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
CA 92093-0114, USA
dimitris@cs.ucsd.edu

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

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

Max J. Egenhofer
National Center for Geographic Information and Analysis
University of Maine, Orono
ME 04469-5711, USA
max@mecanl.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 to 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 gained much attention in Spatial Query Languages (Papadias and Sellis, 1995; Egenhofer, 1994), Image and Multimedia Databases (Sistla et al., 1994), Geographic Applications (Frank, 1995), Spatial Reasoning (Randell et al., 1992) and Cognitive Science (Glasgow and Papadias, 1992). 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 relations *disjoint* and *not_disjoint* and on the retrieval of distance information (Roussopoulos et al., 1995). 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* (the term refers to homogeneously 2-dimensional, connected objects with connected boundaries) as defined by the 9-intersection model (Egenhofer, 1991). According to this model, each object p is represented in R_2 space as a point set which has an interior, a boundary and an exterior. The topological relation between any two objects p and q is described by the nine intersections of p's interior, boundary and exterior, with the interior, boundary and 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 converse relation *inside*) and *covers* (and the converse *covered_by*)(see Figure 1).

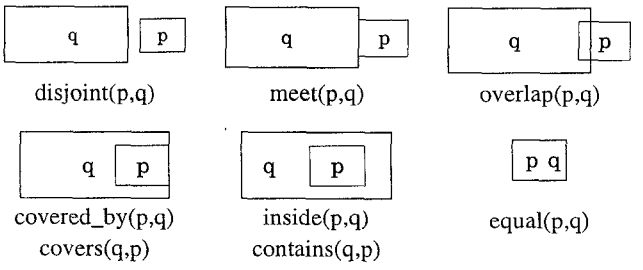


Fig. 1 Topological relations

We call the previous set of topological relations m_{t2} in order to distinguish them from the set m_{t1}={*disjoint, not_disjoint*}. 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*. Sometimes in spatial data structures terminology, the term *overlap* (instead of *not_disjoint*) is used to denote any configuration in which the objects are

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.

Tests with human subjects have shown evidence that the 9-intersection model has potential for defining cognitively meaningful spatial predicates, a fact that makes the above relations a good candidate for commercial systems (Mark and Egenhofer, 1994). In fact, the 9-intersection model has been implemented with 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 MD (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., 1995) and Spatial Reasoning (Grigni et al., 1995).

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 illustrates the possible relations between MBRs and describes the corresponding topological relations. Section 3 investigates the topological information that MBRs convey about the actual objects they enclose with respect to the 9-intersection model. Section 4 applies the results in R-tree-based data structures and compares the retrieval times. Section 5 is concerned with queries that involve complex spatial conditions. Section 6 discusses extensions that deal with imprecision and Section 7 concludes with comments on future work.

2. 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*, *minimum bounding n-corner convexes* etc. 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 term *primary object/MBR* denotes the object/MBR to be located and the term *reference object/MBR* denotes the object in relation to which the primary object/MBR is located (in the examples hereafter, the reference objects q are grey, while the primary objects p are transparent). Let X and Y be functions that give 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 2).

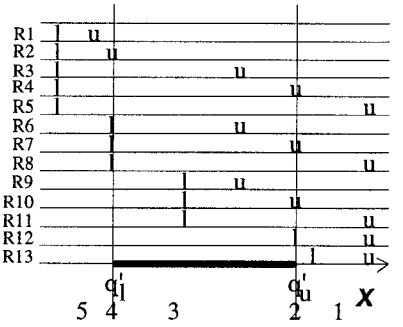


Fig. 2 Possible relations in 1D space

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). The symbols q'_l and q'_u in Figure 2 denote the edge points (lower and upper) for the reference MBR and the characters l and u the lower and the upper points of the primary MBR.

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 3; 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.

