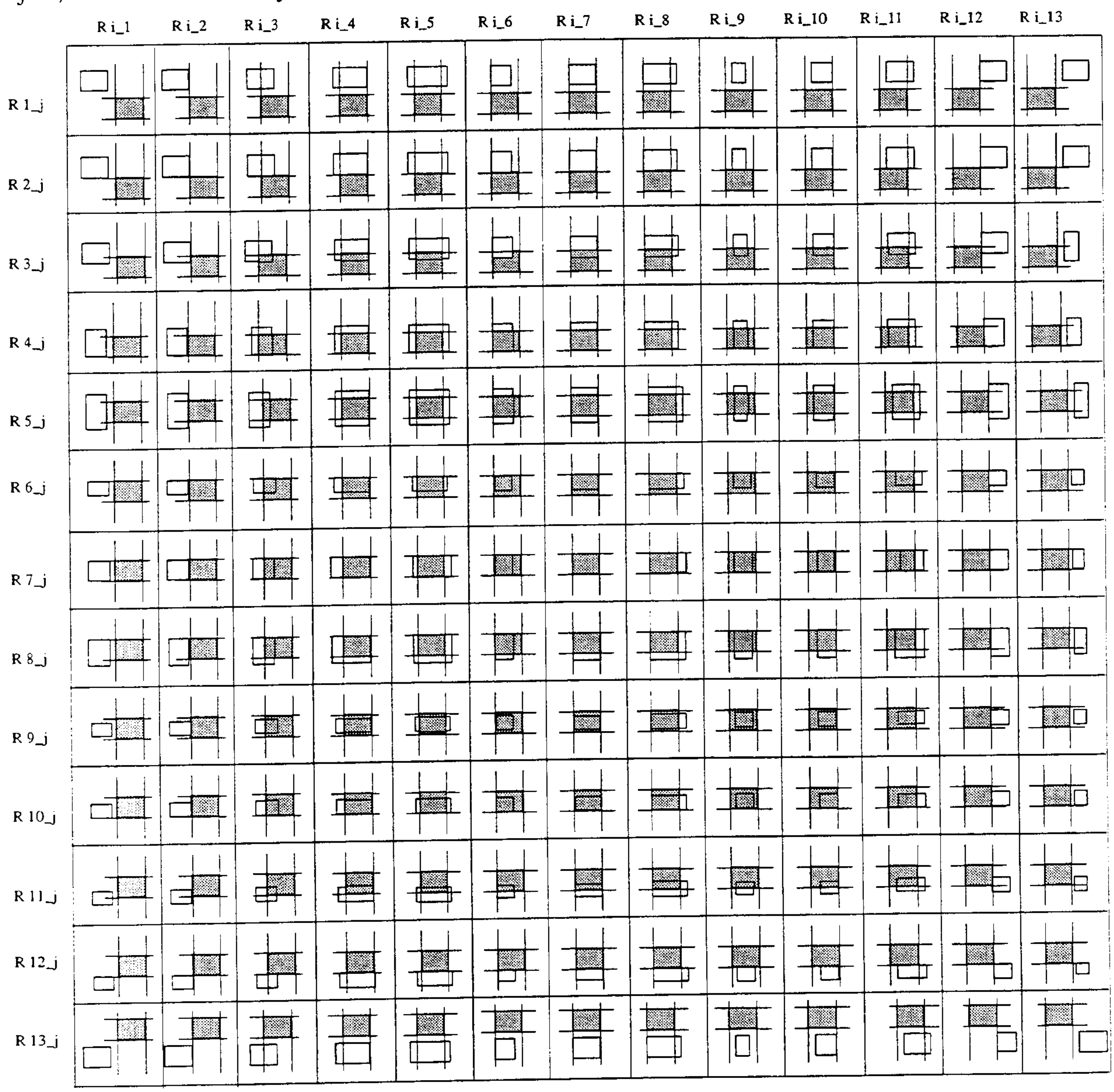



**Fig. 5** Possible relations between MBRs

Figure 6 illustrates the corresponding topological relation for the 169 configurations of Figure 5. For instance, all the configurations of the first row ($R_{1\_j}$ where j can take any value from 1 to 13) correspond to the *disjoint* relation, and the total number of configurations in which the MBRs are *disjoint* is 48. On the other hand, only relation $R_{5\_5}$ corresponds to the topological relation *contains*.

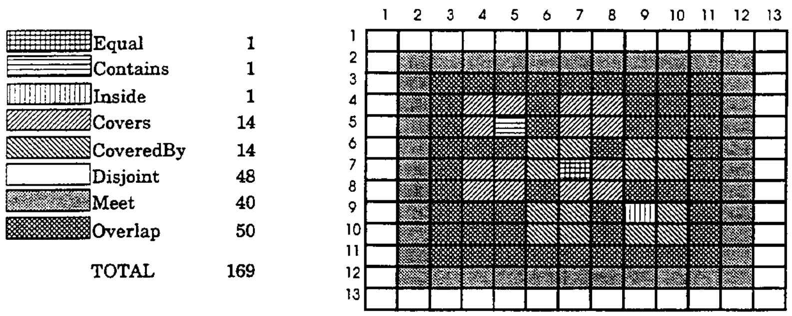



**Fig. 6** Topological relations between MBRs

Since the MBRs are only approximations of the actual objects, the topological relation between MBRs does not necessarily coincide with the topological relation between the objects. In most cases the MBRs of objects that satisfy a given relation, should satisfy a number of possible relations with respect to the MBR of the reference object. For instance, in order to answer the query "find all objects that *cover* a given object" we need to retrieve the MBRs that satisfy the relations *covers, contains,* or *equal* with respect to the MBR of the reference object. Figure 7 illustrates three MBR configurations that satisfy these three relations and the primary object *covers* the reference object.

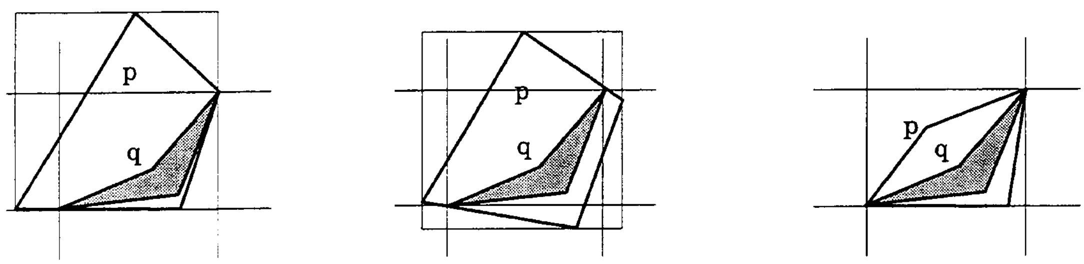



**Fig. 7** Configurations of MBRs for which the primary object *covers* the reference object

According to Figure 6, the MBRs that satisfy the relations *covers, contains,* or *equal* with respect to q' correspond to configurations $R_{i\text{-}j}$ where i and j in {4,5,7,8}. In the next section we distinguish the MBRs that could enclose primary objects that satisfy each topological relation of $m_{t2}$.

## 4. TOPOLOGICAL RELATIONS THAT MBRS CONVEY ABOUT THE ACTUAL OBJECTS

We will start with relations that involve the retrieval of a small number of MBRs and we will gradually move to relations for which a large number of MBRs should be retrieved. In order to answer the query "find all objects p *equal* to object q" we need to retrieve all MBRs that are *equal* with q', that is all MBRs that satisfy the relation $R_{7\text{-}7}$ with respect to q'. Only these MBRs may enclose objects that satisfy the query. On the other hand, the retrieved MBRs may enclose objects that satisfy the relations *equal, overlap, covered_by, covers* or *meet* with respect to q (see Figure 8). As in the case of $m_{t1}$, a refinement step is needed when dealing with the relations of $m_{t2}$ if the MBRs of the retrieved objects are not *disjoint*.

