Proc. 24th Int. Conf. on Very Large Databases (VLDB), New York City, NY, 1998

Improving Adaptable Similarity Query Processing by Using Approximations

Mihael Ankerst, Bernhard Braunmüller, Hans-Peter Kriegel, Thomas Seidl
Institute for Computer Science, University of Munich, Germany
http://www.dbs.informatik.uni-muenchen.de
{ankerst, braunmue, kriegel, seidl}@dbs.informatik.uni-muenchen.de

Abstract

Similarity search and content-based retrieval are becoming more and more important for an increasing number of applications including multimedia, medical imaging, 3D molecular and CAD database systems. As a general similarity model that is particularly adaptable to user preferences and, therefore, fits the subjective character of similarity, quadratic form distance functions have been successfully employed, e.g. for color histograms as well as for 2D and 3D shape histograms. Although efficient algorithms for processing adaptable similarity queries using multidimensional index structures are available, the quadratic nature of the distance function strongly affects the CPU time which in turn represents a high percentage of the overall runtime. The basic idea of our approach is to reduce the number of exact distance computations by adapting conservative approximation techniques to similarity range query processing and, in addition, to extend the concepts to *k*-nearest neighbor search. As part of a detailed analysis, we show that our methods guarantee no false drops. Experiments on synthetic data as well as on a large image database containing 112,000 color images demonstrate a significant performance gain, and the CPU time is improved by a factor of up to 6.

1 Introduction

In recent years, a wide range of database applications has appeared for which new query types turn out to be useful. In particular, similarity search is an essential query type for spatial and multimedia databases containing images, video audio or 3D-objects [Jag 91] [AFS 93] [GM 93] [FRM 94] [ALSS 95] [Kor+ 96] [BK 97]. The last few years of research have produced several results for efficiently supporting similarity search, and among them, quadratic form distance functions have shown their high usefulness. They were successfully used for color histogram similarity [Fal+ 94] [Haf+ 95] [SK 97], 3-D shape similarity [KSS 97] [KS 98], pixel-based similarity [AKS 98], and several other similarity models [Sei 97]. The reason for using quadratic forms as distance functions is the observation that for many applications, the Euclidean distance is not adequate due to its fundamental assumption that all dimensions are independent of each other. Any quadratic form distance function $d_A^2(x, y) = (x - y) \cdot A \cdot (x - y)^T$ is determined by a similarity matrix A whose components represent the mutual similarities, or correlations, of the dimensions. If the matrix A is positive definite, i.e. $d_A^2(x, y) > 0$ for $x \neq y$, meaningless negative distance values are avoided. Whereas the Euclidean distance produces spherical query regions, general quadratic form distance functions represent ellipsoids as query regions which give the new query type its name, *ellipsoid query.* In [SK 97], a novel algorithm for efficient ellipsoid query processing on multidimensional index structures was presented which directly uses the exact representation of an ellipsoid as the query region. However, we may not rely on the applicability of the exact method for the following reasons:

Legacy Systems. Imagine that you are bound to a legacy system that only supports multidimensional window queries or sphere queries, and which resists an extension, for example, one which is necessary for the algorithm for exact ellipsoid query processing as proposed in [SK 97]. In order to provide efficient support for ellipsoid queries in spite of this restriction, we investigate the adaptability of standard approximation techniques to ellipsoid queries.

Performance Aspects. Since the evaluation time for an ellipsoid function is quadratic in the general case, it may bring benefits to use approximations for query processing. Thus we can achieve a reduction of the time complexity for calculations on data pages as well as on directory pages. In a *d*-dimensional space, testing whether a database object is contained in the query ellipsoid requires $O(d^2)$ time, and testing the intersection of a rectilinear box from the index and the query ellipsoid takes $O(d^2 \cdot i)$ time for a small iteration factor i [SK 97]. By

Proceedings of the 24th VLDB Conference
New York, USA, 1998

using appropriate approximations, these complexities may be reduced to linear time. In order to meet this expectation, the selected approximations should be simple for storage and computational reasons. On the other hand, the approximation should have a high quality of approximation to the ellipsoid so as to save as many exact evaluations as possible. The impact of such methods on the performance has to be evaluated by experiments which require the availability of approximation techniques that are competitive with the exact ellipsoid query processing algorithm. The relevance of reducing the number of distance evaluations in comparison with the overall runtime will be demonstrated later in this paper.

In the past, the use of approximation techniques for efficient query processing was extensively investigated in the context of 2-D spatial database systems, in particular to efficiently support point and range queries [BKS 93] [KSB 93] as well as spatial joins [BK 94] [Bri 94]. From the investigated types, *conservative approximations* in particular meet the requirements of range query processing. Since they totally enclose the exact objects, conservative approximations guarantee no false drops for range query processing. In this paper, we demonstrate how to adapt conservative approximations to ellipsoid range and *k*-nearest neighbor query processing in order to keep the number of false hits as low as possible.

Note that the conservative approximation of a given query region itself is a query region. This means that it has a boundary and a defined extension similar to the original query region. However, this idea does not extend to *k*-nearest neighbor queries, which play an important role for similarity search [Sei 97] [SK 98]. This query type does not correspond to defined query regions but is based on similarity distance functions. We have already become acquainted with an approximation technique for distance functions which has a similar importance for similarity query processing as the conservative approximations for spatial query processing, namely the lower-bounding property [FRM 94] [SK 98].

The paper is organized as follows: In section 2, efficient processing of similarity queries is described including the improvements to state-of-the-art algorithms for similarity search. Section 3 covers three approximation types: The minimum bounding box, the minimum bounding sphere and the combination of both. In section 4, we present the results of our experiments that reveal the performance enhancement. Section 5 concludes the paper.

2 Efficient Processing of Similarity Queries

The specific query types that occur in the context of similarity search are range queries, nearest neighbor queries and *k*-nearest neighbor queries. Since in current databases strong efficiency requirements have to be met, a fast processing of these complex similarity queries is crucial. As mentioned above, the evaluation time of our adaptable similarity distance functions is quadratic and, therefore, consumes a great deal of CPU-time. In order to reduce the number of expensive exact distance evaluations, we propose techniques to efficiently process similarity queries by introducing approximation-based distance evaluation. The presented algorithms work on access methods which manage the secondary storage pages by rectilinear hyperboxes, e.g. minimum bounding boxes (MBBs), in order to form higher level directory pages.

2.1 Approximation-based Similarity Range Query

The similarity range query is a fundamental query type which can be defined as follows: Let the symbol O denote the universe of all objects that may occur as database objects or query objects. For every type of similarity search, a distance function $d\colon O \times O \rightarrow \Re_0^+$ has to be provided such that $d(o_1, o_2)$ measures the (dis-)similarity of two objects o_1 and o_2. By $\mathrm{DB} \subseteq O$, let us denote an actual database. We specify similarity range queries by a query object q and a range value ε, and the answer set is defined to contain all the objects s from the database that have a distance to the query object q of less than or equal to ε:

Definition 1 (Similarity range query). For a query object $q \in O$ and a query range $\varepsilon \in \Re_0^+$, the similarity range query returns the set:

$$sim_q(\varepsilon) = \{o \in \mathrm{DB} \mid d(o, q) \leq \varepsilon\}$$

From a geometric point of view, the given distance function and the range value ε define a region around the query object q. Thus, the similarity range query reports all data objects which are contained in this region. Processing range queries on a multidimensional access method is performed as follows: The search algorithm starts from the root and then traverses the tree recursively. At each directory node, the entries (MBBs) which intersect the query region are identified and the search is directed downwards. At data nodes, all objects which are contained in the query region are finally reported. There are two query-dependent components in this algorithm: The method *intersects*(box, region) returns true if a MBB in a directory node intersects the query region, and the method *contains*(object, region) returns true if a data object is located inside the query region.

