Appears: In the Proceedings of the 2nd *International Conference on Spatial Information Theory* (COSIT'95), Semmering - Vienna, September 1995. Springer-Verlag LNCS.

Range Queries Involving Spatial Relations: A Performance Analysis

Yannis Theodoridis[+]

Dept. of Electrical and Computer Engineering
National Technical University of Athens
GREECE 15773
e-mail: theodor@theseas.ntua.gr

Dimitris Papadias[*]

Dept. of Computer Science and Engineering
University of California, San Diego
CA 92093-0114, USA
e-mail: dimitris@cs.ucsd.edu

Abstract: Spatial relations are becoming an important aspect of spatial access methods because of the increasing interest on qualitative spatial information processing. In this paper we show how queries involving spatial relations can be transformed to range queries and implemented in existing DBMSs. We provide a performance analysis of B- and R- tree indexing methods to support such queries and we evaluate the analytical formulas using experimental results. The proposed analytical models for the expected retrieval cost of spatial relations are proved to be good guidelines for a spatial query optimiser.

Keywords: Spatial relations, Performance Analysis, Query Optimisation.

1. Introduction

Spatial information is often processed qualitatively, using spatial relations, rather than absolute coordinates. [Topa95] describes an example of a computerised system for coordinating first-aid vehicles that uses qualitative spatial constraints to navigate vehicles. Additional cases where spatial relations can be used to solve practical problems involving spatial information can be found in [PS94]. As a result, the formalization, representation and processing of spatial relations has become important for user interfaces and query optimization strategies in Geographic Information Systems [PS95, CSE94]. The significance of spatial relations has also been pointed out by a number of researchers in Spatial and Image Databases, [PFK94, SYH94]. The most common types of spatial relations that have been used in geographic applications include *topological*, *direction* and *distance* relations.

[+] Yannis Theodoridis was partially supported by the Department of Research and Technology of Greece (PENED'91).

[*] Dimitris Papadias was partially supported by NSF - IRI 9221276. He is currently with the National Center for Geographic Information and Analysis, University of Maine, Orono ME 04469-5711.

Topological relations deal with concepts of connectedness and inclusion. According to the 4-intersection model [EF91], the most prevalent model in the GIS literature, eight pairwise disjoint relations can be defined using the four intersections between objects' boundaries and interiors: *disjoint, meet, equal, overlap, contains, inside*, *covers,* and *covered-by*. [Egen91] extended the model by also including intersections among objects' exteriors (9-intersection model). Tests with human subjects have shown that the intersection models have potential for defining cognitively meaningful spatial predicates, a fact that makes the above relations a good candidate for commercial systems [ME94].

Direction relations (*north, northeast*) deal with order in space. Unlike the case of topological relations where the intersection models provide generally accepted definitions, there are no such definitions of direction relations (e.g., is France *north* or *northwest* of Italy?). Although experimental findings from Cognitive and Environmental Psychology can be used as guidelines for the direction relations that people evoke in everyday reasoning, so far the psychological results are too vague to be helpful in defining direction relations in actual systems.

Distance relations (e.g., *near*, *far*) involve distance concepts. For example, two objects are assumed to be *near* if their distance (however distance is defined) is less than a predefined threshold. A form of queries closely related to distance, is the *nearest neighbour queries* (e.g., find the 3 objects closest to a reference object). The previous types of spatial relations have been studied both independently and in conjunction with each other. [Fran92], for instance, proposes a method for qualitative reasoning that combines direction with distance relations.

Recently the interest about spatial relations has shifted towards Spatial Access Methods. The retrieval of spatial relations in existing DBMSs can be accomplished by maintaining traditional indexes (e.g., B^+-trees), or, alternatively, by incorporating Abstract Data Types (ADTs) with specialised indexes defined by external code (e.g., R-trees). Furthermore, when using extended-relational systems, like Postgres [SR86], both indexing methods are available (or easily included) and application developers can decide which is the most appropriate for their application needs. In particular, B^+-trees have been used for the retrieval of direction relations [TPS95], and R-trees and their variations for direction relations [PTS94], topological relations [PTSE95] and nearest neighbour queries [RKV95].

In this paper we treat all queries involving spatial relations between region objects as range queries, and we provide an analysis for their performance. The advantage of treating spatial relations as range queries is that we can use well known results to estimate the expected performance. Our work constitutes the first attempt to model the performance of spatial relations since previous work has focused on *window queries* (retrieval of objects that share common points with a given object or area). The proposed formulas can be used as guidelines by the query optimisers of database systems that support spatial relations, in order to estimate the cost of spatial queries.

The paper is organised as follows: In section 2 we describe a set of "representative" relations and we demonstrate how they can be retrieved using B- and R-trees. In section 3 we provide analytical models that estimate the

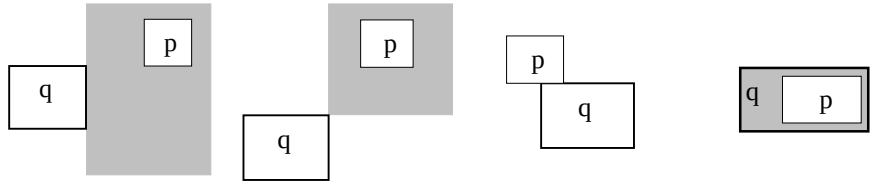
performance of each method. Section 4 evaluates the models of section 3 using experimental tests, and section 5 concludes the paper.

2. Retrieval of Spatial Relations

In this paper we will focus on the direction relations *east* and *northeast,* the topological relations *meet* and *inside*, and some distance relations. For these relations we provide a brief description, we demonstrate their retrieval using MBRs and we outline implementations in spatial data structures. The extension of the results to other relations is straightforward.