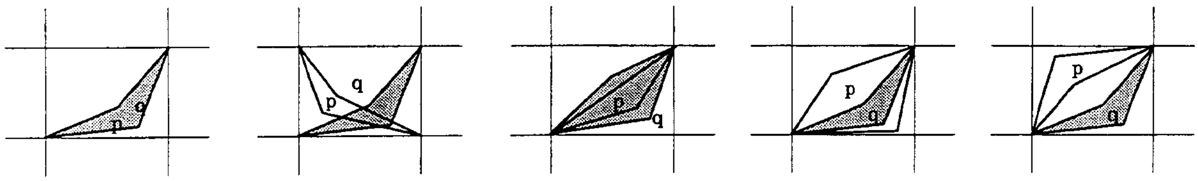


**Fig. 8** Possible relations for objects when the MBRs are *equal*

The relations *contains* and *inside* also involve the retrieval of MBRs that satisfy unique configurations. In particular, the objects that *contain* q can only be in MBRs that *contain* q' (MBRs that satisfy the relation $R_{5\text{-}5}$ with respect to q'), while the objects *inside* q, can only be in MBRs that are *inside* q' (MBRs that satisfy the relation $R_{9\text{-}9}$ with respect to q'). As in the case of *equal*, a refinement step is needed because:

- contains(p',q') $\Rightarrow$ disjoint(p,q)$\vee$meet(p,q)$\vee$overlap(p,q)$\vee$contains(p,q)$\vee$covers(p,q) and
- inside(p',q') $\Rightarrow$ disjoint(p,q)$\vee$meet(p,q)$\vee$overlap(p,q)$\vee$inside(p,q) $\vee$covered_by(p,q)

The rest of the relations involve the retrieval of more than one MBR configurations. In case of *covers,* all MBRs that satisfy the relations $R_{i\text{-}j}$ where I and j in {4,5,7,8} with respect to q' should be retrieved. As Figure 9 illustrates, these MBRs may enclose objects that *cover* the reference object. Similarly the retrieval of *covered_by* involves all MBRs that satisfy the relations $R_{i\text{-}j}$ where i,j in {6,7,9,10} with respect to q'.

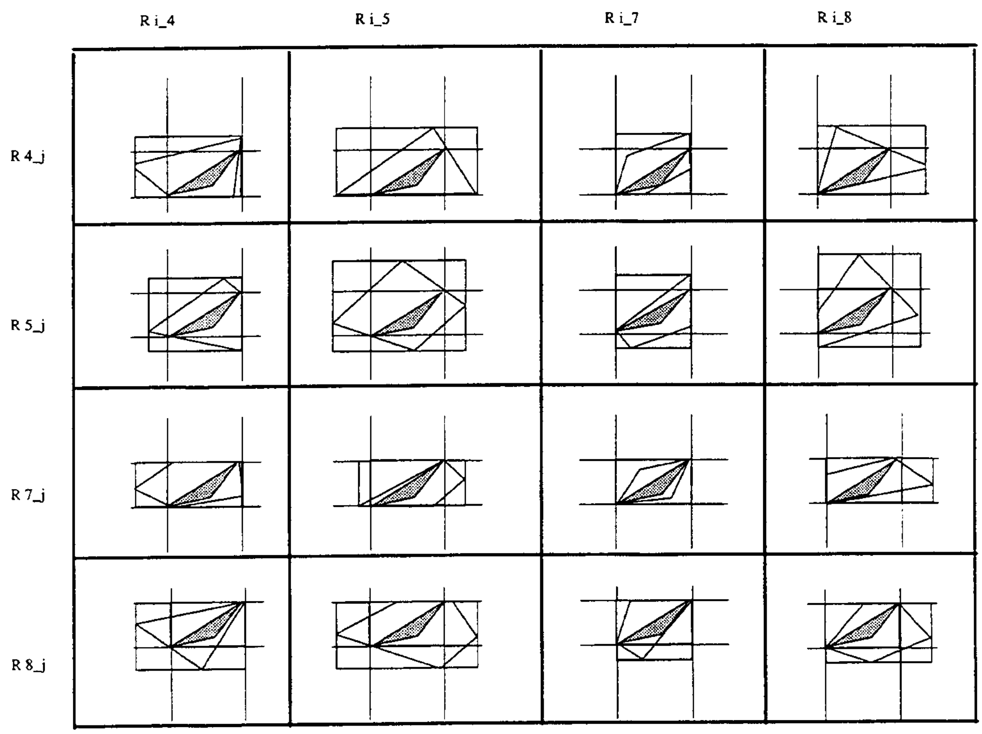



**Fig. 9** Configurations of MBRs that yield the relation *covers* between actual objects

The remaining relations involve the retrieval of a large number of MBRs. In the case of *disjoint*, for instance, all the MBRs may enclose objects that are *disjoint* with the reference object, except for those that satisfy the relation $R_{i\text{-}j}$ where i, in {4,5,7,8} and j in {6,7,9,10}, or i in {6,7,9,10} and j in {4,5,7,8}. Figure 10 illustrates all configurations of MBRs that enclose objects potentially *disjoint* with the reference object.