In the case of adaptable similarity models based on quadratic form distance functions, both methods have to determine the expensive exact distance of each considered object (MBB or data object) to the query region. We observed that the time for distance calculation highly affects the CPU time which in turn represents a high percentage of the overall runtime. Thus, we are strongly interested in reducing the number of exact distance evaluations. The basic idea of our approach is to adapt the concept of conservative approximations to similarity range queries. Conservative approximations of query regions totally enclose the complete query region and can efficiently be used in filter steps to generate candidates since they guarantee no false drops and ideally produce only a small number of false hits. Desired models are approximations that are less complex than the original region (which is an ellipsoid in our case) and therefore need considerably less evaluation time, if possible only linear evaluation time. By introducing conservative approximations to similarity range queries, we now can exploit the inclusion of the query region in the approximation to avoid unnecessary exact distance evaluations. Thus, the exact distance is evaluated only if the approximation of the considered object fulfills the query condition. In figure 1 we show the code of the improved *intersects*(*box, query, approx*) and *contains*(*object, query, approx*) algorithms. Note that in method *intersects* the intersection test with the exact query region could be omitted without affecting correctness. This defers the evaluation of exact distances to data nodes which could improve or decrease performance. We analyze this issue later in section 4.2.

```
method intersects (DirEntry box, Region query,
  Region approx)  →   bool;
{
  if not intersects (box, approx) then return false;
  else if not intersects (box, query) then return false;
    else return true;
}

method contains (DataEntry object, Region query,
  Region approx)  →   bool;
{
  if not contains (object, approx) then return false;
  else if not contains (object, query) then return false;
    else return true;
}
```

Figure 1: Approximation-based intersection and containment evaluation

In order to show the correctness of our approximation-based approach, we prove that the proposed algorithms guarantee no false drops.

Lemma 1. The algorithms of figure 1 produce no false drops for conservative approximations.

Proof. Given a data object *obj*, a MBB *box*, a query region $query = sim_q(\varepsilon)$ and a region *approx*. If *approx* is a conservative approximation of *query*, then $approx \supseteq query$ is true and the following implication holds:

$$\begin{aligned} & box \cap approx = \emptyset \\ \Rightarrow\ & box \cap query = \emptyset \\ \Rightarrow\ & \forall obj \in box: obj \notin sim_q(\varepsilon) \end{aligned}$$

Furthermore, for data nodes we have:

$$obj \notin approx \Rightarrow obj \notin query \Rightarrow obj \notin sim_q(\varepsilon)\ .\ \Diamond$$

Generally, we define the approximation quality Q_{approx} by the ratio of the volume of the approximation to the volume of the original region, e.g. $Q_{approx} = \frac{Vol(approx(region))}{Vol(region)}$. Thus, a larger ratio corresponds to a worse approximation quality. Obviously, the higher the quality of the conservative approximation, the higher is the performance gain in query processing time. In section 3, we will consider several promising conservative approximations models.

2.2 Approximation-based *k*-Nearest Neighbor Query

Since similarity distance functions are quite abstract, the user must be experienced with typical similarity distances in order to specify useful similarity range queries. This is the reason why *k*-nearest neighbor queries are becoming more and more important for similarity search in large databases of complex objects. The *k*-nearest neighbor query retrieves, for any query object, the *k* most similar objects from the database and can be defined as follows:

Definition 2 (*k*-nearest neighbor query). For a query object $q \in O$ and a query parameter $k \geq 1$, the *k*-nearest neighbor query returns the set $NN_q(k) \subseteq \mathrm{DB}$ that exactly contains *k* objects from the database for which the following condition holds:

$$\forall o \in NN_q(k), \forall o' \in \mathrm{DB} - NN_q(k): d(o, q) \leq d(o', q)$$

Note that possibly several objects in the database exist which have the same distance to the query object as the *k*-th object in the answer set. In this case, the *k*-th object in $NN_q(k)$ is a non-deterministic selection of one of those equally distanced objects. Several approaches to process *k*-nearest neighbor queries are available from the literature which are suitable for introducing approximation based distance evaluation, for instance [Hen 94] [RKV 95] [HS 95]. In this paper, we focus on the similarity ranking algorithm proposed in [HS 95] which is proven to be optimal with respect to the number of accessed index pages [BBKK 97] and can easily be adapted to process *k*-nearest neighbor queries by ranking exactly *k* data objects. The basic idea of this algorithm is to visit nodes in the order of their *mindist*, e.g. the minimum distance from the query object to any possible object inside a node. Although the original ranking algorithm employed the Euclidean distance function, the method works for any arbitrary distance function. The algorithm is generally designed for multidimensional access methods that hierarchically manage page regions. Therefore, it can be applied to the R-tree [Gut 84], the R^+-tree [SRF 87], the R*-tree [BKSS 90], the X-tree [BKK 96] [Ber+ 97] and many others [GG 97].

Considering the *k*-nearest neighbor algorithm, we encounter a similar situation as in the standard range query algorithm: For each considered MBB and data object, the exact distance to the query object has to be evaluated, which again has a quadratic complexity for adaptable similarity distance functions. Thus, as in the case of similarity range queries, our goal is to reduce the number of expensive distance evaluations. Obviously, we cannot adapt the concept of conservative approximations to *k*-nearest neighbor queries, since this query type does not correspond to delimited query regions. Rather, we introduce approximate distance functions to the *k*-nearest neighbor algorithm which are lower-bounds to the exact quadratic form distance function.

Formally, for any lower-bounding distance function d_{approx} of a given object distance function d_{exact} the following holds: $\forall o, q \in O\colon d_{\mathrm{approx}}(o, q) \le d_{\mathrm{exact}}(o, q)$.

We can then exploit the lower-bounding property in the following way: When the distance to MBB or a data object has to be evaluated, we first calculate the minimum distance to the query object with respect to the lower-bounding distance function d_{approx} . If this distance is less than or equal to the distance of the query object to the actual *k*-th nearest neighbor, the exact distance to the query object is evaluated using d_{exact} . If during the search process no *k*-th data object has been found yet, the exact distance of the query object to the *k*-th nearest neighbor is defined to be some value that is greater than any possible distance value in the underlying data space. Additionally, we only insert those nodes into the priority queue, which have a minimum distance less than or equal to the distance of the query object to the actual *k*-th nearest neighbor. In figure 2 we present the code of our proposed approximation-based *k*-nearest neighbor algorithm.

```
method XTree :: k_ranking (Object query,
  DistFunction d_exact, DistFunction d_approx, Integer k)
{
  PriorityQueue queue;      // node queue
  SortedList results;       // objects and distances

  queue.insert(0, root);
  while not queue.isempty() do
    Element first = queue.pop();
    if first.distance > results[k].dist then break;
    else case first isa
     DirNode:
      foreach child in first do
        if d_approx(query, child.box) ≤ results[k].dist then
          if d_exact(query, child.box) ≤ results[k].dist then
            queue.insert(d_exact(query, child.box), child);
     DataNode:
      foreach obj in first do
        if d_approx(query, obj) ≤ results[k].dist then
          if d_exact(query, obj) ≤ results[k].dist then
            results.insert(d_exact(query, obj), obj);
    end
  enddo;
  report (results, k);
}
```

Figure 2: Approximation-based *k*-nearest neighbor algorithm

The correctness of our approach is shown by the following lemma 2:

Lemma 2. The algorithm of figure 2 produces no false drops for lower-bounding distance functions.