2.1 Spatial Relations

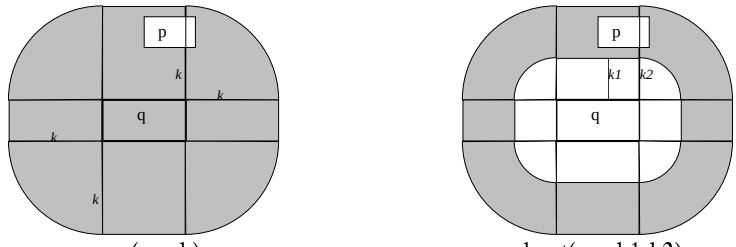
The relation *east*(p,q) means that the x- coordinates of all points of object p (called *primary object*) are larger than or equal to the x- coordinates of all points of object q (called *reference object*). That is, the primary object (p) must be in the grey area of Figure 1a. Similarly, *northeast*(p,q) means that the x- and y- coordinates of all points of object p are larger than or equal to the x- and y- coordinates of all points of object q (Figure 1b). The relations *meet* and *inside* have their usual meaning according to the 4-intersection model (Figure 1c, 1d).



east(p,q) northeast(p,q) meet(p,q) inside(p,q)

Fig. 1. Topological and direction relations

In order to define distance relations between objects we will start from distances between points. The distance between two points (*pp_dist*) p_i ($p_i \in p$) and q_j ($q_j \in q$) is defined according to the Euclidean metric: $pp\_dist(p_i,q_j) = \sqrt{(p_{i\_x} - q_{j\_x})^2 + (p_{i\_y} - q_{j\_y})^2}$ (where $p_{i\text{-}x}$ is the x- coordinate of point p_i, $p_{i\text{-}y}$ is the y- coordinate of point p_i, and so on). Using the distance between points, we define the distance between point p_i and object q (*po_dist*) as the minimum distance of p_i from any point of q: $po\_dist(p_i,q) = \min(pp\_dist(p_i,q_j), \forall q_j \in q)$. Finally we define the distance from object p to object q as the maximum of all po_distances from the points of p to q: $oo\_dist(p,q) = \max(po\_dist(p_i,q), \forall p_j \in p)$. Using the above definitions of distances we define the qualitative relation *near* as: $near(p,q,k) \equiv oo\_dist(p,q) \leq k$, that is, *all* points of p must be within k distance from some point of object q (Figure 2a). Similarly, the relation *about*(p,q,k1,k2), where $0<k1<k2$, can be defined as: $about(p,q,k1,k2) \equiv near(p,q,k2) \wedge \neg\exists\, p_i$ ($po\_dist(p_i,q) \leq k1$). That is, according to Figure 2b, all points of p are within distance k1 and k2.



near(p,q,k) about(p,q,k1,k2)

Fig. 2. Distance relations

We will also consider conjunctions of spatial relations. Figure 3a illustrates a configuration that corresponds to the relation *northeast*(p,q) ∧ *near*(p,q,k), while Figure 3b illustrates a configuration for *east*(p,q) ∧ *meet*(p,q).

northeast(p,q) ∧ near(p,q,k) east(p,q) ∧ meet(p,q)

Fig. 3. Conjunctions of spatial relations

Spatial access methods usually store approximations of objects that need only a few points for their representation instead of the objects themselves. Such approximations are used to efficiently retrieve candidates that could satisfy a query. In the next subsection we show how Minimum Bounding Rectangle (MBR) approximations can be used in the retrieval of spatial relations.

2.2 Minimum Bounding Rectangles

MBRs have been used extensively to approximate objects in Spatial Data Structures and Spatial Reasoning because they need only four numbers for their representation; in particular, each object *p* is represented by the four numbers: p'_{l-x}, p'_{l-y}, p'_{u-x}, p'_{u-y}, where p'_{l-x} stands for the x- coordinate of the lower/left point of MBR p', p'_{l-y} for the y- coordinate of the lower/left point, p'_{u-x} for the x-coordinate of the upper/right point and p'_{u-y} stands for the y- coordinate of the upper/right point.

Since the MBRs are only approximations of the actual objects, the spatial relation between MBRs does not necessarily coincide with the spatial 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 example, the MBRs of objects that *meet* a reference object can be related by any topological relation but *disjoint* [PTSE95]. Figure 4 illustrates three configurations where the MBRs satisfy the relations *overlap*, *meet* and *contain* and the actual objects *meet*.

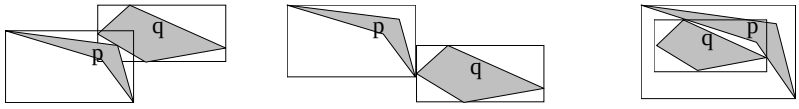


Fig. 4. Some configurations of MBRs for which the actual objects *meet*

Furthermore, it can be concluded from Figures 1, 2 and 3 that all the relations constrain some or all points of the primary MBR to a *range* (subdivision) of space. In case of *east*, for example, all points of p' must be in the right semi-plane as defined by the vertical line that passes from q's eastern boundary. In some cases all the points of the MBRs to be retrieved must be within the range (e.g., *east*) and this is a sufficient condition for retrieval. In some other cases the range is not a sufficient condition and more specific constraints hold (e.g., *east* $\wedge$ *meet* has the same range as *east* but in this case the MBRs should also *meet*). In general, in order to answer the query "find all objects p that satisfy the relation R with respect to an object q" we have to retrieve all MBRs p' that satisfy certain range constraints with respect to the MBR q' of object q. Table 1 illustrates the mapping from spatial relations R between actual objects to constraints between MBR coordinates. Although the constraints for *east* and *northeast* relations are trivial, the rest require some careful study in order to be clearly understood. For example, the constraints for *meet* state that the two MBRs may not be *disjoint* but permits any other topological relation between them, because any not *disjoint* MBRs may contain objects that *meet* (e.g., Figure 4).