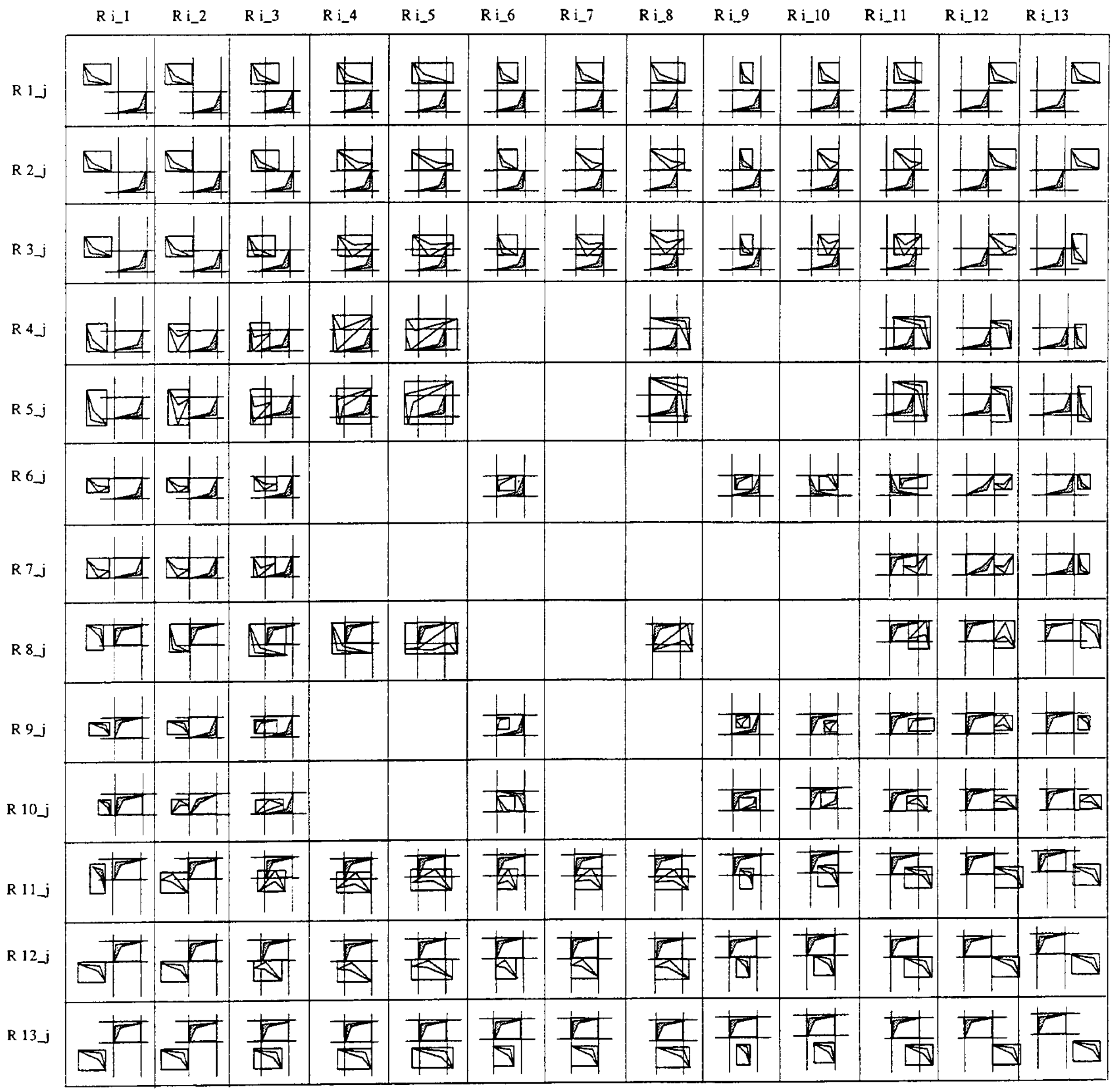



**Fig. 10** Configurations of MBRs that yield the relation *disjoint* between actual objects

The blank configurations in Figure 10 are not to be retrieved when we deal with contiguous objects. If, for instance, the MBRs are related by the relation $R_{5\_9}$, then the MBRs *overlap* and the actual objects also necessarily *overlap*. Figure 11 illustrates such a configuration; in this case a refinement step is not needed.

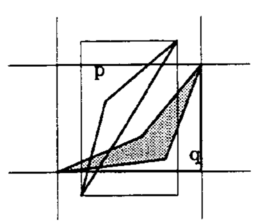
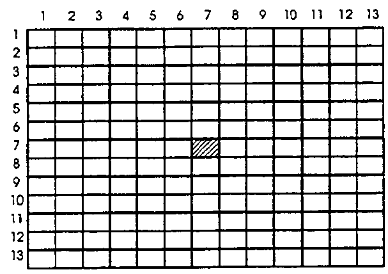
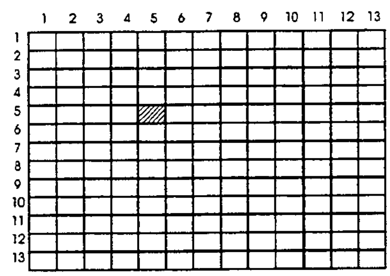
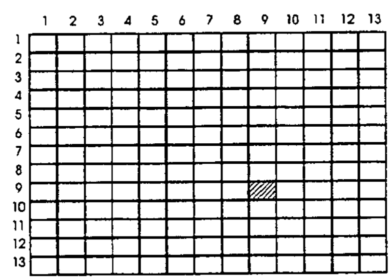
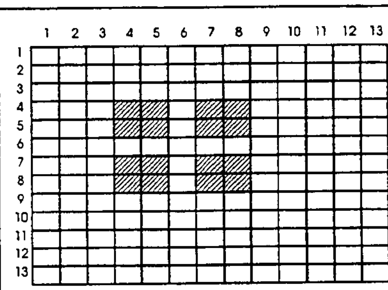



**Fig. 11** A configuration of overlapping MBRs for which the actual objects necessarily *overlap*

We followed the same procedure for the relations *meet* and *overlap*. Table 1 shows the conclusions that can drawn if we use the concept of projections to study topological relations. The first column of Table 1 illustrates the topological relation that the retrieved objects satisfy with respect to the reference object. The second column contains the subset of the 169 configurations that correspond to objects which could satisfy the relation of column one. The third column illustrates the configurations of the second column graphically.

| **Relation between actual objects p and q** | **Projection relations that the MBRs p' to be retrieved should satisfy with respect to MBR q'** | **Illustration of the corresponding projections** |
|---|---|---|
| equal(p,q) | $R_{7\_7}(p',q')$ |  |
| contains(p,q) | $R_{5\_5}(p',q')$ |  |
| inside(p,q) | $R_{9\_9}(p',q')$ |  |
| covers(p,q) | $R_{i\_j}(p',q')$ where i,j in {4,5,7,8} |  |

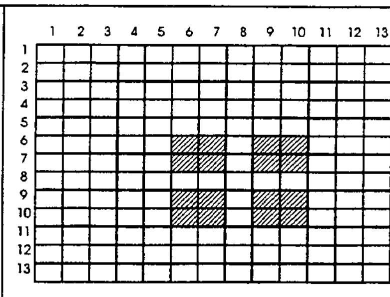
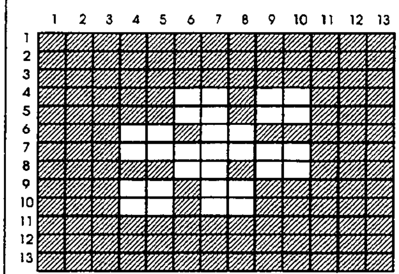
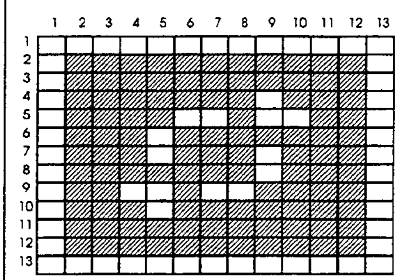
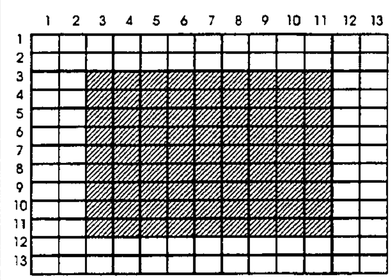
| | | |
|---|---|---|
| covered_by(p,q) | $R_{i\_j}(p',q')$ where i,j in {6,7,9,10} |  |
| disjoint(p,q) | any relation except $R_{i\_j}(p',q')$ where i in {4,5,7,8}, j in {6,7,9,10} or i in {6,7,9,10}, j in {4,5,7,8} |  |
| meet(p,q) | any relation except $R_{i\_j}(p',q')$ where i in {1,13} or j in {1,13} or (i,j) in {(4,9), (5,6), (5,7), (5,9), (5,10), (6,5), (7,5), (7,9), (8,9), (9,4), (9,5), (9,7), (9,8), (10,5)} |  |
| overlap(p,q) | any relation except $R_{i\_j}(p',q')$ where i in {1,2,12,13} or j in {1,2,12,13} |  |

**Table 1** Topological relations implemented

For some cases the refinement step can be avoided, that is, the relation between MBRs unambiguously determines the topological relation between the actual objects (e.g., the configuration of Figure 11). Figure 12 illustrates the configurations for which the refinement step is not needed (this happens only in the cases of *disjoint* and *overlap*).