Proof. Given a data object *obj*, a directory entry *box*, a query object *query* and two distance functions d_{exact} and d_{approx} . Let nn_k be the actual *k*-th nearest neighbor of the query object *query*. If d_{approx} is a lower-bounding distance function of d_{exact} , then for all $o, q \in O$, $d_{\mathrm{approx}}(o, q) \le d_{\mathrm{exact}}(o, q)$ is true and the following implication for directory nodes holds:

$$d_{approx}(query, box) > d_{exact}(query, nn_k)$$
$$\Rightarrow \quad d_{exact}(query, box) > d_{exact}(query, nn_k)$$
$$\Rightarrow \quad \forall obj \in box\colon d_{exact}(query, obj) > d_{exact}(query, nn_k)$$
$$\Rightarrow \quad \forall obj \in box\colon obj \notin NN_{query}(k)$$

Additionally, the following implication holds for data nodes:

$$d_{approx}(query, obj) > d_{exact}(query, nn_k)$$
$$\Rightarrow \quad d_{exact}(query, obj) > d_{exact}(query, nn_k)$$
$$\Rightarrow \quad obj \notin NN_{query}(k) \,.\ \Diamond$$

Obviously, the efficiency of our approach depends on the quality of the lower-bounding distance function. Mainly, we are interested in approximation models that yield lower-bounding distance functions which are less complex to evaluate than the original distance function. Furthermore, the maximum improvement is achieved with the greatest of all lower-bounding distance functions with respect to the selected approximation model. In the following, we propose distance functions which exactly meet these requirements.

3 Conservative Approximation Techniques

Various types of conservative approximation techniques have been investigated in the context of Geographic Information Systems and 2-D spatial database systems [BKS 93] [KSB 93], and we adapted them to our ellipsoid queries in *d*-dimensional spaces. Both the *Minimum Bounding Box* (MBB) and the *Minimum Bounding Sphere* (MBS) require only $O(d)$ space and $O(d)$ time for testing intersections and containments. The *Convex Hull* as well as *Minimum Bounding n-Corners* mismatch the spherical character of ellipsoids. In comparison with the MBB, the *Rotated Minimum Bounding Box* (RMBB) is not restricted to be rectilinear which, in general, yields a better approximation quality. However, the RMBB requires $O(d^2)$ space to represent its orientation in the *d*-dimensional space, and the computation of intersections, containments and distances is, at best, performed by linear programming in $O(d^{\lceil d/2 \rceil})$ [PTVF 92] or $O(d!)$ [Sei 90] time. Thus, the RMBB is not expected to be beneficial when approximating ellipsoids, and we concentrate on the MBB and the MBS as the most promising approximation techniques. On top of these basic approximations, we demonstrate how to combine them to exploit the advantages of both.

Each technique, MBB and MBS as well as the combined approximation, are applied to both similarity range queries and *k*-nearest neighbor queries. For this purpose, we have to provide two instances for each model: First, the conservative approximation itself which represents a geometric region enclosing the query ellipsoid, and second, an approximate distance function that lower-bounds the respective quadratic form distance function. Since both instances are closely related, we focus mainly on the more general case of lower-bounding distance functions.

Whereas we already defined the ellipsoid distance function $d_A^2(p, q)$ to be a quadratic form, we additionally introduce the symbol $\mathrm{ellip}(A, q, \varepsilon)$ to represent an ellipsoid of level ε around a query point q as query region:

$$\mathrm{ellip}(A, q, \varepsilon) = \{p \in \Re^d : d_A^2(p, q) \le \varepsilon\}$$

3.1 Minimum Bounding Box Approximation

The *Minimum Bounding Box* (MBB) of a spatial object is the smallest rectilinear box that totally encloses the object. The MBB is a favorite approximation technique due to its compact representation which requires only $2 \cdot d$ parameters in d-dimensional spaces since it suffices to store the lower and upper bound in each dimension. It is easy to determine and highly compatible to rectilinearly organized multidimensional access methods. Figure 3 provides a 2-D example.

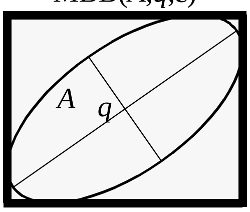

Figure 3: Minimum Bounding Box (MBB) of a 2-D ellipsoid of level ε.

The $\mathrm{MBB}(A, q, \varepsilon)$ of an ellipsoid $\mathrm{ellip}(A, q, \varepsilon)$ may be computed by determining the tangential hyperplanes whose normal vectors are parallel to the coordinate axes. Thus, we obtain for the i-th component of $\mathrm{MBB}(A, q, \varepsilon)$:

$$\mathrm{MBB}(A, q, \varepsilon)_i = [q_i - \sqrt{\varepsilon \cdot (A^{-1})_{ii}}, q_i + \sqrt{\varepsilon \cdot (A^{-1})_{ii}}]$$

We defer the formal derivation of this formula since the same result is immediately obtained from the corresponding lower-bounding box distance function which we derive in the following.

Lower-Bounding Box Distance Function. The generalization of boxes to distance functions involves a weighted maximum norm L_∞ which corresponds to rectilinear rectangular query regions (cf. figure 4). The common case of non-square rectangles is represented by involving weighting factors for the individual dimensions. The following definition formalizes the minimum bounding box distance function as required for our purpose.

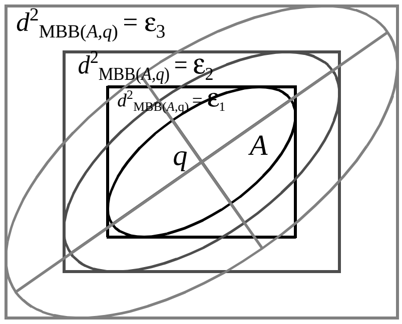

Figure 4: Greatest lower-bounding box distance function

Definition 3 (MBB distance function). Let A be a similarity matrix, and A^{-1} its inverse. The *minimum bounding box distance function* $d^2_{\mathrm{MBB}(A)}$ of A is defined as follows:

$$d^2_{\mathrm{MBB}(A)}(p, q) = \max_{i=1}^{d} \left(\frac{(p_i - q_i)^2}{(A^{-1})_{ii}} \right)$$

Note that $d^2_{\mathrm{MBB}(A)}$ is well-defined since A^{-1} exists for every positive definite matrix A. The following theorem indicates that the MBB distance function represents a lower bound of the original ellipsoid distance function.

Theorem. For every similarity matrix A and every $p, q \in \Re^d$, the MBB distance function $d^2_{\mathrm{MBB}(A)}$ is a lower bound of the ellipsoid distance function d_A^2:

$$d^2_{\mathrm{MBB}(A)}(p, q) \le d_A^2(p, q)$$

Proof. We show that for every $p, q \in \Re^d$, an intermediate point p_0 exists such that the following formula is true which immediately implies the proposition:

$$d^2_{\mathrm{MBB}(A)}(p, q) = d_A^2(p_0, q) \le d_A^2(p, q)$$

Figure 5 demonstrates the existence of such an auxiliary point p_0, given as the tangential point of the box and the ellipsoid of which the box is the MBB. This definition implies $d^2_{\mathrm{MBB}(A)}(p_0, q) = d_A^2(p_0, q)$. Obviously, p_0 is located on the same box as p, i.e. $d^2_{\mathrm{MBB}(A)}(p_0, q) = d^2_{\mathrm{MBB}(A)}(p, q)$, and the ellipsoid of p_0 is smaller than the ellipsoid on which p is located, i.e. $d_A^2(p_0, q) \le d_A^2(p, q)$. From these considerations, the proposition follows immediately. A formal proof is provided in the appendix of this paper. ◊

The fact that p itself may be the tangential point p_0 shows that $d^2_{\mathrm{MBB}(A)}(p, q)$ can reach $d_A^2(p, q)$. This case indicates that $d^2_{\mathrm{MBB}(A)}$ represents the greatest of all box-shaped lower-bounding distance functions. As a consequence, $d^2_{\mathrm{MBB}(A)}$ guarantees the best filtering quality that can be achieved for lower-bounding distance functions that are based on a weighted maximum norm.