Relation	Constraints on the $p'_{l\text{-}x}$, $p'_{l\text{-}y}$, $p'_{u\text{-}x}$, $p'_{u\text{-}y}$ parameters with respect to the reference MBR q'
east(p,q)	$p'_{l\text{-}x} \geq q'_{u\text{-}x}$
northeast(p,q)	$(p'_{l\text{-}y} \geq q'_{u\text{-}y}) \wedge (p'_{l\text{-}x} \geq q'_{u\text{-}x})$
meet(p,q)	$(p'_{l\text{-}y} \leq q'_{u\text{-}y}) \wedge (p'_{u\text{-}y} \geq q'_{l\text{-}y}) \wedge (p'_{l\text{-}x} \leq q'_{u\text{-}x}) \wedge (p'_{u\text{-}x} \geq q'_{l\text{-}x})$
inside(p,q)	$(q'_{l\text{-}y} < p'_{l\text{-}y} < q'_{u\text{-}y}) \wedge (q'_{l\text{-}y} < p'_{u\text{-}y} < q'_{u\text{-}y}) \wedge$ $(q'_{l\text{-}x} < p'_{l\text{-}x} < q'_{u\text{-}x}) \wedge (q'_{l\text{-}x} < p'_{u\text{-}x} < q'_{u\text{-}x})$
near(p,q,k)	$(q'_{l\text{-}y}\text{-}k \leq p'_{l\text{-}y} \leq q'_{u\text{-}y}+k) \wedge (q'_{l\text{-}y}\text{-}k \leq p'_{u\text{-}y} \leq q'_{u\text{-}y}+k) \wedge$ $(q'_{l\text{-}x}\text{-}k \leq p'_{l\text{-}x} \leq q'_{u\text{-}x}+k) \wedge (q'_{l\text{-}x}\text{-}k \leq p'_{u\text{-}x} \leq q'_{u\text{-}x}+k)$
about(p,q,k1,k2)	$(q'_{l\text{-}y}\text{-}k2 \leq p'_{l\text{-}y} \leq q'_{u\text{-}y}+k2) \wedge (q'_{l\text{-}y}\text{-}k2 \leq p'_{u\text{-}y} \leq q'_{u\text{-}y}+k2) \wedge$ $(q'_{l\text{-}x}\text{-}k2 \leq p'_{l\text{-}x} \leq q'_{u\text{-}x}+k2) \wedge (q'_{l\text{-}x}\text{-}k2 \leq p'_{u\text{-}x} \leq q'_{u\text{-}x}+k2) \wedge$ $\neg ((p'_{l\text{-}y} \leq q'_{u\text{-}y}+k1) \wedge (p'_{u\text{-}y} \geq q'_{l\text{-}y}\text{-}k1) \wedge$ $(p'_{l\text{-}x} \leq q'_{u\text{-}x}+k1) \wedge (p'_{u\text{-}x} \geq q'_{l\text{-}x}\text{-}k1))$
near(p,q,k)$\wedge$ northeast(p,q)	$(q'_{u\text{-}y} \leq p'_{l\text{-}y} \leq q'_{u\text{-}y}+k) \wedge (q'_{u\text{-}y} \leq p'_{u\text{-}y} \leq q'_{u\text{-}y}+k) \wedge$ $(q'_{u\text{-}x} \leq p'_{l\text{-}x} \leq q'_{u\text{-}x}+k) \wedge (q'_{u\text{-}x} \leq p'_{u\text{-}x} \leq q'_{u\text{-}x}+k)$
meet(p,q) $\wedge$ east(p,q)	$(p'_{l\text{-}x} = q'_{u\text{-}x}) \wedge (p'_{l\text{-}y} < q'_{u\text{-}y}) \wedge (p'_{u\text{-}y} > q'_{l\text{-}y})$

Table 1. Constraints for the retrieval of spatial relations using MBRs

Because MBRs differ from the actual objects they enclose, they are not always adequate to express the relation between the actual 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. The relations *meet* and *inside* require a refinement step [PTSE95], while *east* and *northeast* do not [PTS94], that is, all MBRs enclose objects that satisfy the query.

The distance relations may also need a refinement because near(p',q',k) does not necessarily imply near(p,q,k) and vice versa (the same is true for *about*). If it is not the case that near(p',q',k), but three out of four vertices of the primary MBR p' are within k distance (*po_dist*) from q' (and one vertex in po_dist > k), then p' may enclose an object that satisfies the relation near(p,q,k) (see Figure 5a). Obviously, the same MBR may enclose an object that is not *near* and a refinement step is needed to make the distinction. If however, two vertices of p' are further than k distance, then p cannot be *near* (see Figure 5b). This conclusion is based on the observation that each edge of the MBR coincides with at least one point of the enclosed object. If two vertices are further than distance k, it means that at least a whole edge (and therefore some point of the enclosed object) is further than k. In some cases it is also possible that the entire MBR is within distance k, but the objects are not *near* because some (or all) points of p are further than k from any point of q (Figure 5c). Thus, the refinement step is needed for all MBRs p' for which there exist points that are further than k from any point of q'. Similarly, conjunctions of relations in which one relation needs a refinement step (*northeast* $\wedge$ *near*, *meet* $\wedge$ east), also require refinement.