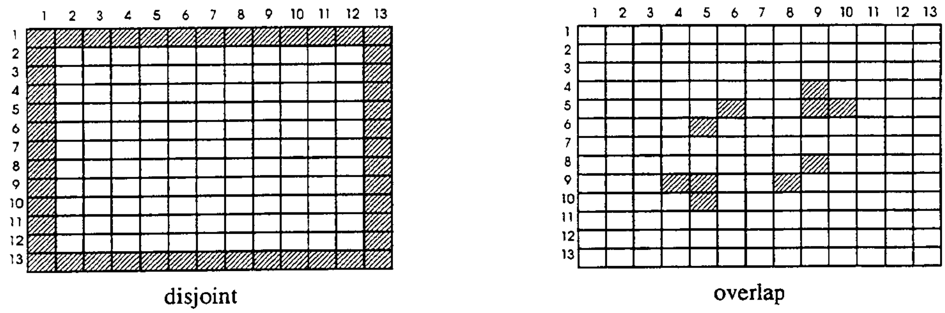



**Fig. 12** Configurations for which a refinement step is not needed

Clementini et al. (1994) studied the use of MBR approximations in query processing involving topological relations. Furthermore, they defined a minimal subset of the nine intersections that can optimally determine the relation between the actual objects taking in account the frequency of the relations. Their findings can be used during the retrieval step in order to minimise the cost for the computation of intersections between potential candidates. In the next section we apply the results of sections 3 and 4 in actual implementations based on R-trees.

## 5. IMPLEMENTATION OF TOPOLOGICAL RELATIONS IN R-TREES

In order to retrieve the topological relations of $m_{12}$ using R-trees one needs to define more general relations that will be used for propagation in the intermediate nodes of the tree structure. For instance, the intermediate nodes P that could enclose MBRs p' that *meet* the MBR q' of the reference object q, should satisfy the more general constraint *meet(P,q')* ∨ *overlap(P,q')* ∨ *covers(P,q')* ∨ *contains(P,q')*. Figure 13 illustrates examples of such configurations.

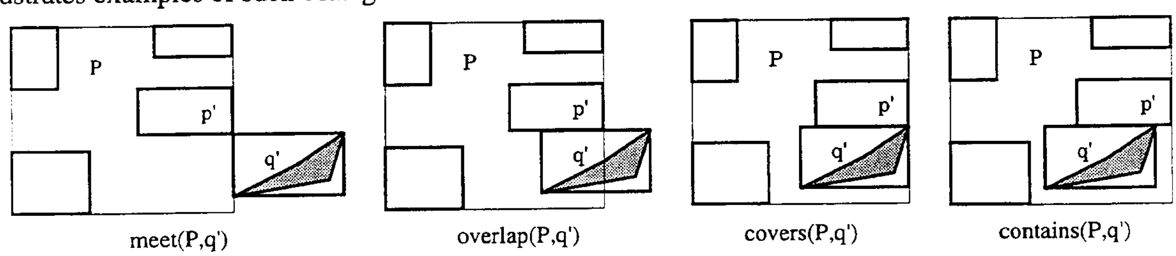



**Fig. 13** Intermediate nodes that may contain MBRs that *meet* the MBR of the reference object

Following this strategy, the search space is pruned by excluding the intermediate nodes P that do not satisfy the previous constraint. Table 2 presents the relations that should be satisfied between an intermediate node P and the MBR q' of the reference object, so that the node will be selected for propagation.

| Relation between MBRs p' and q' | Relation between intermediate node P (that may enclose MBRs p') and reference MBR q' |
|---|---|
| equal(p',q') | equal(P,q') ∨ covers(P,q') ∨ contains(P,q') |
| contains(p',q') | contains(P,q') |
| inside(p',q') | overlap(P,q') ∨ covered_by(P,q') ∨ inside(P,q') ∨ equal(P,q') ∨ covers(P,q') ∨ contains(P,q') |
| covers(p',q') | covers(P,q') ∨ contains(P,q') |
| covered_by(p',q') | overlap(P,q') ∨ covered_by(P,q') ∨ equal(P,q') ∨ covers(P,q') ∨ contains(P,q') |
| disjoint(p',q') | disjoint(P,q') ∨ meet(P,q') ∨ overlap(P,q') ∨ covers(P,q') ∨ contains(P,q') |
| meet(p',q') | meet(P,q') ∨ overlap(P,q') ∨ covers(P,q') ∨ contains(P,q') |
| overlap(p',q') | overlap(P,q') ∨ covers(P,q') ∨ contains(P,q') |

**Table 2** Relations for the intermediate nodes

Notice that the same relation between intermediate nodes and the reference MBR exists for all the levels of the tree structure. For instance, the intermediate nodes that could enclose other intermediate nodes P that satisfy the general constraint *meet(P,q')* ∨ *overlap(P,q')* ∨ *covers(P,q')* ∨ *contains (P,q')* should also satisfy the same constraint. This conclusion can be easily extracted from the above table and is applicable to all the topological relations of Table 2.

Summarising, the processing of a query of the form "find all objects p that satisfy a given topological relation with respect to object q" in R-tree-based data structures involves the following steps:

1. Compute the MBRs p' that could enclose objects that satisfy the query. This procedure involves Table 1.
2. Find the topological relations that the MBRs p' of the first step may satisfy with respect to q'. This procedure involves Figure 6.
3. Starting from the top node, exclude the intermediate nodes P which could not enclose MBRs that satisfy the topological relations of the second step and recursively search the remaining nodes. This procedure involves Table 2.
4. Follow a refinement step for the retrieved MBRs, except for the cases illustrated in Figure 12.

In order to experimentally quantify the performance of the above algorithm, we created tree structures by inserting 10000 MBRs randomly generated. We tested three data files:

- the first file contains *small* MBRs: the size of each rectangle is at most 0,02% of the global area
- the second file contains *medium* MBRs: the size of each rectangle is at most 0,1% of the global area
- the third file contains *large* MBRs: the size of each rectangle is at most 0,5% of the global area.

The search procedure used a search file for each data file containing 100 rectangles, also randomly generated, with similar size properties as the data rectangles. Table 3 illustrates the number of hits per search, that is, the number of retrieved MBRs for the three files. The number of hits per search is inversely proportional to the selectivity of the relation.

| data size | number of hits per search | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | disjoint | meet | overlap | covered_by | inside | equal | covers | contains |
| *small* | 9999 | 3,20 | 2,76 | 1,06 | 0,03 | 1,00 | 1,07 | 0,02 |
| *medium* | 9998 | 11,15 | 10,37 | 1,30 | 0,21 | 1,00 | 1,26 | 0,17 |
| *large* | 9996 | 53,94 | 53,56 | 2,75 | 1,47 | 1,00 | 2,52 | 1,25 |

**Table 3** Retrieved MBRs per relation for each data file

*Disjoint* is the least selective relation since it involves the larger number of output MBRs. Notice that the sum of retrieved MBRs in each row is larger than the total number of MBRs in the database because the same MBR may be retrieved for more than one topological relations between two objects; the refinement process that filters out the inappropriate MBRs is beyond the scope of this paper. We used the previous data files for retrieval of topological relations in R-, R+- and R*-trees. In the implementation of R-trees we selected the quadratic-split algorithm and we set the minimum node capacity to m=40%; in the implementation of R*-trees we set m=40% while in the implementation of R+-trees the "minimum number of rectangle splits" was selected to be the cost function. These settings seem to be the most efficient ones for each method (Beckmann et al., 1990; Sellis et al., 1987).The results are illustrated in Table 4.