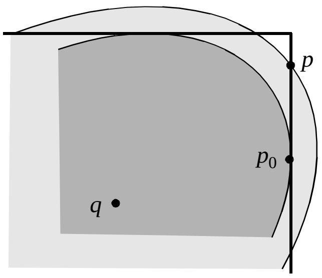

Figure 5: The auxiliary point p_0 shares its box distance with p but is located on a smaller ellipsoid than p. At p_0, the box and ellipsoid distances are equal: $d_A^2(p_0, q) = d^2_{\mathrm{MBB}(A)}(p_0, q)$

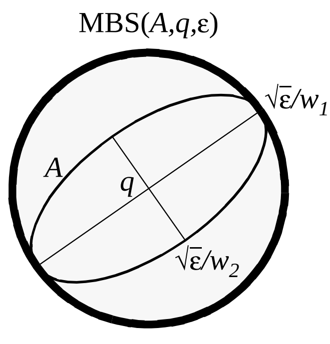

Figure 6: Minimum Bounding Sphere (MBS) of an ellipsoid $\mathrm{ellip}(A, q, \varepsilon)$. The radius of the MBS depends on the smallest eigenvalue $w^2_{\min}$ of A, and on the level ε.

Geometry of the Minimum Bounding Box. Concluding the introduction of MBB approximations, we demonstrate how the MBB of an ε-ellipsoid is obtained from the MBB distance function. Lemma 3 shows that an ε-range of the MBB distance function actually represents the minimum bounding box $\mathrm{MBB}(A, q, \varepsilon)$ of the corresponding ellipsoid.

Lemma 3. For every similarity matrix A, query point q, and range parameter ε, the MBB distance range $\{p \mid d^2_{\mathrm{MBB}(A)}(p, q) \le \varepsilon\}$ exactly represents the minimum bounding box $\mathrm{MBB}(A, q, \varepsilon)$ of the ellipsoid $\mathrm{ellip}(A, q, \varepsilon)$.

Proof. For all p, the following equivalences are true:

$$d^2_{\mathrm{MBB}(A)}(p, q) \le \varepsilon \quad \Leftrightarrow$$

$$\Leftrightarrow \quad \max_{i=1}^{d}\left(\frac{(p_i - q_i)^2}{(A^{-1})_{ii}}\right) \le \varepsilon \quad \Leftrightarrow \quad \forall i: \frac{(p_i - q_i)^2}{(A^{-1})_{ii}} \le \varepsilon \quad \Leftrightarrow$$

$$\Leftrightarrow \quad \forall i: q_i - \sqrt{\varepsilon \cdot (A^{-1})_{ii}} \le p_i \le q_i + \sqrt{\varepsilon \cdot (A^{-1})_{ii}} \quad \Leftrightarrow$$

$$\Leftrightarrow \quad p \in \mathrm{MBB}(A, q, \varepsilon) . \Diamond$$

3.2 Minimum Bounding Sphere Approximation

The *Minimum Bounding Sphere* (MBS) of a spatial object is the smallest sphere that totally encloses the object. The MBS requires only $d + 1$ parameters in d-dimensional spaces to store the radius and the d coordinates of the center point. For ellipsoids, the center of the MBS coincides with the center of the ellipsoid. Figure 6 provides an example in the 2-D.

Lower-Bounding Sphere Distance Function. Also for the MBS approximation model, we provide a distance function $d^2_{\mathrm{MBS}(A)}$ that lower-bounds the ellipsoid distance function d^2_A. An appropriate generalization of spheres to distance functions leads to the Euclidean distance which is scaled by a factor that corresponds to the radius of the sphere.

Definition 4 (MBS distance function). Let A be a similarity matrix, and $w^2_{\min}$ the minimum eigenvalue of A. The *minimum bounding sphere distance function* $d^2_{\mathrm{MBS}(A)}$ of A is defined to be the scaled and squared Euclidean distance function:

$$d^2_{\mathrm{MBS}(A)}(p, q) = w^2_{\min} \cdot (p - q)^2$$

Theorem. For every similarity matrix A, the MBS distance function $d^2_{\mathrm{MBS}(A)}$ of A is a lower bound of the ellipsoid distance function d^2_A, i.e. for all $p, q \in \mathfrak{R}^d$ the following holds:

$$d^2_{\mathrm{MBS}(A)}(p, q) \le d^2_A(p, q)$$

Proof. Since the matrix A is positive definite, the diagonalization $A = V \cdot W \cdot V^T$ exists where $V \cdot V^T = Id$, and the diagonal matrix $W = \mathrm{diag}(w_1^2, \ldots, w_d^2)$ consists of the eigenvalues $w_1^2, \ldots, w_d^2$ of A. When considering the minimum $w^2_{\min}$ of these eigenvalues, we obtain:

$$d^2_A(p, q) = (p - q) \cdot V \cdot W \cdot V^T \cdot (p - q)^T =$$

$$= \sum_{i=1}^{d} w_i^2 \cdot (pV - qV)_i^2 \ \ge \ \sum_{i=1}^{d} w^2_{\min} \cdot (pV - qV)_i^2 =$$

$$= w^2_{\min} \cdot (p - q) \cdot V \cdot V^T \cdot (p - q)^T = d^2_{\mathrm{MBS}(A)}(p, q) . \Diamond$$

Note that for a certain p, $d^2_{\mathrm{MBS}(A)}(p, q)$ reaches $d^2_A(p, q)$ and, therefore, $d^2_{\mathrm{MBS}(A)}$ represents the greatest lower-bounding distance function of the spherical type. This optimality criterion ensures the best approximation quality that could be achieved for the type of scaled Euclidean distance functions.

Geometry of the Minimum Bounding Sphere. For a given center point c and radius r, the sphere is represented by the function $\mathrm{sphere}_{c,r}(p) = (p - c)^2 / r^2$, and the inequality:

$$\mathrm{sphere}_{c,r}(p) \le 1 \quad \Leftrightarrow \quad (p - c)^2 \le r^2$$

It remains to determine the radius of the minimum circumscribing sphere. Observe that the minimum bounding sphere in particular touches the ellipsoid, i.e. the ellipsoid and its MBS have some points in common. A necessary condition which holds for all touching and smooth surfaces is that the normal vectors of the objects at any tangential point are linearly dependent, i.e. parallel. The normal vector of a surface is equal to the gradient of the corresponding surface function. We set the gradient of the centered sphere function, p^2/r^2, in relation to the gradient of the centered ellipsoid function, $p \cdot A \cdot p^{\mathrm{T}}$, in order to state the linear dependency:

$$\begin{aligned} \nabla \mathrm{ellip}_A(p) &= \lambda \cdot \nabla \mathrm{sphere}_r(p) \\ 2 \cdot p \cdot A &= \lambda \cdot 2 \cdot p / r^2 \\ p \cdot A &= \lambda / r^2 \cdot p \end{aligned}$$

By means of linear algebra, this statement says that λ / r^2 is one of the eigenvalues $w_1^2, \ldots, w_d^2$ of the matrix *A*, and the tangential point *p* corresponds to an eigenvector. Since we are not yet aware of the value of λ, we assume the equality of the spherical function and the ellipsoid function for every tangential point p_t that fulfills the above gradient equation:

$$\mathrm{ellip}_A(p_t) = \mathrm{sphere}_r(p_t) \Leftrightarrow p_t \cdot A \cdot p_t^T = p_t^2 / r^2 \Leftrightarrow$$
$$\lambda / r^2 \cdot p_t \cdot p_t^T = p_t^2 / r^2 \Leftrightarrow \lambda = 1 .$$

Thus, every eigenvalue of the matrix *A* directly represents the reciprocal square of a radius $1/r^2$ that belongs to the corresponding tangential point. Vice versa, the candidate values for the radius are given by the reciprocal square roots of the eigenvalues of A, that is $1/w_1, \ldots, 1/w_d$. Since we have to determine the bounding sphere as a conservative approximation of a given ellipsoid region, that is: $\mathrm{ellip}_A(p_t) \le \varepsilon$, we have to select the maximum radius $r_{\max}$ that occurs over all tangential points to obtain the sphere $p^2/r^2 \le \varepsilon \Leftrightarrow p^2 \le \varepsilon \cdot r^2$. This requirement immediately implies that we have to choose the minimum eigenvalue of *A* for the computation of the desired radius *r* of the minimum bounding sphere:

$$r_{\mathrm{MBS}(A, \varepsilon^2)} = \max\left(\frac{\sqrt{\varepsilon}}{w_1}, \ldots, \frac{\sqrt{\varepsilon}}{w_d}\right) = \frac{\sqrt{\varepsilon}}{w_{\min}}$$

3.3 Combined Conservative Approximations

Both the MBB and MBS approximation have specific characteristics with respect to their approximation quality and their potential of improving query processing efficiency. In order to exploit the advantages of both techniques, it is near at hand to look for combinations of these basic approximations. In the following, we demonstrate how basic conservative approximations are combined to complex approximations, and how to combine basic lower-bounding distance functions to complex ones.