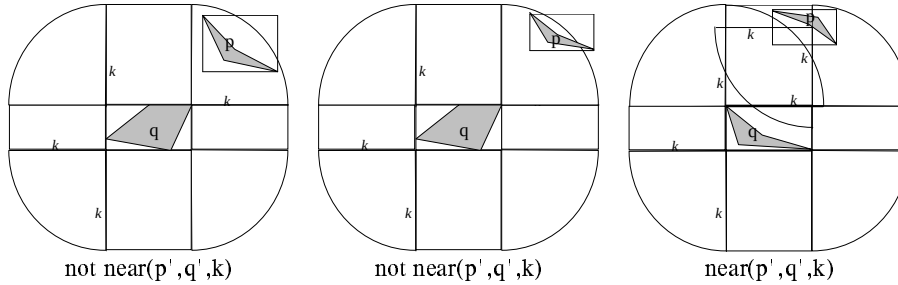


Fig. 5. Configurations for which a refinement step is needed in the case of *near*

In the rest of the section we will demonstrate how we can use existing data structures to retrieve spatial relations.

2.3 Implementation of Spatial Relations

The first solution for the retrieval of direction relations includes the maintenance of four B-tree indexes (in the rest of the paper we will refer to B^+-trees using the

general term "B-trees"). Each index corresponds to one of the four numbers: $p'_{l\text{-}x}$, $p'_{l\text{-}y}$, $p'_{u\text{-}x}$, $p'_{u\text{-}y}$. Obviously, some relations imply search on one B-tree while others imply search on more B-trees. For instance, the query "find all objects p that are *east* of object q" is transformed to the constraint $p'_{l\text{-}x} \geq q'_{u\text{-}x}$ (see Table 1) which is a simple range query in the corresponding B-tree. On the other hand, most queries need to search two or more B-trees and, in a second phase, to compute the intersection of the intermediate answer sets (for details see [TPS95]).

In general, the processing of a query of the form "find all objects p that satisfy a given spatial relation with respect to object q" using B-trees involves the following steps:

Step 1. Depending on the relation to be retrieved, select the B-trees to be searched from the set of four indexes. This procedure involves Table 1.

Step 2. Search each index involved to find the corresponding answer sets.

Step 3. If multiple indexes are involved, find the intersection set. A "realistic" assumption is that this procedure is executed in main memory.

Step 4. If necessary, follow a refinement step for the selected object IDs.

The performance of the retrieval mechanism using B-trees depends significantly on the particular relation because the number of B-trees to be searched relies on the number of constraints that are involved in the definition of the relation. *East*, for example, involves only one constraint ($p'_{l\text{-}y} \geq q'_{u\text{-}y}$,), while *near* contains four constraints. As it will be shown later this fact significantly affects the efficiency of retrieval.

Another data structure that is efficient for spatial relations is the R-tree. The R-tree [Gutt84] is a height-balanced tree, which consists of intermediate and leaf nodes. 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. Improved variations of R-trees include the R^+-trees [SRF87] and the R^*-trees [BKSS90]. In this paper we use R*-trees because we found them to have consistently better performance in the retrieval of spatial relations than both R- and R^+- trees.

In order to retrieve objects that satisfy a spatial relation with respect to a reference object we have to specify the MBRs that could enclose such objects using Table 1 and then to search the intermediate nodes that contain these MBRs. For instance, the intermediate nodes P that could contain MBRs p' that satisfy the relation *east* with respect to q' ($p'_{l\text{-}x} \geq q'_{u\text{-}x}$) should satisfy the constraint $P_{u\text{-}x} \geq q'_{u\text{-}x}$. Table 2 presents the constraints for the intermediate nodes for each direction relation of Table 1. Notice that the same relation between intermediate nodes and the reference MBR holds for all the levels of the tree structure.

Relation	Constraints for the intermediate Nodes P to be Searched with respect to the reference MBR q'
east(p,q)	$P_{u\text{-}x} \geq q'_{u\text{-}x}$
northeast(p,q)	$(P_{u\text{-}x} \geq q'_{u\text{-}x}) \wedge (P_{u\text{-}y} \geq q'_{u\text{-}y})$
meet(p,q)	$(P_{l\text{-}y} \leq q'_{u\text{-}y}) \wedge (P_{u\text{-}y} \geq q'_{l\text{-}y}) \wedge (P_{l\text{-}x} \leq q'_{u\text{-}x}) \wedge (P_{u\text{-}x} \geq q'_{l\text{-}x})$
inside(p,q)	$(P_{l\text{-}y} < q'_{u\text{-}y}) \wedge (P_{u\text{-}y} > q'_{l\text{-}y}) \wedge (P_{l\text{-}x} < q'_{u\text{-}x}) \wedge (P_{u\text{-}x} > q'_{l\text{-}x})$
near(p,q,k)	$(P_{l\text{-}y} \leq q'_{u\text{-}y}+k) \wedge (P_{u\text{-}y} \geq q'_{l\text{-}y}-k) \wedge$ $(P_{l\text{-}x} \leq q'_{u\text{-}x}+k) \wedge (P_{u\text{-}x} \geq q'_{l\text{-}x}-k)$
about(p,q,k1,k2)	$(P_{l\text{-}y} \leq q'_{u\text{-}y}+k2) \wedge (P_{u\text{-}y} \geq q'_{l\text{-}y}-k2) \wedge (P_{l\text{-}x} \leq q'_{u\text{-}x}+k2) \wedge$ $(P_{u\text{-}x} \geq q'_{l\text{-}x}-k2) \wedge \neg ((q'_{l\text{-}y}-k1 \leq P_{l\text{-}y} \leq q'_{u\text{-}y}+k1) \wedge$ $(q'_{l\text{-}y}-k1 \leq P_{u\text{-}y} \leq q'_{u\text{-}y}+k1) \wedge (q'_{l\text{-}x}-k1 \leq P_{l\text{-}x} \leq q'_{u\text{-}x}+k1) \wedge$ $(q'_{l\text{-}x}-k1 \leq P_{u\text{-}x} \leq q'_{u\text{-}x}+k1))$
near(p,q,k)∧ northeast(p,q)	$(q'_{l\text{-}y}-k \leq P_{l\text{-}y} \leq q'_{u\text{-}y}+k) \wedge (P_{u\text{-}y} \geq q'_{u\text{-}y}) \wedge$ $(q'_{l\text{-}x}-k \leq P_{l\text{-}x} \leq q'_{u\text{-}x}+k) \wedge (P_{u\text{-}x} \geq q'_{u\text{-}x})$
meet(p,q)∧east(p,q)	$(P_{l\text{-}y} < q'_{u\text{-}y}) \wedge (P_{u\text{-}y} > q'_{l\text{-}y}) \wedge (P_{l\text{-}x} \leq q'_{u\text{-}x}) \wedge (P_{u\text{-}x} > q'_{u\text{-}x})$