| data structure | data size | disk accesses per search | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | disjoint (d) | meet (m) | overlap (o) | covered_by (cb) | inside (i) | equal (e) | covers (cv) | contains (cn) |
| R-trees | *small* | 296 | 4,13 | 4,04 | 4,04 | 4,04 | 3,58 | 3,58 | 3,42 |
| | *medium* | 297 | 5,35 | 5,26 | 5,26 | 5,26 | 3,92 | 3,92 | 3,75 |
| | *large* | 298 | 9,16 | 8,99 | 8,99 | 8,99 | 4,31 | 4,31 | 4,13 |
| R+-trees | *small* | 638 | 3,39 | 3,28 | 3,28 | 3,28 | 2,81 | 2,81 | 2,75 |
| | *medium* | 1176 | 4,47 | 4,19 | 4,19 | 4,19 | 2,39 | 2,39 | 2,28 |
| | *large* | 5373 | 21,39 | 19,13 | 19,13 | 19,13 | 2,33 | 2,33 | 2,25 |
| R*-trees | *small* | 304 | 3,65 | 3,57 | 3,57 | 3,57 | 3,13 | 3,13 | 2,91 |
| | *medium* | 296 | 4,70 | 4,60 | 4,60 | 4,60 | 3,53 | 3,53 | 3,32 |
| | *large* | 293 | 8,52 | 8,24 | 8,24 | 8,24 | 3,87 | 3,87 | 3,63 |

**Table 4** Results of tests on topological relations

With the exception of *disjoint*, the improvement of the retrieval using R-trees compared to serial retrieval is immense; especially for small size MBRs the difference is almost two orders of magnitude[4]. The difference drops as the MBRs become larger. The increase in the size, increases the density and, therefore, the possibility that the reference object is not *disjoint* with other MBRs or intermediate nodes. The retrieval of *disjoint* is, as expected, worse than serial retrieval because this relation requires the retrieval of all the nodes of the tree structure as derived from the 4-step strategy presented above. Clearly, a "real" system would do a serial retrieval in such a case instead of using the tree structure.

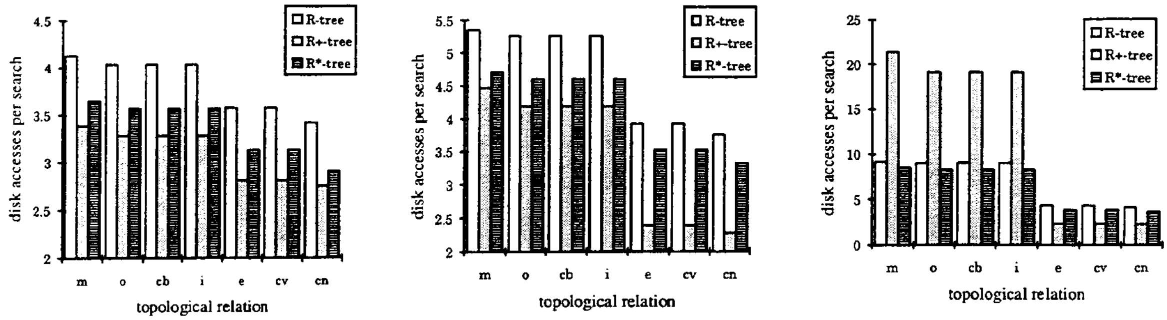
Notice that the retrieval time does not only depend on the number of the MBR configurations to be retrieved, but also on the intermediate nodes. The relation *inside* is more expensive than *covers*, although it

[4]The number of disk accesses per search using serial retrieval is equal to 200 for all relations because the size of the data file is equal to 10000 entries and the page capacity is equal to 50 entries.

retrieves only one output MBR configuration (relation $R_{9\text{-}9}$ with respect to q'). This is due to the large number of intermediate nodes that could contain MBRs that satisfy the relation $R_{9\text{-}9}$.

According to Table 4, the relations of $m_{t2}$ can be grouped in three categories with respect to the cost of retrieval. The first group contains *disjoint* which is the most expensive relation and should be processed by serial search. The second group consists of *meet*, *overlap*, *inside* and *covered_by*. The third group consists of the three relations (*equal*, *covers*, *contains*) which are the least expensive to process. The cost difference between the second and the third group increases as the size of the MBRs becomes larger.

The comparison of the various R-tree-based structures follows, in general, the conclusions drawn in the literature i.e., the variations $R^+$-trees and $R^*$-trees outperform the original R-trees[5]. Figure 14 illustrates a graphical representation of the performance of each method, for small, medium and large data size respectively.




Small data size

Medium data size

Large data size

**Fig. 14** Performance comparison of R-tree variants

According to Figure 14, $R^+$-trees perform slightly better than $R^*$-trees. However this advantage is lost if the duplicate entries generate one extra level in the tree structure, as happened for the large data size of our tests. In such case the performance of $R^+$-trees is inferior for the most expensive relations but remains competent for the least expensive ones. The irregular behaviour of $R^+$-trees is due to the lack of overlap between nodes, which results to the quick exclusion of the majority of the intermediate nodes when one of the least expensive relations is retrieved. On the other hand, for the expensive relations there is no significant gain and the performance is worse compared to the other structures because of the great number of nodes in the tree.

In section 5 we have studied queries that involve the retrieval of objects that satisfy a topological relation with respect to a reference object. In the next section we extend our results for queries that involve complex spatial conditions in the form of disjunctions and conjunctions with respect to one and two reference objects.

[5] When the data density becomes high it is possible that all of the entries in a full node coincide on the same point of the plane. In such cases $R^+$ trees do not work (Greene, 1989). This happened for some data sets involving large MBRs during our tests.

## 6. COMPLEX QUERIES

In some cases the refinement provided by the relations of $m_{t2}$ is not needed. In a cadastral application, for instance, the difference between *inside* and *covered_by* may not be important. Consider the query "find all land parcels *in* a given area". The land parcels of the result should be *inside* or should be *covered_by* the area, that is, the interpretation of *in* is *inside* $\vee$ *covered_by*. We can define topological relations of lower qualitative resolution using disjunctions of the relations of $m_{t2}$. This is a common strategy in qualitative spatial knowledge representation; it has been used for direction relations in (Papadias and Sellis, 1994a) and for the above topological relations in (Randell et al., 1992).

Queries involving low resolution relations can also be processed by the 4-step method of section 5. The only difference is that the set of the MBRs to be retrieved at the first step is the union of the MBRs to be retrieved by each of the relations that belong to the disjunction. Furthermore, in some cases the retrieval times do not change. In the previous query (relation *in*), the MBRs of the result are the same that would be retrieved if the relation of interest were *covered_by*. This is because the MBRs to be retrieved for *inside* constitute a subset of the MBRs for *covered_by* (see Table 1). Figure 15 illustrates the subset relations with respect to the output MBRs.

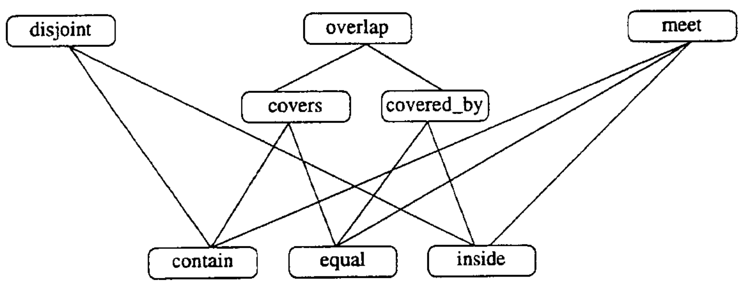



**Fig. 15** Subset relations according to the output MBRs

According to Figure 15 the retrieval of the query "find all objects that satisfy the relation *meet* or *contain* or *equal* or *inside* q" involves the same retrieval time with the query "find all objects that *meet* q". On the other hand, queries that involve conjunctions of topological relations with respect to one reference object (e.g., find all objects that are *inside* and *covered_by* q) have an empty result because the relations of $m_{t2}$ are pairwise disjoint. Nevertheless we can perform semantic query optimisation for queries of the form "find all objects that are *inside* $q_1$ and *overlap* with $q_2$". We can determine that the result of this query is empty if we know that $q_1$ and $q_2$ are *disjoint*. As Figure 16 illustrates, if p is *inside* $q_1$ and $q_1$ *disjoint* $q_2$, it cannot be the case that p *overlaps* $q_2$.