Combination of Approximations. Given an ellipsoid *ellip*(*A*, *q*, ε), let $C = \{\mathrm{APP}(A, q, \varepsilon)\}$ be a set of conservative approximations of *ellip*, e.g. $C = \{\mathrm{MBB}(A, q, \varepsilon), \mathrm{MBS}(A, q, \varepsilon)\}$. By the following lemma 4 we show that the intersection of the approximations of *C* again is a conservative approximation. For the proof, we exploit the property that each of the conservative approximations totally encloses the original object, and hence, their intersection also encloses the object:

Lemma 4. Given an ellipsoid *ellip*(*A*, *q*, ε), let $C = \{\mathrm{APP}(A, q, \varepsilon)\}$ be a set of conservative approximations of *ellip*. Then, the intersection of all APP(*A*, *q*, ε) is again a conservative approximation of *ellip*:

$$\bigcap_{\mathrm{APP} \in C} \mathrm{APP}(A, q, \varepsilon) \supseteq \mathrm{ellip}(A, q, \varepsilon)$$

Proof. Since every $\mathrm{APP} \in C$ is a conservative approximation of *ellip*, it fulfills the relationship $\mathrm{APP}(A, q, \varepsilon) \supseteq \mathrm{ellip}(A, q, \varepsilon)$ which is equivalent to the implication

$$\forall p \in \mathfrak{R}^d : p \in \mathrm{ellip}(A, q, \varepsilon) \Rightarrow p \in \mathrm{APP}(A, q, \varepsilon)$$

This implication is true for all $\mathrm{APP} \in C$ and, hence, also for the intersection of the APPs. Overall, we obtain the following implication which is equivalent to the proposition as it holds for every $p \in \mathfrak{R}^d$:

$$p \in \mathrm{ellip}(A, q, \varepsilon) \Rightarrow p \in \bigcap_{\mathrm{APP} \in C} \mathrm{APP}(A, q, \varepsilon) \ \Diamond$$

Figure 7 shows a 2-D example for a conservative approximation that combines the minimum bounding box (MBB) and the minimum bounding sphere (MBS) approximation of an ellipsoid. Obviously, the volume of the intersection is smaller than the volumes of the individual components which results in an improved approximation quality in comparison with the basic approximations.

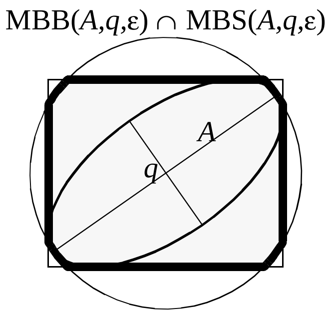

Figure 7: Combined approximation (here: MBB and MBS) of an ellipsoid of level ε.

Combination of Lower-Bounding Distance Functions. Analogously to the approximation techniques mentioned above, we present a combination of lower-bounding distance functions that again lower-bounds the exact similarity distance function. By the subsequent formal proposition, we show that the maximum of the component distance functions fulfills this requirement.

Definition 5 (Combined distance function). Let $C = \{d_i^2\}$ be a set of distance functions. Then, the combined distance function d_C^2 is defined to be the maximum of the component functions:

$$d_C^2(p, q) = \max\{d_i^2(p, q)\}$$

Theorem. For every similarity matrix A and every set of lower-bounding distance functions $C = \{d^2_{\mathrm{APP}(A)}\}$, i.e. $d^2_{\mathrm{APP}(A)}(p, q) \le d_A^2(p, q)$ for all $p, q \in \Re^d$, the combined distance function d_C^2 is a lower bound of the ellipsoid distance function d_A^2, and, for all $p, q \in \Re^d$ it holds that:

$$d_C^2(p, q) \le d_A^2(p, q)$$

Proof. For all $p, q \in \Re^d$, the following equivalences are true: $d_C^2(p, q) \le d_A^2(p, q) \Leftrightarrow \max\{d^2_{\mathrm{APP}(A)}(p, q)\} \le d_A^2(p, q)$ $\Leftrightarrow \forall d^2_{\mathrm{APP}(A)}: d^2_{\mathrm{APP}(A)}(p, q) \le d_A^2(p, q)$. The final inequality represents the precondition. ◊

In particular, the maximum distance function is the greatest lower-bounding distance function that can be derived from a set of distance functions since it always returns the greatest value of all component functions. This maximum property guarantees an optimal selectivity and, therefore, yields the best performance improvement for *k*-nearest neighbor query processing.

4 Experimental Evaluation

In the experimental evaluation, we applied our approximation techniques to a large image database, containing 8-D color histograms of 112,000 images as well as to a database of 1,000,000 objects that are uniformly distributed in the 8-D. The experiments were performed on an HP-735 under HP-UX 10.20. The approximation techniques will be denoted by BOX for box approximation, SPHERE for sphere approximation, and COMB for the combination of BOX and SPHERE. The symbol NONE stands for the pure exact ellipsoid evaluation without using any approximation.

All similarity matrices we applied were derived from our color similarity search system. In the context of this system, the user can specify the four parameters σ, w_r, w_g, and w_b from which the components a_{ij} of the similarity matrix A are determined by the following formula from [Haf+ 95]:

$$a_{ij} = e^{-\sigma \cdot (d_w(c_i, c_j)/d_{max})^2}$$

Thus, σ is a positive constant that affects the overall shape of the query ellipsoid, and $d_w(c_i, c_j)$ represents the weighted Euclidean distance of the basic colors c_i and c_j. The weighting factors $w = (w_r, w_g, w_b)$ denote the relative weight of the red, green, and blue component in the RGB color space. In the following, we specify our similarity matrices by these four parameters.

Since the performance aspect is a basic motivation for our approach, we first show the high impact of the quadratic evaluation time for an ellipsoid function on the total query time (cf. figure 8). For this experiment, we used different matrices (cf. table 1) to perform 100 different range queries as well as 100 different 5-nearest neighbor queries. The measured average percentage of the evaluation time for the corresponding ellipsoid function compared with the total query time was as high as 74%. Such a high percentage of the evaluation time clearly underlines the relevance for efficiency improvements.

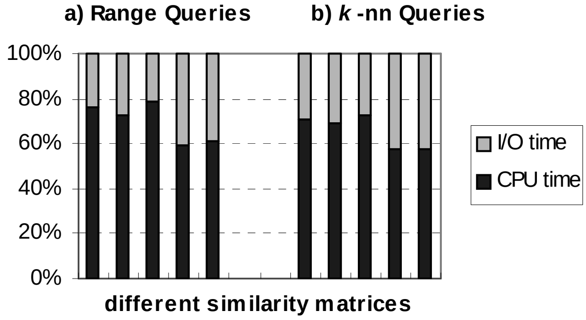

Figure 8: For adaptable similarity search, CPU time is a high percentage of the overall runtime. a) Range queries on ImageDB, b) *k*-nn queries (*k*=5) on ImageDB.