Table 2 Constraints for intermediate nodes of R-trees

In general, the processing of a query of the form "find all objects p that satisfy a given spatial relation with respect to object q" using R-trees involves the following steps:

Step 1. Starting from the top node, exclude the intermediate nodes P which could not enclose MBRs that satisfy the spatial relation and recursively search the remaining nodes. This procedure involves Table 2.

Step 2. Among the leaf nodes retrieved, select the ones that satisfy the spatial relation. This procedure involves Table 1.

Step 3. If necessary, follow a refinement step for the selected MBRs.

Intuitively, R-trees perform better than B-trees in cases where many constraints are involved in the definition of the direction relation of interest. The next section provides a mathematical analysis that supports this argument.

3. Cost Analysis

In this section we provide analytical formulas that estimate the performance of B-trees and R-trees on the retrieval of spatial relations. Existing formulas for the expected performance of the above structures focus on traditional retrieval (matching queries on B-trees [Yao78, Come79, Bato81] and overlap queries on R-trees [FSR87, PSTW93, FK94]). Our work extends previous work and estimates the expected cost (i.e., number of disk accesses) for the retrieval of several types of spatial relations and combinations. In this discussion we assume that both data and query rectangles are uniformly distributed over the unit square address space.

3.1. Analysis of B-trees

As explained in section 2, searching between one and four B-trees is necessary depending on the relation we want to retrieve. Constraints can be grouped in two categories:

(a) *exact matching* constraints (e.g., $p'_{l\text{-}y} = q'_{u\text{-}y}$) and

(b) *partial matching* constraints (e.g., $p'_{l\text{-}y} > q'_{u\text{-}y}$, $q'_{l\text{-}y} < p'_{l\text{-}y} < q'_{u\text{-}y}$) which are characterised by a range r ($0 \le r \le 1$).

If we suppose that the data keys are stored in a B-tree index of height h with L leaf nodes then the average cost C(r) for the retrieval of a constraint with range r is [Come79]

$$C(r) = h + r \cdot L \tag{1}$$

It is obvious that the exact matching constraint is a special case of partial matching constraint ($r = 0$). It is also clear that the cost for the retrieval of a relation is the sum of the costs for each constraint involved.

In order to compute the expected cost C(r) for the retrieval of a constraint characterised by a range r we need to provide equations for the parameters of Eq. 1, namely h, L, r. Suppose now that m is the maximum number of entries in a B-tree node, c is the average capacity of a node, and N is the total number of keys stored in the leaf nodes. We have the following equations [Bato81] for the average h and L values in order to use them in Eq. 1:

$$h = 1 + \left\lceil \log_{c \cdot m} \frac{N}{m} \right\rceil \tag{2}$$

$$L = N / c \cdot m \tag{3}$$

What remains in order to have a complete expression for Eq. 1 is the value of parameter r which depends on the particular constraint. If we assume that the size of a data object MBR is $p_x \bullet p_y$ and the size of a query object MBR is $q_x \bullet q_y$ then we can provide in Table 3 the values of parameters r according to possible constraints (the constraints refer only to x- coordinate since constraints for y- coordinate can be expressed in a similar way).

Constraint	Average range r
$p'_{l\text{-}x} < q'_{l\text{-}x}$ or $p'_{u\text{-}x} > q'_{u\text{-}x}$	$r = (1 - q_x) / 2$
$q'_{l\text{-}x} < p'_{l\text{-}x} < q'_{u\text{-}x}$ or $q'_{l\text{-}x} < p'_{u\text{-}x} < q'_{u\text{-}x}$	$r = q_x$
$p'_{l\text{-}x} > q'_{u\text{-}x}$ or $p'_{u\text{-}x} < q'_{l\text{-}x}$	$r = (1 - (2 \bullet p_x + q_x)) / 2$
$p'_{l\text{-}x} < q'_{u\text{-}x}$ or $p'_{u\text{-}x} > q'_{l\text{-}x}$	$r = (1 + q_x) / 2$