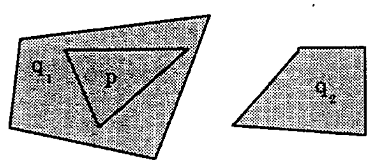



**Fig. 16** A query with two reference objects and empty result

Table 5 illustrates the relations between reference objects for which an empty result is returned without running the query. When a query involving two reference objects $q_1$ and $q_2$ is given, the topological relation between the reference objects is examined, and if it is one of the relations of the array, the output is empty[6]. For the above query, in addition to *disjoint*, if $q_1$ and $q_2$ are related by *meet, equal, inside* or *covered_by*, the result is also empty.

| | **disjoint(p,q₂)** | **meet(p,q₂)** | **equal(p,q₂)** | **inside(p,q₂)** | **cvrdby(p,q₂)** | **contain(p,q₂)** | **covers(p,q₂)** | **overlap(p,q₂)** |
|---|---|---|---|---|---|---|---|---|
| **disjoint(p,q₁)** | --- | e ∨ ct ∨ cv | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | e ∨ ct ∨ cv | e ∨ ct ∨ cv | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | e ∨ ct ∨ cv |
| **meet(p,q₁)** | e ∨ i ∨ cb | i ∨ ct | d ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ ct ∨ cv | d ∨ e ∨ ct ∨ cv | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | e ∨ ct ∨ cv ∨ o |
| **equal(p,q₁)** | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | d ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ i ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ i ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ i ∨ cb ∨ cv ∨ o | d ∨ m ∨ e ∨ i ∨ cb ∨ ct ∨ o | d ∨ m ∨ e ∨ i ∨ cb ∨ ct ∨ cv |
| **inside(p,q₁)** | e ∨ i ∨ cb | d ∨ m ∨ e ∨ i ∨ cb | d ∨ m ∨ e ∨ i ∨ cb ∨ cv ∨ o | d ∨ m | d ∨ m ∨ e ∨ i ∨ cb | d ∨ m ∨ e ∨ i ∨ cb ∨ cv ∨ o | d ∨ m ∨ e ∨ i ∨ cb ∨ cv ∨ o | d ∨ m ∨ e ∨ i ∨ cb |
| **cvrdby(p,q₁)** | e ∨ i ∨ cb | d ∨ e ∨ i ∨ cb | d ∨ m ∨ e ∨ i ∨ cb ∨ ct ∨ o | d ∨ m ∨ e ∨ ct ∨ cv | d ∨ m ∨ i ∨ ct | d ∨ m ∨ e ∨ i ∨ cb ∨ cv ∨ o | d ∨ m ∨ e ∨ i ∨ cb ∨ o | d ∨ m ∨ e ∨ i ∨ cb |
| **contain(p,q₁)** | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ cb ∨ ct ∨ cv ∨ o | --- | e ∨ ct ∨ cv | e ∨ ct ∨ cv |
| **covers(p,q₁)** | m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | e ∨ i ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ i∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ cb ∨ ct ∨ cv ∨ o | d ∨ m ∨ e ∨ ct ∨ cv ∨ o | e ∨ i ∨ cb | i ∨ ct | e ∨ ct ∨ cv |
| **overlap(p,q₁)** | e ∨ i ∨ cb | e ∨ i ∨ cb | d ∨ m ∨ e ∨ i ∨ cb ∨ ct ∨ cv | d ∨ m ∨ e ∨ ct ∨ cv | d ∨ m ∨ e ∨ ct ∨ cv | e ∨ i ∨ cb | e ∨ i ∨ cb | --- |

**Table 5** Conjunctions of relations that yield empty results

[6] Each entry at row $r_i(p,q_1)$ and column $r_j(p,q_2)$ (where $r_i$ and $r_j$ are relations of $m_{t2}$) is the complement of the composition relation $r'_i(q_1,p)$ and $r_j(p,q_2)$ with respect to $m_{t2}$, (where $r'_i$ is the converse relation of $r_i$). For an extensive discussion about composition of topological relations see (Egenhofer, 1991).

If the relation between the reference objects is not one of the relations of the array, then one of the two relations is retrieved using R-trees and the qualifying MBRs p' are filtered with respect to the other reference object. This process can be performed in main memory by checking the relation that each output MBR of the first step satisfies with respect to the second reference object. Therefore the number of disk accesses depends only on the first relation. The selection of the first relation is based on the size of the reference MBR and on the cost group that the relation belongs. The relations *covers*, *contains* and *equal* are preferable because they require the least disk accesses. If the sizes of the reference MBR have a considerable difference, then the smallest reference MBR must be selected because the cost of retrieval is proportional to the data size (see Figure 14).

Although the above methods provide a good foundation for practical applications, the underlying theoretical assumptions do not always hold in practical (GIS) environments. In the next section we discuss how our work can be extended to capture real-world imprecision.

## 7. NON-CRISP MINIMUM BOUNDING RECTANGLES

The use of MBR filters for processing topological relations has a very delicate aspect, as it relies heavily on the fact that the MBRs are crisp representations for the objects. A *crisp* MBR has to satisfy two constraints:

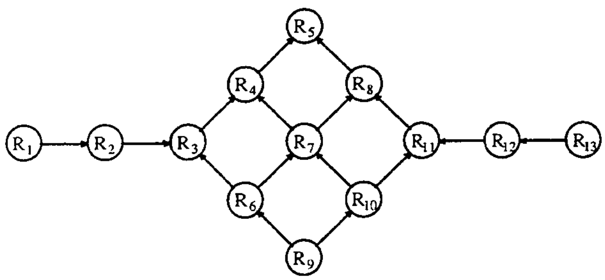
1. the object to be approximated is *fully* contained within the rectangle and
2. each boundary of the MBR *coincides* with some part of the object's boundary.

One may argue that a violation of either condition would not constitute an MBR in its literal sense (the rectangle would not be bounding, or the rectangle would not be minimal); however, even with careful implementations of algorithms that create MBRs for spatial objects, these constraints can be regularly violated.

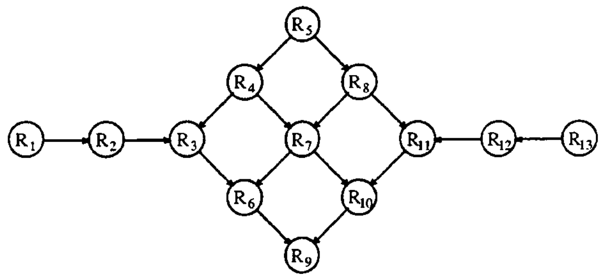
This problem is due to the difficulties of implementing coordinate-based computer algorithms for geometry that preserve topology. One cannot always *reliably* derive from a coordinate representation topological properties such as the coincidence of two points, or whether a point close to a line is actually on the line etc. While the first constraint in the definition of crisp MBRs can usually be fulfilled, e.g., by making the MBRs slightly larger than required, it is the second constraint—the minimality of rectangles—that introduces the problem. Actually, the remedy to constraint 1 is often the fate of constraint 2.