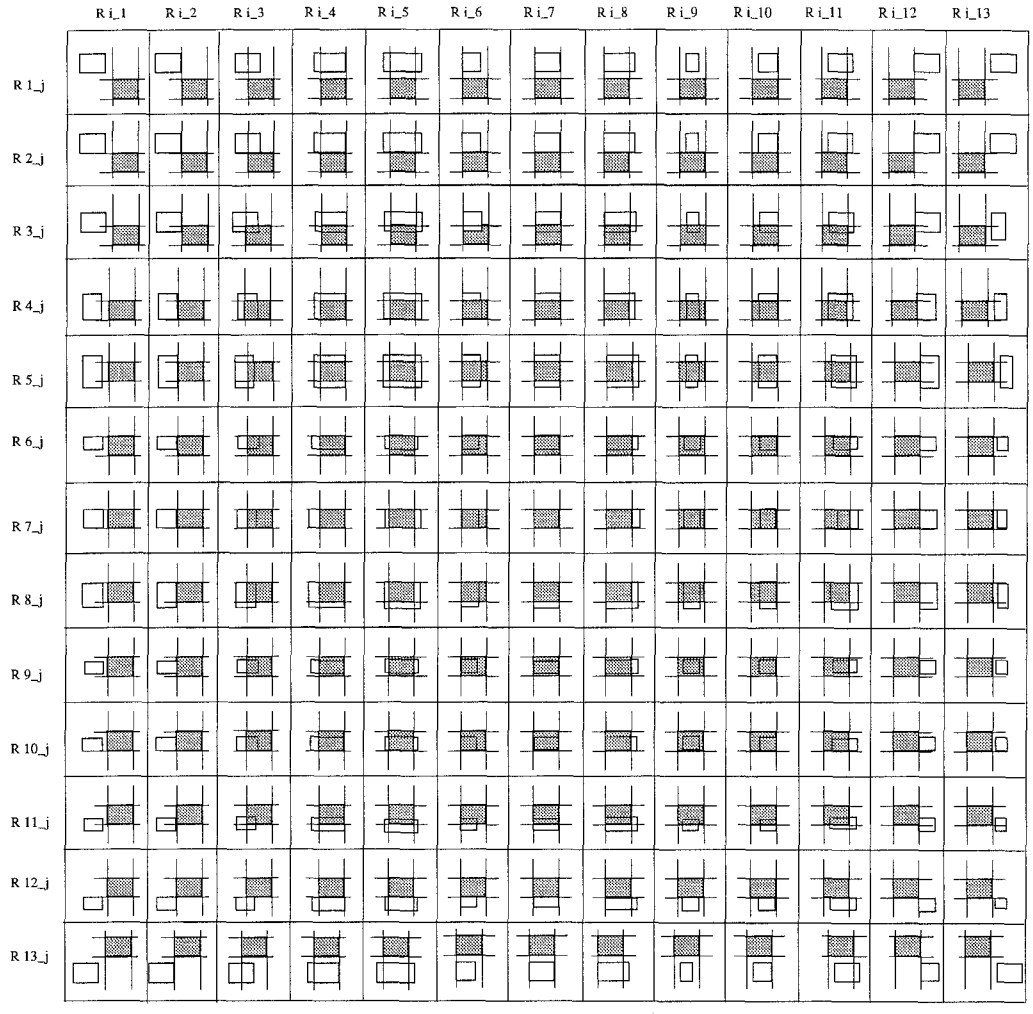


Fig. 3 Possible relations between MBRs

Figure 4 illustrates the corresponding topological relation for the 169 configurations of Figure 3. 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*.

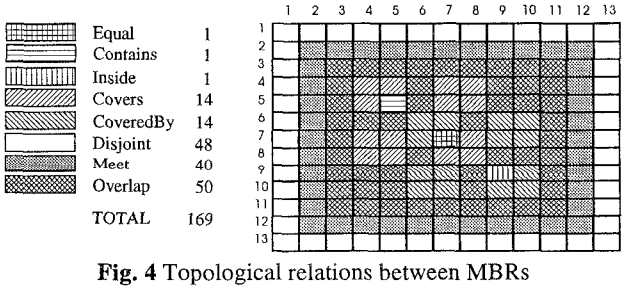


Fig. 4 Topological relations between MBRs

A number of spatial data structures based on MBRs have been developed. The most promising group includes R-trees (Guttman, 1984) and their variations. The R-tree 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}.

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, the 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.

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: First a *filter step* based on MBRs is used to rapidly eliminate MBRs of objects that could not possibly satisfy the query and select a set of potential candidates. Then during a *refinement step* each candidate is examined (by using computational geometry techniques) and false hits are detected and eliminated. Brinkhoff et al. (1994) 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.