Matrix	σ	w_r	w_g	w_b
ZT11	10	1,000	1	1
Z711	10	700	1	1
Z411	10	400	1	1
ZZ11	10	10	1	1
Z111	10	1	1	1

Table 1: User-defined parameters for the matrices used in our experiments

4.1 Approximation Quality

In our further experiments, we measured the performance of our approximation algorithms with respect to their dependency on different similarity matrices. Since the effects and performance of an approximation is mainly influenced by the shape of the corresponding ellipsoid, we characterize the corresponding ellipsoid through a geometric measure instead of user-defined parameters.

For explaining the quality of the sphere approximation, we denote *sphericity* as the ratio of the volume of the sphere divided by the volume of the ellipsoid, which complies with the definition of the approximation quality in section 2. This means a sphericity of about 1 characterizes a similarity matrix almost representing a sphere, whereas a high sphericity value indicates that the minimum bounding sphere is considerably larger than the ellipsoid.

To demonstrate the quality of the box approximation, two measures seem to be adequate. First, the approximation quality of the minimum bounding box can be used for our purposes. The disadvantage of this measure is that it does not consider the obliqueness of the ellipsoid which obviously affects the approximation quality. Therefore, a second possible measure is the volume ratio of the minimum bounding box and the rotated minimum bounding box.

The influence of all these matrices on the parameters is reflected in Figure 9. We can ascertain for our different matrices that matrices with high values of sphericity also have high values in the two measures for the minimum bound box quality. Similarly, matrices with low values of sphericity have low values in each measure. In the following, we will use the parameter sphericity for describing the matrices used in our experiments.

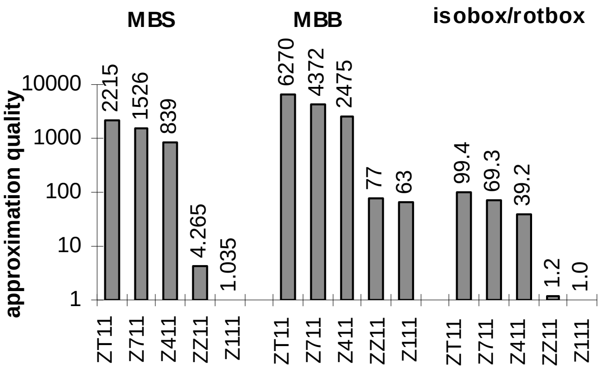

Figure 9: The relative volume of the approximations (approximation quality) are used as shape parameters of ellipsoids.

4.2 Approximations and Exact Evaluations in the Directory

Considering the algorithms of section 2, the question emerges if evaluating exact distances solely in data nodes but not in directory nodes could be more efficient than our approach. Obviously, deferring exact evaluations to data nodes results in a reduced evaluation time per directory node. However, as directory nodes are not exactly evaluated, the effect of this approach is that a larger number of data nodes have to be tested. Thus, the decision to evaluate exact distances only in data nodes is a trade off between a reduction of computation time in the index and an increased number of data nodes that are evaluated. To analyze this effect, we performed a test of range queries for various query ranges, and the similarity matrix corresponds to an ellipsoid with sphericity 1.035. As figure 10 depicts, evaluating the exact distance in both directory nodes and data nodes yields a better overall time in comparison with restricting the exact distance evaluation to data nodes.

4.3 Dependency on the Similarity Matrix

For our next experiments, we performed a sample of range queries for different similarity matrices corresponding to ellipsoids having a sphericity of 1.035 up to 2,200. On both databases, the image database as well as the uniformly distributed data, the range queries returned between 1 and 10 results on the average. Figure 11 depicts the per-

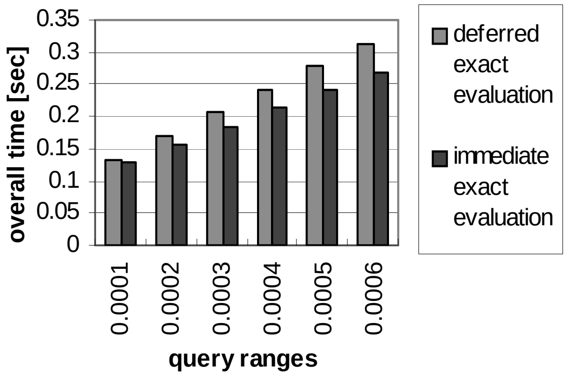

Figure 10: Comparison of deferred and immediate exact evaluations in the index (example: image database).

centage of exact ellipsoid evaluations that were saved by using the approximation techniques, due to approximation based exclusions. For the image database, more than 90% of the ellipsoid evaluations are avoided in all of our experiments. In case of uniformly distributed data, 90% of ellipsoid evaluations are avoided only for ellipsoids that are quite similar to spheres, and for less spherical ellipsoids, still 20% to 60% of the expensive ellipsoid evaluations are avoided. Obviously, the combined approximation yields the most savings. So we have found out that our approximation yields a very high percentage of saved exact evaluations. Next, we investigated the result of the savings.

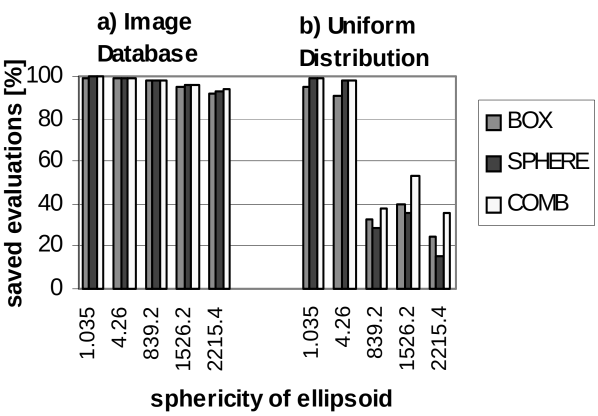

Figure 11: Saved evaluations of intersection and containment tests for range queries using similarity matrices that correspond to ellipsoids with different sphericities.

In figure 12, the impact of avoiding exact ellipsoid evaluations on the elapsed time is illustrated for the same sample of range queries as above. For the image database, the factor of performance improvement ranges from 2.8 to 6.3, depending on the sphericity of the ellipsoid. For the uniformly distributed data, we observed the same improvement factor of 6 only for almost spherical ellipsoids. For higher sphericity values, the approximation quality is worse and in some cases, it would be better off to directly test the exact ellipsoid tests without using approximations. An optimizer could use this information in order to decide which approximation should be used, if any, depending on the shape of the query ellipsoid.

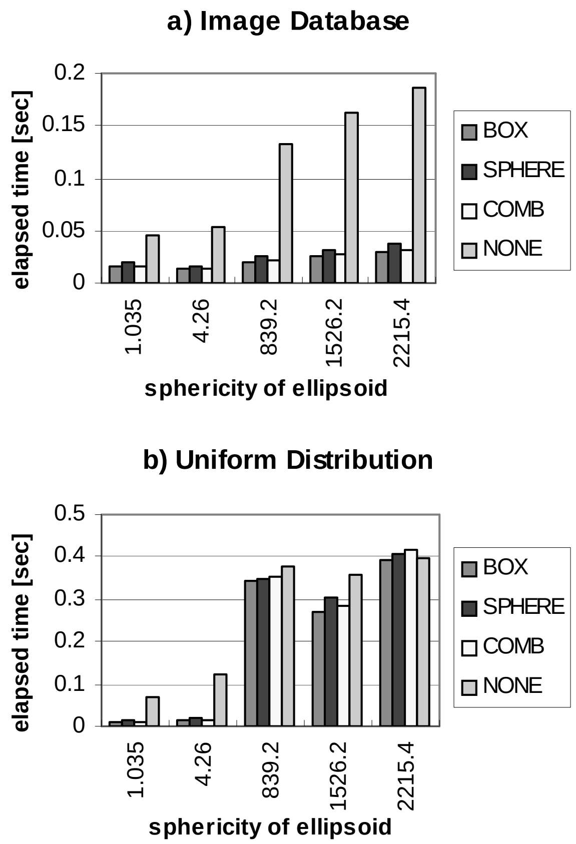

Figure 12: Elapsed time for range queries depending on the sphericity of the query ellipsoid.