Sometimes precision is traded for performance and fast, but slightly inaccurate, algorithms may be used to calculate MBRs. Such algorithms make sure that constraint 1 is fulfilled, but occasionally violate condition 2 so that some MBRs may actually be larger than necessary. The same may happen due to numerical inaccuracies such as rounding errors, particularly, if floating point arithmetic is used for representing the objects' coordinates and MBRs are expressed by two integer pairs. Therefore, often in implementations of MBR-based spatial access one has to assume some numerical inaccuracies.

The 4-step algorithm of section 5 can be extended to deal with inaccuracies involving slightly larger than crisp MBRs. Our extension is based on the concept of *conceptual neighbourhood* (Freksa 1992, Egenhofer and Al-Taha 1992). The conceptual neighbourhood of the 13 one-dimensional relations of Figure 4 forms a graph, in which each pair of directly connected nodes corresponds to a pair of relations that are conceptual neighbours. Given an initial relation between two objects, if we continuously enlarge the primary object, we follow the path indicated by the arrows in Figure 17a. For instance, if the relation between the objects is $R_1$, then extending the primary object, while keeping the reference object constant, gradually leads to relations $R_2$, $R_3$, $R_4$ and $R_5$ (see Figure 4). Likewise, enlarging the reference object changes the relation between the two objects according to the directions of the edges in Figure 17b.




Relation transitions when the primary object is extended



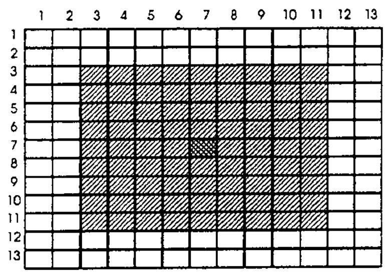

Relation transitions when the reference object is extended

**Fig. 17** Conceptual neighbourhoods for relations in 1d space

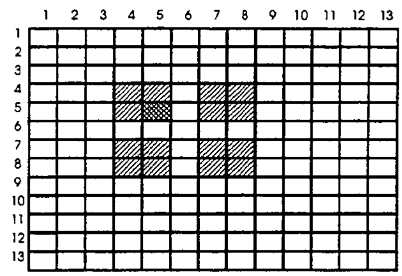
A *first-degree conceptual neighbour* of a relation $R_i$ is a relation $R_j$ that can be reached from $R_i$ via a directed edge in either neighbourhood graph. For example, relation 7 has hour first-degree conceptual neighbours (relations 4 and 8 if we enlarge the primary object, and relations 6 and 10 if we enlarge the reference object). On the other hand, relation 13 has one first-degree conceptual neighbour, relation 12 which is obtained by enlarging either object.

A *second-degree conceptual neighbour* of $R_i$ is a relation that has at least two first-degree conceptual neighbours that are also first-degree neighbours of $R_i$. For example, the second-degree conceptual neighbours of relation 7 comprises the set of relations 3, 5, 7, 9, and 11—they all have at least two common first-degree neighbours with 7. On the other hand, relation 2 does not have any second-degree neighbours. These results can be easily extended to 2d space where two relations are k-neighbours if they are k-neighbours in any dimension.

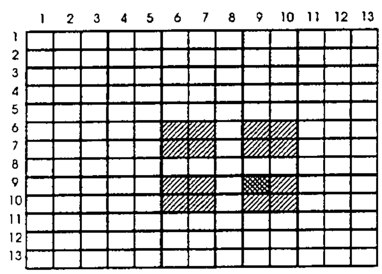
We use the concept of conceptual neighbourhood to overcome the potential problems that may arise from inaccuracy. In addition to the configurations of Table 1, the candidate MBRs for the first step may also satisfy the first- and second-degree conceptual neighbour relations with respect to the crisp MBRs that would be retrieved. Table 6 should be used instead of Table 1 in such cases. The dark grey rectangles of Table 6 correspond to the crisp MBR relations displayed in Table 1. The light grey rectangles are the additional ones that correspond to first and second neighbourhood relations.



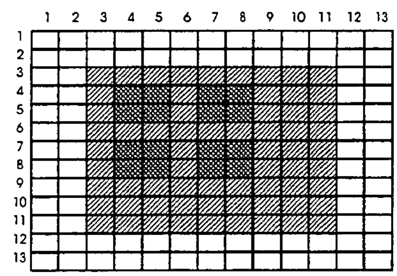
equal



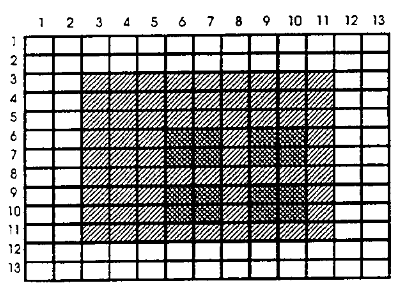
contains



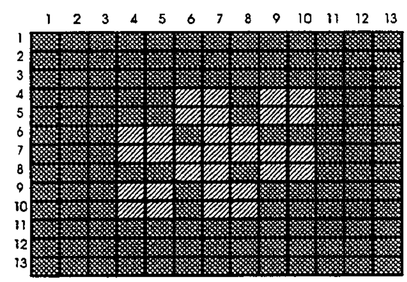
inside



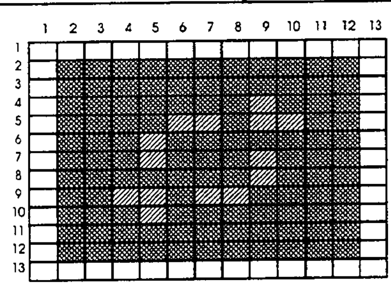
covers



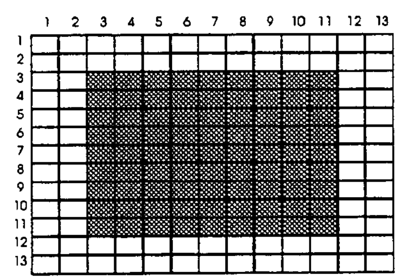
covered_by



disjoint



meet



overlap

**Table 6** Retrieval using 2-neighbours

The extra MBRs to be retrieved increase the retrieval times, but assure that all potential objects that satisfy the query are retrieved even if there exists inaccuracy that can result in 2-degree relation deformation. The largest increase in output MBRs is observed for the relation *equal*, for which 81 MBR relations have to be considered (in lieu of 1 MBR relation in the crisp case). On the other hand, the output MBRs for the relation *overlap* remain constant.

## 8. CONCLUSION

The representation and processing of topological relations is an important topic in a number of areas including Spatial Query Languages, Image and Multimedia Databases and Geographic Information Systems. Despite the attention that topological relations have attracted in application domains, they have not been extensively applied in spatial data structures. In this paper we focus on the retrieval of topological relations in MBR-based data structures. In particular we have shown how the topological relations *disjoint, meet, equal, overlap, contains, inside, covers* and *covered_by* (defined by the 9-intersection model) can be retrieved from R-trees and their variations.

First we illustrated the possible relations between MBRs and we described the corresponding topological relations. Then we studied the topological information that MBRs convey about the actual objects they enclose using the concept of projections. Finally we applied the results in R-tree-based data structures and we concluded that they are suitable for topological relations, with $R^+$- and $R^*$-trees outperforming the original R-trees for most cases. We also investigated queries that involve complex spatial conditions in the form of disjunctions and conjunctions and we demonstrated how our method can be applied for non-crisp MBRs.