The above techniques speed-up the retrieval of relations of m_{t1} using R-trees and variations. The present work builds on the original two-step strategy to explore a larger, and more detailed set of topological relations[1]. 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 THAT MBRS CONVEY ABOUT THE ACTUAL OBJECTS

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. 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* to 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 also enclose objects that satisfy the relations *overlap*, *covered_by*, *covers* or *meet* with respect to q (see Figure 5). 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*.

[1] 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., 1994).

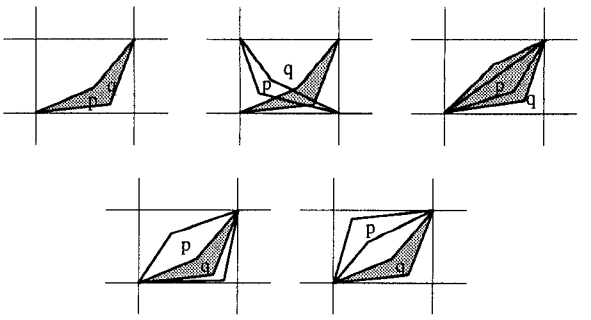


Fig. 5 Possible relations for objects when the MBRs are *equal*

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_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. 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. As Figure 6 illustrates, these MBRs satisfy the projection relations $R_{i\text{-}j}$ where i and j in {4,5,7,8}. 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'.

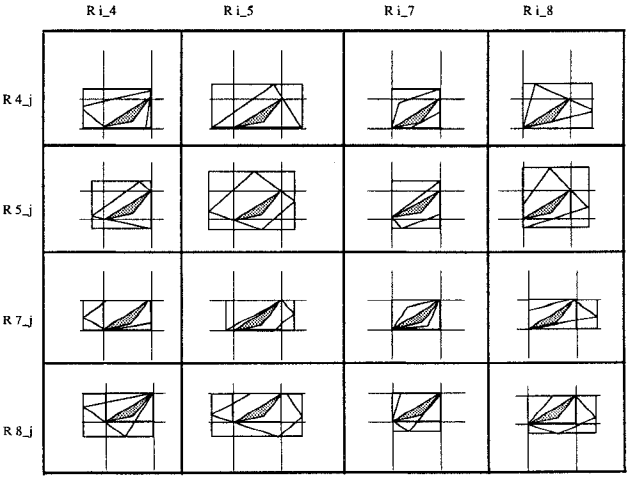


Fig. 6 Configurations of MBRs that yield the relation *covers* between actual objects