Table 3 Average values for range r of a constraint

Using information from Table 3 and Eq. 2 and 3 we can estimate the expected cost for each constraint (see Eq. 1). Summing up, the expected cost C(R,k) for the retrieval of a spatial relation R with k constraints is:

$$C(R,k) = \sum_{i=1}^{k} C(r_i) = \sum_{i=1}^{k} (h + r_i \cdot L) \tag{4}$$

3.2. Analysis of R-trees

Most of the work in the literature has dealt with the expected performance of R-trees for processing overlap queries i.e., the retrieval of data objects p that share common area with a query window q. More particularly, let N be the total number of data objects indexed in a R-tree, h be the height of the tree, c the average node capacity at every level of the tree and m the maximum number of entries in a node. If we assume that the average node size at level j is $n_{j,x} \cdot n_{j,y}$ (the root is assumed at level $j=h$ and the leaf-nodes at level $j=1$) and the average size of a query object MBR is $q_x \cdot q_y$ then the expected retrieval cost (number of disk accesses) of an overlap query using R-trees is [PSTW93, FK94]

$$C(q_x,q_y)=\sum_{j=1}^{h}\left\{\frac{N}{(c\cdot m)^j}\cdot\left(n_{j,x}+q_x\right)\cdot\left(n_{j,y}+q_y\right)\right\} \tag{5}$$

The expression for computing the height h of the R-tree is similar to that of B-trees (Eq. 2). If we name N_j the number of nodes at level j, and d_j the density (i.e., the sum of the nodes' areas divided by the global area) of these nodes then the average node sizes $n_{j,x}$ and $n_{j,y}$ are given by the following equations [TS95]:

$$n_{j,x} = n_{j,y} = \left(d_j / N_j\right)^{1/2} \tag{6}$$

where

$$d_j = \left\{1+\frac{(d_{j-1})^{1/2}-1}{(c\cdot m)^{1/2}}\right\}^2 \tag{7}$$

and

$$N_j = \frac{N_{j-1}}{c\cdot m} \tag{8}$$

Therefore, d_j and N_j can be computed recursively using d_0 and N_0 which denote the density d and the amount N respectively of the data object MBRs. Qualitatively, this means that we can estimate the retrieval cost of a window query just with the knowledge of the data set and the query window.

Since Eq. 5 expresses the expected performance of R-trees on overlap queries using a query window q, in order to estimate the retrieval cost of a spatial relation R(p,q) we need the following transformation: R(p,q) $\Rightarrow$ overlap(p',Q). In other words, the retrieval of a spatial relation using R-trees is equivalent (in terms of cost) to the retrieval of an overlap query using an appropriate query window Q. The necessary transformation Q for each spatial relation R should take into consideration the corresponding constraint of the intermediate nodes because only these nodes are important when estimating the retrieval cost [PTSE95].

For the spatial relations that we consider in this paper, the appropriate query windows Q are illustrated in Figure 6. Each query window is an appropriate transformation of the corresponding constraints presented in Table 2. Notice that *meet* and *inside* correspond to the same query window Q. This is a property that can be extracted by examining the constraints of Table 2. The same property holds

in our tests for *near* and *about* (if k=k2) because, according to Table 2, the constraints for the intermediate nodes to be searched for about(p,q,k1,k2) are identical to the ones for near(p,q,k2) plus some extra constraints which can be evaluated during the refinement step.

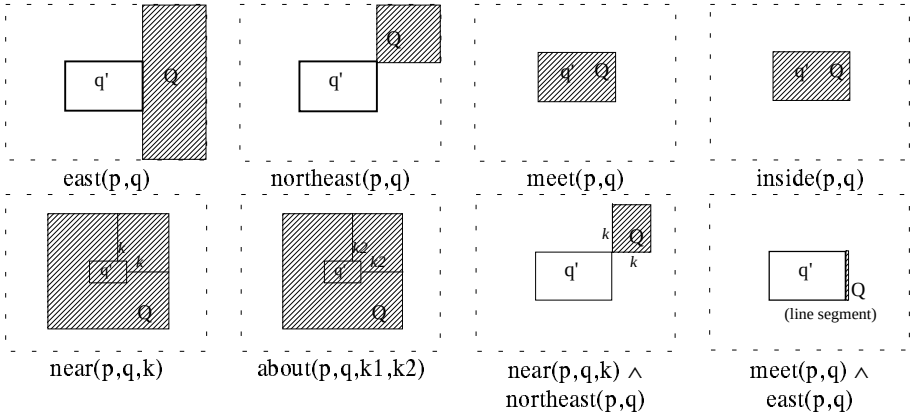


Fig. 6. Query windows for the estimation of the retrieval cost

4. Experimental Results

In order to evaluate the quality of the proposed analytical approximations, we compare the expected and the experimental cost for the retrieval of several representative spatial relations using both the B-tree and the R-tree indexing mechanisms. For the experimental tests we used several *data files* that contained 10,000 data object MBRs with small up to large data sizes (p_x and p_y ranged from 0.5% up to 5% of global side size respectively). The sizes of the reference object MBRs used for the retrieval of spatial relations were equal to the corresponding data object MBRs (i.e., $p_x=p_y=q_x=q_y$). For the distance relations we set $k=3 \cdot q_x$ (*near*), $k1=q_x$ and $k2=3 \cdot q_x$ (*about*).

The expected cost using B-trees was computed by using Eq. 4, information from Tables 1 and 3, and the following typical values:

- average capacity at leaf nodes $c = 0.67$,
- maximum number of entries in a node $m = 126$ (1 page of 1024 bytes includes 126 keys • 4 bytes + 127 pointers • 4 bytes + 12 bytes node-overhead),
- total number of keys $N = 10,000$.

On the other hand, the expected cost using R-trees was computed by using Eq. 5, information from Figure 6 and the following typical values:

- average node capacity $c = 0.67$,
- maximum number of entries in a node $m = 50$ (1 page of 1024 bytes includes 50 entries • 4 values per entry • 4 bytes + 50 pointers • 4 bytes + 24 bytes node-overhead),
- total number of data $N = 10,000$.

The results for each spatial relation are illustrated in Figure 7. The experimental results are illustrated with columns and the analytical results with lines.