Although we have dealt with contiguous regions, practical applications do not necessarily deal with contiguous objects. Geographic entities, such as countries with islands, consist of disconnected components. The previous results can be extended for this case. The only difference is that the number of MBRs to be retrieved for some relations increases since the relaxation of the contiguity constraint qualifies more MBRs as potential candidates. In particular, all MBRs may contain objects *disjoint* with some reference object. For the relation *meet*, the only MBRs that can be excluded are the ones in configurations $R_{i\_j}$ where i or j in {1,13}. Furthermore in case of *overlap*, the refinement step is needed for all MBR configurations (including those of Figure 12).

In order to model linear and point data we need further extensions because the topological relations that can be defined, as well as the number of possible projection relations between MBRs, depend on the type of objects. Egenhofer (1993), for instance, defined 33 relations between lines based on the 9-intersection model, while Papadias and Sellis (1994a) have shown that the number of different projections between a region reference object and a line primary object is 221. The ideas of the paper can be extended to include linear and point data, objects with holes etc.

## REFERENCES

Allen, J.F. (1983) Maintaining Knowledge about Temporal Intervals. Communications of ACM, Vol 26(11), pp. 832-843.

Beckmann, N., Kriegel, H.P. Schneider, R., Seeger, B. (1990) The R*-tree: an Efficient and Robust Access Method for Points and Rectangles. In the Proceedings of ACM-SIGMOD Conference.

Brinkhoff, T., Kriegel, H.P, Schneider, R. (1993a) Comparison of Approximations of Complex Objects used for Approximation-based Query Processing in Spatial Database Systems. In the Proceedings of 9th International Conference on Data Engineering.

Brinkhoff, T., Horn, H., Kriegel, H.P., Schneider, R. (1993b) A Storage and Access Architecture for Efficient Query Processing in Spatial Database Systems. In the Proceedings of the Third Symposium on Large Spatial Databases (SSD). Springer Verlag LNCS.

Brinkhoff, T., Kriegel, H.P., Schneider, R., Seeger, B. (1994) Multi-Step Processing of Spatial Joins. In the Proceedings of ACM-SIGMOD Conference.

Clementini, E., Sharma, J., Egenhofer, M. (1994) Modeling Topological Spatial Relations: Strategies for Query Processing. To appear in the International Journal of Computer and Graphics.

Egenhofer, M. (1991) Reasoning about Binary Topological Relations. In the Proceedings of the Second Symposium on the Design and Implementation of Large Spatial Databases (SSD). Springer Verlag LNCS.

Egenhofer, M. (1993) Definition of Line-Line Relations for Geographic Databases. Data Engineering, Vol 16(6), pp. 40-45.

Egenhofer, M. (1994) Spatial SQL: A Query and Presentation Language. IEEE Transactions on Knowledge and Data Engineering, Vol 6(1), pp. 161-174.

Egenhofer, M., Franzosa R. (1991) Point-Set Topological Spatial Relations. International Journal of Geographic Information Systems, Vol 5(2), pp. 160-174.

Egenhofer, M., Al-Taha, K. (1992) Reasoning about Gradual Changes of Topological Relationships. In the Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Springer Verlag LNCS.

Egenhofer, M., Sharma, J. (1993) Assessing the Consistency of Complete and Incomplete Topological Information. Geographical Systems, Vol 1, pp. 47-68.

Frank, A. U. (1992) Qualitative Spatial Reasoning about Distances and Directions in Geographic Space. Journal of Visual Languages and Computing, Vol 3, pp. 343-371.

Frank, A. U. (1994) Qualitative Spatial Reasoning: Cardinal Directions as an Example. To appear in the International Journal of Geographic Information Systems.

Freksa, C. (1992) Temporal Reasoning Based on Semi-Intervals. Artificial Intelligence Vol 54, pp. 199-227.

Glasgow, J.I (1994) Spatial Reasoning in Indeterminate Worlds. In the Proceedings of AAAI Conference.

Glasgow, J.I., Papadias, D. (1992) Computational Imagery. Cognitive Science, Vol 16, pp. 355-394.

Greene, D. (1989) An Implementation and Performance Analysis of Spatial Data Access Methods. In the Proceedings of the 5th International Conference on Data Engineering.

Guttman, A. (1984) R-trees: a Dynamic Index Structure for Spatial Searching. In the Proceedings of ACM-SIGMOD Conference.

Hadzilacos, T., Tryfona, N. (1992) A Model for Expressing Topological Integrity Constraints in Geographic Databases. In the Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Springer Verlag LNCS.

Kainz, W., Egenhofer, M., Greasley, I. (1993) Modelling Spatial Relations and Operations with Partially Ordered Sets. International Journal of Geographic Information Systems, Vol 7(3), pp. 215-229.

Keighan, E. (1993) Managing Spatial Data within the Framework of the Relational Model. TR Oracle Corporation, Canada.

Mark, D., Xia, F. (1994) Determining Spatial Relations between Lines and Regions in Arc/Info using the 9-Intersection Model. In ESRI User Conference.

Mark, D., Egenhofer, M. (1994) Calibrating the Meaning of Spatial Predicates from Natural Language: Line Region Relations. In the Proceedings of the 6th International Symposium on Spatial Data Handling (SDH). Taylor Francis.

MGE (1993) MGE Analyst Reference Manual. Intergraph Corporation.

Papadias, D., Sellis, T. (1993) The Semantics of Relations in 2D Space Using Representative Points: Spatial Indexes. In the Proceedings of the European Conference on Spatial Information Theory (COSIT). Springer Verlag LNCS.

Papadias, D., Sellis, T. (1994a) The Qualitative Representation of Spatial Knowledge in two-dimensional Space. VLDB Journal, Special Issue on Spatial Databases, Vol 4(4), pp. 100-138.

Papadias, D., Sellis, T. (1994b) A Pictorial Query-By-Example Language. To appear in the Journal of Visual Languages and Computing, Special Issue on Visual Query Systems. A short version appears in the Proceedings of the 6th International Symposium on Spatial Data Handling (SDH). Taylor Francis.

Papadias, D., Frank, A.U., Koubarakis, M. (1994a) Constraint-Based Reasoning in Geographic Databases: The Case of Symbolic Arrays. In the Proceedings of the 2nd ICLP Workshop on Deductive Databases.

Papadias, D., Theodoridis, Y., Sellis, T. (1994b) The Retrieval of Direction Relations Using R trees. In the Proceedings of the 5th Conference on Database and Expert Systems Applications (DEXA). Springer Verlag LNCS.

Pullar, D., Egenhofer, M. (1988) Toward Formal Definitions of Topological Relations among Spatial Objects. In the Proceedings of the 3rd International Symposium on Spatial Data Handling (SDH).

Randell, D. A., Cui, Z., Cohn., A., (1992) A Spatial Logic Based on Regions and Connection. In the Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann.

Sellis, T., Roussopoulos, N., Faloutsos, C. (1987) The $R^+$-tree: A Dynamic Index for Multi-Dimensional Objects. In the Proceedings of the 13th VLDB conference.

Sharma, J., Flewelling, D., Egenhofer, M. (1994) A Qualitative Spatial Reasoner. In the Proceedings of the 6th International Symposium on Spatial Data Handling (SDH). Taylor Francis.

Sistla, P., Yu, C., Haddad, R. (1994) Reasoning about Spatial Relationships in Picture Retrieval Systems. In the Proceedings of the 20th VLDB Conference.

Vieu, L. (1993) A Logical Framework for Reasoning about Space. In the Proceedings of the European Conference on Spatial Information Theory (COSIT). Springer Verlag LNCS.