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

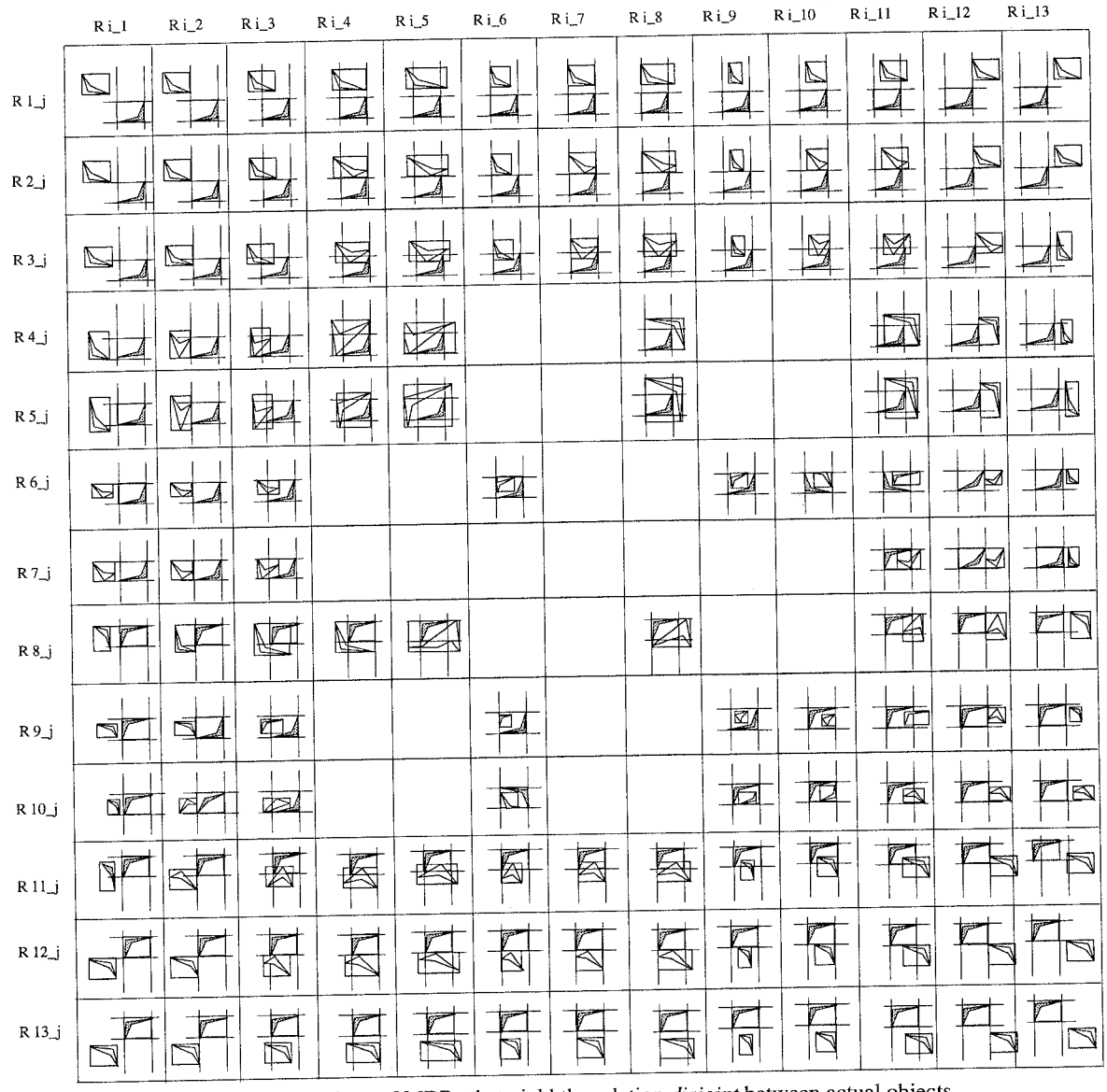


Fig. 7 Configurations of MBRs that yield the relation *disjoint* between actual objects

in{6,7,9,10} and j in {4,5,7,8}. Figure 7 illustrates all configurations of MBRs that enclose objects potentially *disjoint* with the reference object.

The blank configurations in Figure 7 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 8 illustrates such a configuration; in this case a refinement step is not needed.

We followed the same procedure for the relations *meet* and *overlap*. Table 1 shows the conclusions that can be drawn if we use the concept of projections to study topological relations. For each relation, the Table illustrates the subset of the 169 MBR configurations that could contain objects that satisfy the relation.