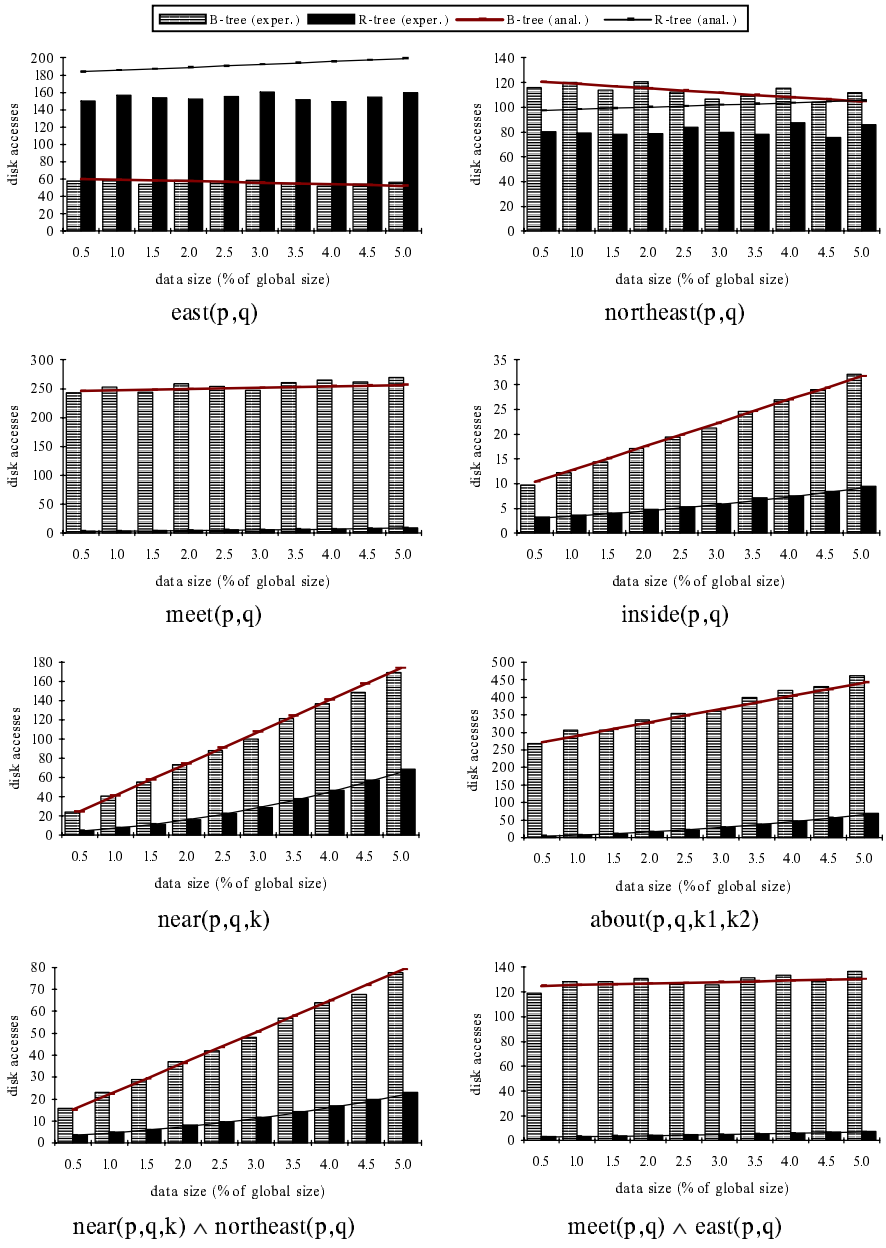


Fig. 7. Experimental vs. analytical results

The common observation in all the graphs of Figure 7 is that the analytical estimate is very close to the experimental results. With the exception of the R-tree estimation for *east* and *northeast*, the relative error is usually below 5%. The weakness of the R-tree model for *east* and *northeast* is, more or less, expected because of the very large window queries Q (see Figure 6) that make R-trees an unstable index for these relations. On the other hand, the range of a constraint in a B-tree query does not affect the B-tree retrieval mechanism and, therefore, the B-tree estimate is always very close to the experimental results.

The comparison of the B- and R- tree indexing mechanism depends on two factors: the number of constraints involved (for B-trees) and the size of the query window Q (for R-trees). For example, B-trees perform better than R-trees when one constraint and large Q are involved (*east*). When the opposite happens, R-trees perform better than B-trees. Following these guidelines, a spatial query optimiser can predict the efficiency of the one or the other indexing mechanism on the support of several spatial queries.

According to our experimental tests, the analytical models of section 3 are proved to be efficient for the estimation of the particular spatial queries that we have implemented using transformation to range queries. Following the same procedure, the performance of other spatial queries can also be estimated with similar accuracy.

5. Conclusion

Relations in space are becoming an important aspect of access methods as a result of the increasing interest on qualitative spatial information processing. In this paper we focus on the retrieval of spatial relations using classic alphanumeric (B-trees) and spatial (R-trees) indexing methods. First we transform queries involving spatial relations into range queries, then we provide analytical formulas for their expected performance (extending previous work on analysis of indexing methods), and finally we evaluate the analytical model. In most cases we found the analytical estimate almost identical to the actual results, a fact that leads to the conclusion that the derived formulas can be used successfully in query optimisers of Geographic Databases in order to estimate the cost of spatial queries.