4.4 Dependency on Query Parameters

For our next series of experiments, we show the robustness of our approximation approach concerning different query types. Therefore, we performed samples of range queries and *k*-nearest neighbor queries for various query ranges and query parameters *k*. The similarity matrix corresponds to an ellipsoid with sphericity 1.035. Figure 13 depicts the elapsed time for query processing depending on the average number of results that are returned by the range queries. On average, the used query ranges return 2.8 to 19 results from the image database and 5.2 to 50.6 results from the uniformly distributed data. In these experiments, the approximations outperform the pure ellipsoid evaluation by a factor of 2.7 (image database) and 4.2 to 6.3 (uniform distribution).

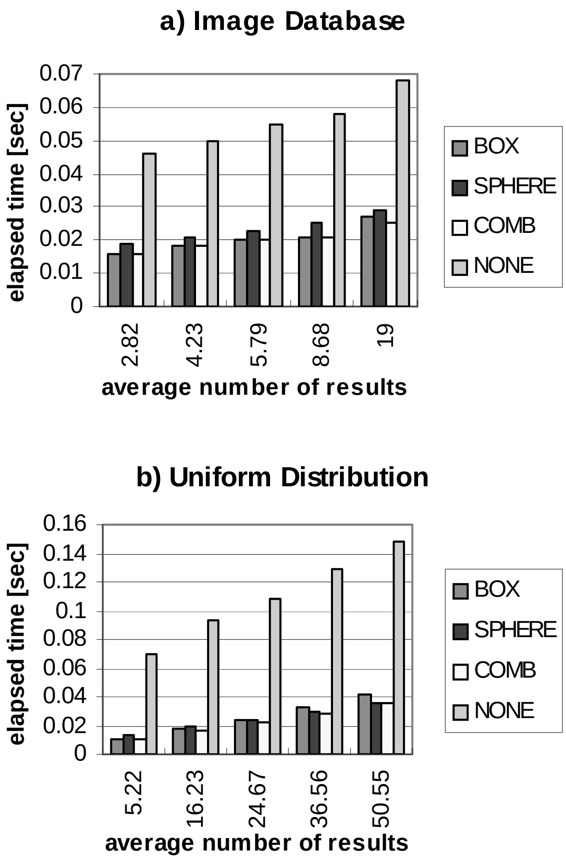

Figure 13: Elapsed time for range queries depending on the query range.

In figure 14, we demonstrate the improvement that we achieved for *k*-nearest neighbor queries for a varying value of *k*. For the image database, we achieved a performance gain of approximately 40% for the MBS approximation, and for the uniform distribution an acceleration of 35% to 40%.

5 Conclusions

In this paper, we investigated the efficiency of adaptable similarity search as it occurs in a variety of modern database applications including multimedia, molecular biology, medical imaging, and CAD/CAM. Based on the observation that the exact evaluation of the underlying quadratic form distance functions consumes a high percentage of the overall search time, we developed an approximation-based approach for improving the performance of similarity query processing. We adapted the concept of conservative approximations in order to accelerate similarity range queries, and, in particular, investigated the Minimum Bounding Box (MBB), the Minimum Bounding Sphere (MBS), and the combination of these two approximations. Additionally, we extended the concepts of these approximation types to

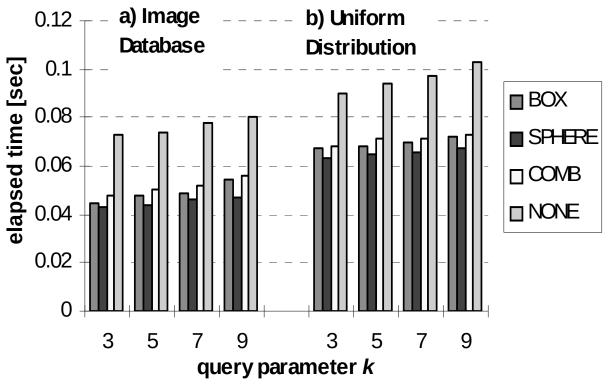

Figure 14: Elapsed time for *k*-nearest neighbor queries for various values of *k*.

k-nearest neighbor queries. These queries are directly based on similarity distance functions rather than on geometric query regions. For this purpose, we developed greatest lower-bounding distance functions for each of the considered approximation types. In a detailed analysis, we proved the correctness of our techniques. For our experiments, we used an image database containing 112,000 color histograms, and a synthetic database containing 1,000,000 uniformly distributed 8-D points. The results demonstrate that by using the approximation techniques, a high percentage of the expensive exact evaluations can be avoided, depending on the data, on the similarity matrix, and on the query parameters. We observed an improvement of the CPU time by factors between 2 and 6 for range queries, and between 1.4 and 1.7 for *k*-nearest neighbor queries.

In our future work, we plan to investigate the impact of the similarity matrix, i.e. the geometry of the query ellipsoid, on the performance of similarity query processing. Provided with this knowledge, a query optimizer can be developed that is able to select the most efficient execution plan that may or may not include approximations for similarity search.

Acknowledgements

We gratefully thank the anonymous reviewers for their detailed comments that helped us to improve the presentation of our concepts.

References

[AFS 93] Agrawal R., Faloutsos C., Swami A.: *'Efficient Similarity Search in Sequence Databases'*, Proc. 4th. Int. Conf. on Foundations of Data Organization and Algorithms (FODO'93), Evanston, ILL, in: Lecture Notes in Computer Science, Vol. 730, Springer, 1993, pp. 69-84.

[AKS 98] Ankerst M., Kriegel H.-P., Seidl T.: *'Pixel-based Shape Similarity Search in Large Image Databases'*, submitted for publication.

[ALSS 95] Agrawal R., Lin K.-I., Sawhney H. S., Shim K.: *'Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases'*, Proc. 21th Int. Conf. on Very Large Databases (VLDB'95), Morgan Kaufmann, 1995, pp. 490-501.

[BBKK 97] Berchtold S., Böhm C., Keim D., Kriegel H.-P.: *'A Cost Model for Nearest Neighbor Search in High-Dimensional Data Spaces'*, Proc. 16th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS), Tucson, AZ, 1997, pp. 78-86.

[Ber+ 97] Berchtold S., Böhm C., Braunmüller B., Keim D., Kriegel H.-P.: *'Fast Parallel Similarity Search in Multimedia Databases'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, Tucson, AZ, 1997, Best Paper Award, pp. 1-12.

[BKK 96] Berchtold S., Keim D., Kriegel H.-P.: *'The X-tree: An Index Structure for High-Dimensional Data'*, Proc. 22nd Int. Conf. on Very Large Data Bases (VLDB'96), Mumbai, India, 1996, pp. 28-39.

[BK 94] Brinkhoff T., Kriegel H.-P.: *'Approximations for a Multi-Step Processing of Spatial Joins'*, Proc. Int. Workshop on Advanced Research in Geographic Information Systems, Monte Verita, Ascona, Switzerland, 1994, in: Lecture Notes in Computer Science, Vol. 884, 1994, pp. 25-34.

[BK 97] Berchtold S., Kriegel H.-P.: *'S3: Similarity Search in CAD Database Systems'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, 1997.

[BKS 93] Brinkhoff T., Kriegel H.-P., Schneider R.: *'Comparison of Approximations of Complex Objects Used for Approximation-based Query Processing in Spatial Database Systems'*, Proc. 9th Int. Conf. on Data Engineering, Vienna, Austria, 1993, pp. 40-49.

[BKSS 90] Beckmann N., Kriegel H.-P., Schneider R., Seeger B.: *'The R*-tree: An Efficient and Robust Access Method for Points and Rectangles'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, 1990, pp. 322-331.