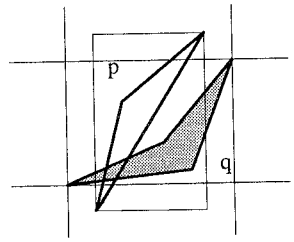


Fig. 8 Overlapping MBRs for which the actual objects necessarily *overlap*

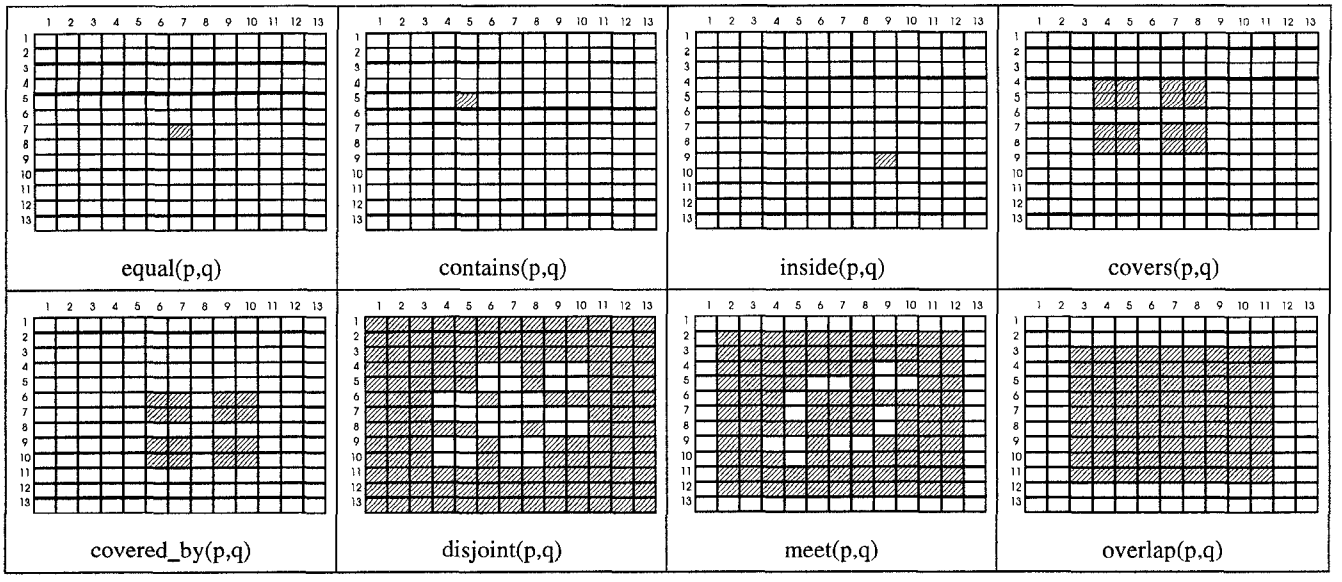


Table 1 Topological relations implemented

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 8). Figure 9 illustrates the configurations for which the refinement step is not needed (this happens only in some of the cases for *disjoint* and *overlap*).

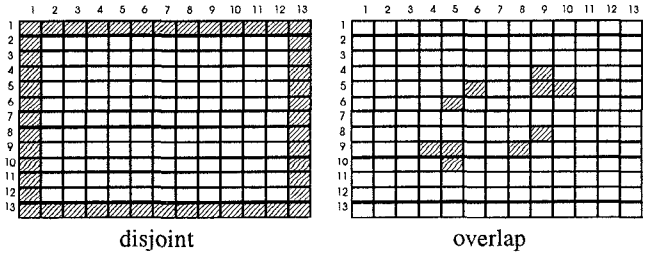


Fig. 9 Configurations for which a refinement step is not needed

Clementini et al. (1995) 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 into account the frequency of the relations. Their findings can be used during the refinement step in order to minimise the cost of the computation of intersections between potential candidates. In the next section we apply the results of sections 2 and 3 to actual implementations based on R-trees.

4. 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 10 illustrates examples of such configurations.

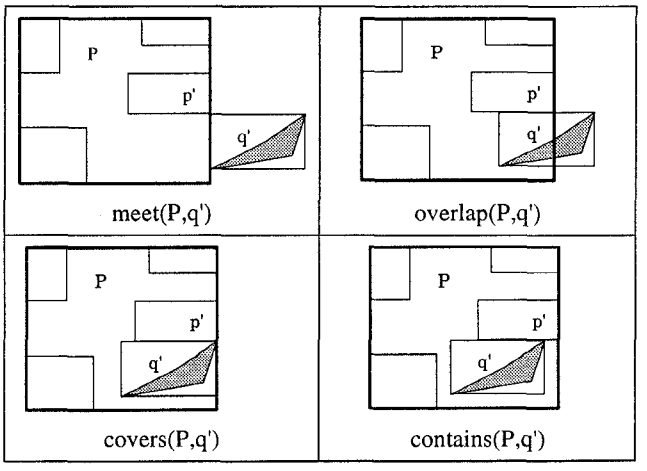


Fig. 10 Intermediate nodes that may contain MBRs that *meet* the MBR of the reference object

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.

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.

Relation between MBRs	Relation between intermediate node P (that may enclose MBRs p') and reference MBR
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

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 mapping 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 4.
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 9.

In order to experimentally quantify the performance of the above algorithm, we created tree structures by inserting 10,000 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. 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).

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 and it is related to the number of disk access. Note that the total number of MBRs in each column is larger than 10,000 because a MBR may be retrieved for more than one relation.