Although we have worked with a small set of representative topological, direction and distance relations, the results are directly applicable to other spatial relations and combinations. Future work can be done:

(a) to apply the analytical models on specialised spatial relations such as nearest-neighbour or furthest-neighbour and combinations,

(b) to provide analytical models for other spatial indexing methods, such as Grid files [NHS84] or K-D-B-trees [Robi81], in order to evaluate their efficiency with respect to the particular spatial relations and

(c) to derive analytical formulas assuming a general (non-uniform) distribution of objects over the work space in order to efficiently support any kind of spatial information.

Acknowledgements

We thank Emmanuel Stefanakis for implementing and testing B-trees. We also thank Timos Sellis for providing useful comments.

References

[Bato81] Batory, D.S., "B+ Trees and Indexed Sequential Files: A Performance Comparison", In the Proceedings of ACM SIGMOD Conference, 1981.

[BKSS90] Beckmann, N., Kriegel, H.P. Schneider, R., Seeger, B., "The R*-tree: an Efficient and Robust Access Method for Points and Rectangles", In the Proceedings of ACM SIGMOD Conference, 1990.

[CSE94] Clementini, E., Sharma, J., Egenhofer, M., "Modeling Topological Spatial Relations: Strategies for Query Processing", International Journal of Computer and Graphics, 18(6), 815-822.

[Come79] Comer, D., "The Ubiquitous B-Tree", ACM Computing Surveys, Vol. 11(2), pp. 121-137, 1979.

[Egen91] Egenhofer, M., "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, 1991.

[EF91] Egenhofer, M., Franzosa R., "Point-Set Topological Spatial Relations", International Journal of Geographic Information Systems, Vol 5(2), pp. 160-174, 1991.

[FSR87] Faloutsos, C., Sellis, T., Roussopoulos, N., "Analysis of Object Oriented Spatial Access Methods", In the Proceedings of ACM SIGMOD Conference, 1987.

[FK94] Faloutsos, C., Kamel, I., "Beyond Uniformity and Independence: Analysis of R-trees Using the Concept of Fractal Dimension", In the Proceedings of the 13th ACM PODS Symposium, 1994.

[Fran92] Frank, A.U., "Qualitative Spatial Reasoning about Distances and Directions in Geographic Space", Journal of Visual Languages and Computing, Vol. 3, pp. 343-371, 1992.

[Gutt84] Guttman, A., "R-trees: A Dynamic Index Structure for Spatial Searching", In the Proceedings of ACM SIGMOD Conference, 1984.

[Knut73] Knuth, D., "The Art of Computer Programming, vol.3: Sorting and Searching", Addison-Wesley, 1973.

[ME94] Mark, D., Egenhofer, M., "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, 1994.

[NHS84] Nievergelt, J., Hinterberger, H., Sevcik, K.C., "The Grid File: An Adaptable, Symmetric Multikey file Structure", ACM Transactions on Database Systems, Vol 9(1), pp. 38-71, 1984.

[PSTW93] Pagel, B., Six, H., Toben, H., Widmayer, P., "Towards an Analysis of Range Query Performance", In the Proceedings of the 12th ACM PODS Symposium, 1993.

[PFK94] Papadias, D., Frank, A.U., Koubarakis, M., "Constraint-Based Reasoning in Geographic Databases: The Case of Symbolic Arrays", In the Proceedings of the 2nd ICLP Workshop on Deductive Databases, 1994.

[PS94] Papadias, D., Sellis, T., " Qualitative Representation of Spatial Knowledge in two-dimensional Space", Very Large Data Bases Journal, Special Issue on Spatial Databases, Vol 3(4), pp. 479-516, 1994.

[PTS94] Papadias, D., Theodoridis, Y., Sellis, T., "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, 1994.

[PS95] Papadias, D., Sellis, T., "A Pictorial Query-by-Example Language", Journal of Visual Languages and Computing, Special Issue on Visual Query Systems, 6(1), pp 53-72, 1995.

[PTSE95] Papadias, D., Theodoridis, Y., Sellis, T., Egenhofer, M., "Topological Relations in the World of Minimum Bounding Rectangles: a Study with R-trees", In the Proceedings of ACM SIGMOD Conference, 1995.

[Robi81] Robinson, J.T., "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes", In the Proceedings of ACM SIGMOD Conference, 1981.

[RKV95] Roussopoulos, N., Kelley, F., Vincent, F., "Nearest Neighbor Queries", In the Proceedings of ACM SIGMOD Conference, 1995.

[SRF87] Sellis, T., Roussopoulos, N., Faloutsos, C., "The R^+-tree: A Dynamic Index for Multi-Dimensional Objects", In the Proceedings of the 13th Very Large Data Bases Conference, 1987.

[SR86] Stonebraker, M., Rowe, L., "The Design of Postgres", In the Proceedings of ACM SIGMOD Conference, 1986.

[SYH94] Sistla, P., Yu, C., Haddad, R., "Reasoning about Spatial Relationships in Picture Retrieval Systems", In the Proceedings of the 20th Very Large Data Bases Conference, 1994.

[Topa95] Topaloglou, T., "Spatial Databases with Partial Information: Representation and Reasoning", Forthcoming Ph.D Thesis, University of Toronto, Canada, 1995.

[TPS95] Theodoridis, Y., Papadias, D., Stefanakis, E., "Supporting Direction Relations in Spatial Database Systems", Technical Report, KDBSLAB-TR-95-02, National Technical University of Athens, Athens, Greece, 1995.

[TS95] Theodoridis, Y., Sellis, T., "Indexing Point and Non-point Spatial Data: A Performance Analysis", Technical Report, KDBSLAB-TR-95-03, National Technical University of Athens, Athens, Greece, 1995.

[Yao78] Yao, A.C., "On Random 2-3 Trees", Acta Informatica, Vol. 9(2), pp. 159-168, 1978.