[Bri 94] Brinkhoff T.: *'Spatial Join in Spatial Database Systems'*, Ph.D. Thesis, Institute for Computer Science, University of Munich, 1994. (in German)

[Fal+ 94] Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., Petkovic D., Equitz W.: *'Efficient and Effective Querying by Image Content'*, Journal of Intelligent Information Systems, Vol. 3, 1994, pp. 231-262.

[FRM 94] Faloutsos C., Ranganathan M., Manolopoulos Y.: *'Fast Subsequence Matching in Time-Series Databases'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, 1994, pp. 419-429.

[GG 97] Gaede V., Günther O.: *'Multidimensional Access Methods'*, ACM Computing Surveys, 1997.

[GM 93] Gary J. E., Mehrotra R.: *'Similar Shape Retrieval using a Structural Feature Index'*, Information Systems, Vol. 18, No. 7, 1993, pp. 525-537.

[Gut 84] Guttman A.: *'R-trees: A Dynamic Index Structure for Spatial Searching'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, Boston, MA, 1984, pp. 47-57.

[Haf+ 95] Hafner J., Sawhney H. S., Equitz W., Flickner M., Niblack W.: *'Efficient Color Histogram indexing for Quadratic Form Distance Functions'*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 17, No. 7, 1995, pp. 729-736.

[Hen 94] Henrich, A.: *'A Distance-Scan Algorithm for Spatial Access Structures'*, Proc. 2nd ACM Workshop on Advances in Geographic Information Systems, Gaithersburg, Maryland, 1994, pp. 136-143.

[HS 95] Hjaltason G. R., Samet H.: *'Ranking in Spatial Databases'*, Proc. 4th Int. Symposium on Large Spatial Databases (SSD'95), Lecture Notes in Computer Science, Vol. 951, Springer, 1995, pp. 83-95.

[Jag 91] Jagadish H. V.: *'A Retrieval Technique for Similar Shapes'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, 1991, pp. 208-217.

[Kor+ 96] Korn F., Sidiropoulos N., Faloutsos C., Siegel E., Protopapas Z.: *'Fast Nearest Neighbor Search in Medical Image Databases'*, Proc. 22nd VLDB Conference, Mumbai, India, 1996, pp. 215-226.

[KS 98] Kriegel H.-P., Seidl T.: *'Approximation-Based Similarity Search for 3-D Surface Segments'*, GeoInformatica Journal, Kluwer Academic Publishers, 1998, to appear.

[KSB 93] Kriegel H.-P., Schneider R., Brinkhoff T.: *'Potentials for Improving Query Processing in Spatial Database Systems'*, invited talk, Proc. 9èmes Journées Bases de Données Avancées (9th Conference on Advanced Databases), Toulouse, France, 1993.

[KSS 97] Kriegel H.-P., Schmidt T., Seidl T.: *'3D Similarity Search by Shape Approximation'*, Proc. Fifth Int. Symposium on Large Spatial Databases (SSD'97), Berlin, Germany, Springer LNCS 1262, 1997, pp.11-28.

[PTVF 92] Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.: *'Numerical Recipes in C'*, 2nd ed., Cambridge University Press, 1992.

[RKV 95] Roussopoulos N., Kelley S., Vincent F.: *'Nearest Neighbor Queries'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, 1995, pp. 71-79.

[Sei 90] Seidel R.: *'Linear Programming and Convex Hulls Made Easy'*, Proc. 6th Annual Symp. on Computational Geometry, Berkeley, CA, 1990, pp. 211-215.

[Sei 97] Seidl T.: *'Adaptable Similarity Search in 3-D Spatial Database Systems'*, Ph.D. Thesis, Institute for Computer Science, University of Munich, 1997; Herbert Utz Publishers, Munich, Germany, http://utzverlag.com, ISBN: 3-89675-327-4.

[SK 97] Seidl T., Kriegel H.-P.: *'Efficient User-Adaptable Similarity Search in Large Multimedia Databases'*, Proc. 23rd Int. Conf. on Very Large Databases (VLDB'97), Athens, Greece, 1997, pp. 506-515.

[SK 98] Seidl T., Kriegel H.-P.: *'Optimal Multi-Step k-Nearest Neighbor Search'*, Proc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, Washington, 1998.

[SRF 87] Sellis T., Roussopoulos N., Faloutsos C.: *'The R+-Tree: A Dynamic Index for Multi-Dimensional Objects'*, Proc. 13th Int. Conf. on Very Large Databases, Brighton, England, 1987, pp 507-518.

Appendix

Formal Proof of the MBB Theorem in Section 3.1:

For every $p, q \in \Re^d$, we show the existence of an auxiliary point p_0 for which the following formula is true:

$$d^2_{\mathrm{MBB}(A)}(p, q) \;=\; d^2_A(p_0, q) \;\le\; d^2_A(p, q) \quad (*)$$

Let j be the index of the component of the difference vector $p - q$ that has the maximum value. Then, the MBB distance function appears as:

$$d^2_{\mathrm{MBB}(A)}(p, q) \;=\; \frac{(p_j - q_j)^2}{(A^{-1})_{jj}}$$

Now we introduce the desired intermediate point p_0 by the following definition where e_j denotes the j-th unit vector:

$$p_0 \;=\; q + \frac{(p_j - q_j)}{(A^{-1})_{jj}} \cdot e_j \cdot A^{-1}$$

At this point, we are prepared to establish the left hand side equation of (*) that we proposed at the beginning of the proof:

$$\text{(i)} \quad d^2_A(p_0, q) \;=\; (p_0 - q) \cdot A \cdot (p_0 - q)^{\mathrm{T}} \;=$$

$$= \frac{(p_j - q_j)}{(A^{-1})_{jj}} \cdot e_j \cdot A^{-1} \cdot A \cdot (A^{-1})^{\mathrm{T}} \cdot e_j^{\mathrm{T}} \cdot \frac{(p_j - q_j)}{(A^{-1})_{jj}} \;=$$

$$= \frac{(p_j - q_j)}{(A^{-1})_{jj}} \cdot (A^{-1})_{jj} \cdot \frac{(p_j - q_j)}{(A^{-1})_{jj}} \;=\; \frac{(p_j - q_j)^2}{(A^{-1})_{jj}} \;=$$

$$= d^2_{\mathrm{MBB}(A)}(p, q)\,.$$

In order to prove the estimation on the right hand side of (*), let us represent the vector p by $p_0 + \Delta p$, and expand the ellipsoid distance function as follows:

$$\text{(ii)} \quad d^2_A(p, q) \;=\; (p_0 - q + \Delta p) \cdot A \cdot (p_0 - q + \Delta p)^{\mathrm{T}} \;=$$

$$= d^2_A(p_0, q) + 2 \cdot (p_0 - q) \cdot A \cdot \Delta p^T + \Delta p \cdot A \cdot \Delta p^T\,.$$

Note that the last term of the sum, $\Delta p \cdot A \cdot \Delta p^T$, is greater or equal to zero since A is positive definite. In order to prove that the overall sum is greater or equal to $d^2_A(p_0, q)$, it suffices to show that the second term of the sum vanishes:

$$2(p_0 - q) \cdot A \cdot \Delta p^T \;=\; 2 \cdot \frac{(p_j - q_j)}{(A^{-1})_{jj}} \cdot e_j \cdot A^{-1} \cdot A \cdot \Delta p^T$$

$$= \; 2 \cdot \frac{(p_j - q_j)}{(A^{-1})_{jj}} \cdot \Delta p_j \;=\; 0 \quad \text{since}$$

$$\Delta p_j \;=\; (p - p_0)_j \;=\; p_j - q_j - \frac{(p_j - q_j)}{(A^{-1})_{jj}} \cdot (A^{-1})_{jj} \;=\; 0$$

From (i) and (ii), we obtain the overall proposition $d^2_{\mathrm{MBB}(A)}(p, q) \le d^2_A(p, q)$. ◊