Topological Relation	small MBRs	medium MBRs	large MBRs
disjoint	9,999	9,998	9,996
meet	3.20	11.15	53.94
overlap	2.76	10.37	53.56
covered_by	1.06	1.30	2.75
inside	0.03	0.21	1.47
equal	1.00	1.00	1.00
covers	1.07	1.26	2.52
contains	0.02	0.17	1.25

Table 3 Retrieved MBRs per relation for each data file

The number of disk accesses per search is illustrated graphically in Figure 11. With the exception of *disjoint*, the improvement in the efficiency of retrieval using R-trees compared to serial retrieval is immense; this is particularly true for small size MBRs where the difference is almost two orders of magnitude[2]. 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.

Obviously, topological relations can be retrieved using the traditional window query (i.e., *not_disjoint* relation) used in spatial data structures. In that case, the distinction between the *not_disjoint* relation and the actual topological relation to be retrieved can be made during the refinement step. The cost of this approach is roughly equivalent to the cost of *meet* (according to Table 1, *meet* involves the retrieval of almost all MBRs that are not *disjoint* with the reference MBR). Depending on the relation and the MBR size, our approach can improve the performance by up to 60%. Another important improvement is the reduction of the MBRs that are selected for the refinement step. As illustrated in Table 3, the number of hits per search for some relations (e.g., *inside*, *covers*) is considerably lower than the hits per

[2]The number of disk accesses per search using serial retrieval is equal to 200 for all relations (the size of the data file divided by the page capacity which is equal to 50 entries).

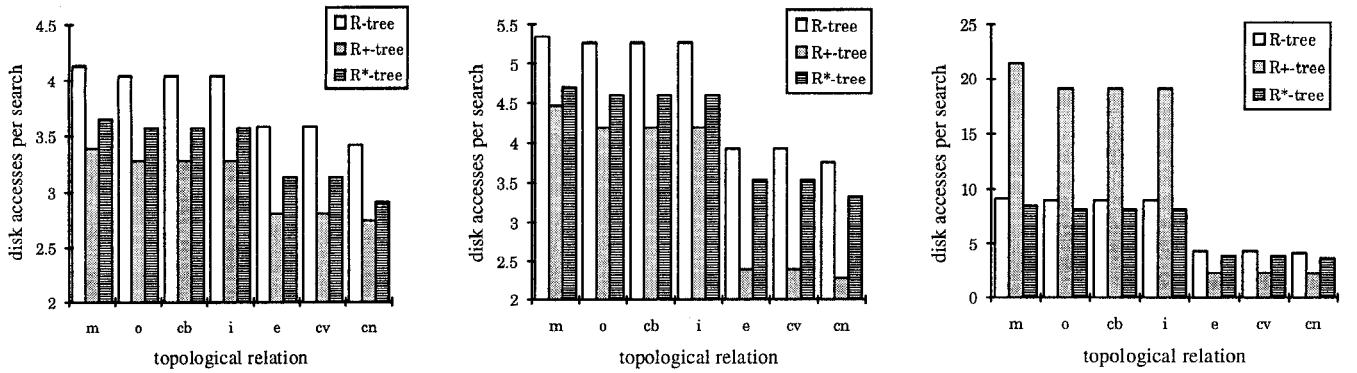


Fig. 11 Performance comparison of R-tree variants

search for *meet* (usually below 10%). Therefore the input of the refinement step is a small subset of the MBRs that would be selected using the window query.

According to Figure 11, the relations of m_{t2} can be grouped into 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. Note also that the primary factor for the retrieval cost is the relation for the intermediate nodes (Table 2). The relation *inside* is more expensive than *covers*, although it 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 $R_{9\text{-}9}$. On the other hand, the cost of *inside* is almost the same as the cost of *covered_by*, which can be inferred by Table 2.

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[3]. For small and medium MBRs, 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 in 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 greater number of nodes in the tree.

In section 4 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. 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 *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, 1994) 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 4. The only difference is that the set of the MBRs to be retrieved by 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 as those 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 12 illustrates the subset relations with respect to the output MBRs.

[3]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.

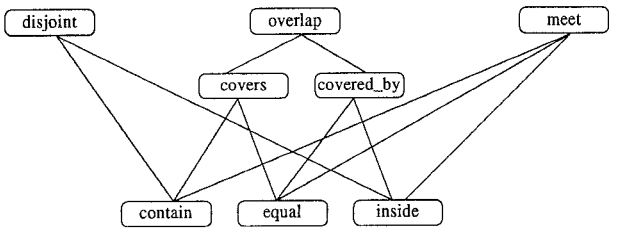


Fig. 12 Subset relations according to the output MBRs

According to Figure 12 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 13 illustrates, if p is *inside* q_1 and q_1 *disjoint* q_2, it cannot be the case that p *overlaps* q_2.

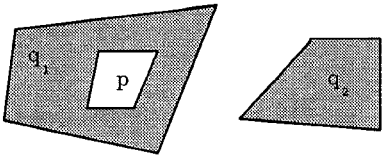


Fig. 13 A query with two reference objects and empty result

Table 4 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 table, the output is empty. 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.

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 to. The relations *covers*, *contains* and *equal* are preferable because they require the least disk accesses. If the sizes of the reference MBRs are considerably different, then the smallest reference MBR must be selected because the cost of retrieval is proportional to the data size (see Figure 11).

	disjoint(p,q_2)	**meet(p,q_2)**	**equal(p,q_2)**	**inside(p,q_2)**	**cvrdby(p,q_2)**	**contain(p,q_2)**	**covers(p,q_2)**	**overlap(p,q_2)**
disjoint(p,q_1)	---	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,q_1)	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,q_1)	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,q_1)	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,q_1)	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,q_1)	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,q_1)	m ∨ e ∨ i ∨ cb ∨ ct ∨ cv ∨ o	e ∨ i ∨ cb ∨ ct ∨ cv ∨ o	d ∨ m ∨ e ∨ iv 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,q_1)	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 4 Conjunctions of relations that yield empty results

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.

6. NON-CRISP MBRS

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 of 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.

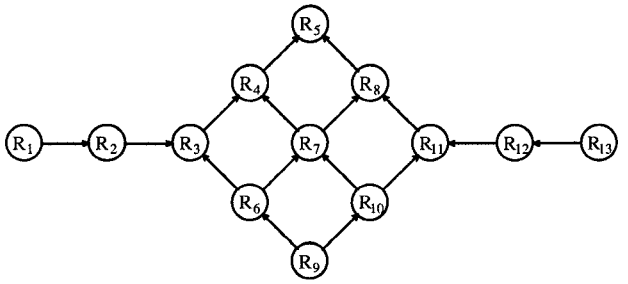
This problem is due to the difficulties of implementing coordinate-based computer algorithms for geometry that preserve topology. One cannot always *reliably* derive from a coordinate representation topological properties such as the coincidence of two points, or whether a point close to a line is actually 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.

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 constraint 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 to represent 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.

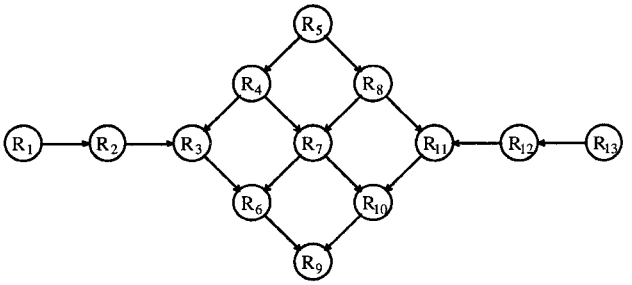
The 4-step algorithm of section 4 can be extended to deal with inaccuracies involving slightly larger than crisp MBRs. Our extension is based on the concept of *conceptual neighbourhood* (Freksa 1992, Egenhofer and Al-Taha 1992). The conceptual neighbourhood of the 13 one-dimensional relations of Figure 2 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 14a. 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 2). Likewise, enlarging the reference object changes the relation between the two objects according to the directions of the edges in Figure 14b.

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 four 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 R_j 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.



(a) Enlargement of primary object



(b) Enlargement of reference object

Fig. 14 Conceptual neighbourhoods for relations in 1d space

We use the notion 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 5 should be used instead of Table 1 in such cases. The dark grey rectangles of Table 5 correspond to the crisp MBR relations displayed in Table 1. The light grey

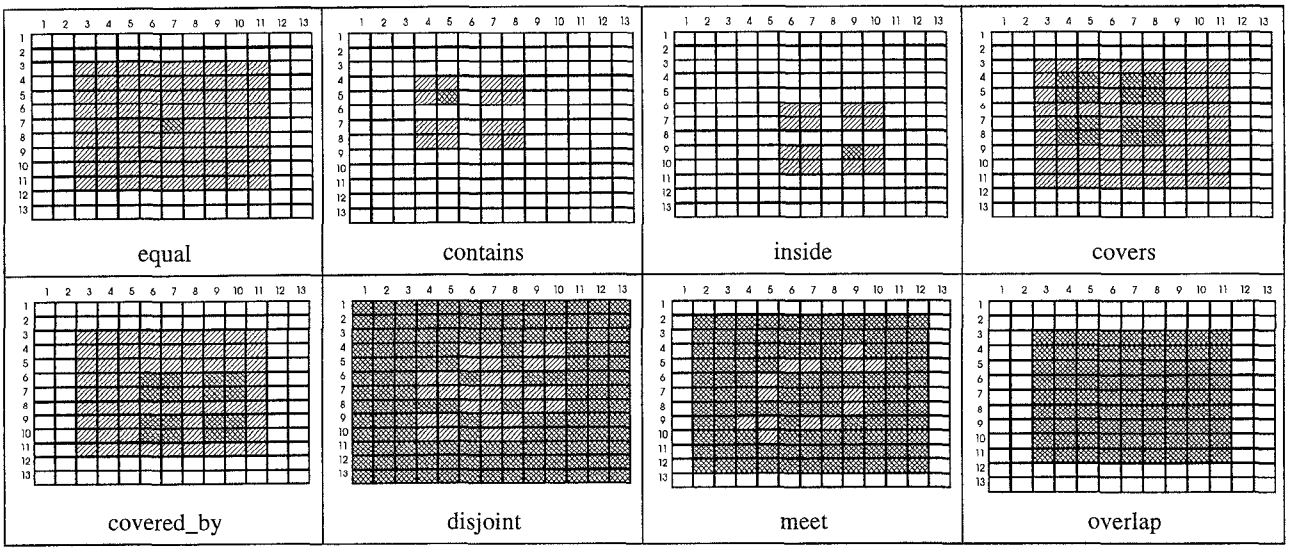


Table 5 Retrieval using 2-neighbours

rectangles are the additional ones that correspond to first and second neighbourhood relations.

The extra MBRs to be retrieved increase the retrieval cost, 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.

7. 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 their importance, topological relations have not been extensively used in spatial data structures. In this paper we have focused 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 to 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 have been 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. An implementation of spatial relations for non-contiguous objects, as well as geographic examples and detailed descriptions can be found (Papadias and Theodoridis, 1994)

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 (1994) have shown that the number of different projections between a region reference object and a line primary object is 221. The ideas in this paper can be extended to include linear and point data, objects with holes etc.

ACKNOWLEDGEMENTS

We would like to thank Cheryl Jacob for proof-reading this paper and providing useful suggestions.

Dimitris Papadias was partially supported by NSF - IRI 9221276. Part of this work was done while he was with the Department of Geoinformation, Technical University of Vienna.

Yannis Theodoridis and Timos Sellis were partially supported by the Department of Research and Technology of Greece (PENED 91).

Max Egenhofer was 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 Organisation, and the Maine Mathematics and Science Alliance.

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. (1993) 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., 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. (1995) 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. 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 Data and Knowledge Engineering, 6(1), 86-95.

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.

Frank, A. U. (1995) 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., 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.

Grigni M., Papadias, D., Papadimitriou, C. (1995) Topological Inference. Submitted.

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.

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

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. Taylor Francis.

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

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

Papadias, D., Sellis, T. (1994) The Qualitative Representation of Spatial Knowledge in two-dimensional Space. VLDB Journal, Special Issue on Spatial Databases, Vol 3, pp. 479-516.

Papadias, D., Theodoridis, Y. (1994) Spatial Relations, Minimum Bounding Rectangles and Spatial Data Structures. Technical Report, KDBSLAB-TR-94-06, National Technical University of Athens, Greece.

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

Papadias, D., Sellis, T. (1995) A Pictorial Query-By-Example Language. To appear in the Journal of Visual Languages and Computing, Special Issue on Visual Query Systems, March 95.

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.

Roussopoulos, N., Kelley, F., Vincent, F. (1995) Nearest Neighbor Queries. In the Proceedings of ACM-SIGMOD Conference.

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.

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