Appears in: Veltkamp R., Burkhardt H., Kriegel H.-P. (eds.): *State-of-the-Art in Content-Based Image and Video Retrieval*. Kluwer publishers, 2001.

Chapter 14

ADAPTABLE SIMILARITY SEARCH IN LARGE IMAGE DATABASES

Thomas Seidl, Hans-Peter Kriegel
University of Munich, Institute for Computer Science
Oettingenstr. 67, 80538 Munich, Germany
{seidl, kriegel}@dbs.informatik.uni-muenchen.de

Abstract Similarity has highly application dependent and even subjective characteristics. Similarity models therefore have to be adaptable to application specific requirements and individual user preferences. We focus on two aspects of adaptable similarity search: (1) *Adaptable Similarity Models*. Examples include pixel-based shape similarity as well as 2D and 3D shape histograms, applied to biomolecular and image databases. (2) *Efficient Similarity Query Processing*. Similarity models based on quadratic forms result in ellipsoid queries on high-dimensional data spaces. We present algorithms to efficiently process ellipsoid queries on index structures, and improve the performance by introducing various approximation techniques that guarantee no false dismissals for both similarity range queries and *k*-nearest neighbor queries.

Keywords: User-adaptable similarity search, quadratic forms, similarity matrix, ellipsoid query, efficient query processing, index structures

14.1 Introduction

A wide range of image databases has established its relevance in areas like medicine, journalism, fashion, art, and industry. For a long time, retrieving images has been restricted to queries by filename, captions, or keywords. Recently, a variety of concepts and systems support querying image databases by characteristics of the content such as color, texture, and shape. Most approaches, however, are based on predefined similarity models which neglect the fact that similarity often is highly subjective and may vary from user to user or, moreover, from query to query.

Two aspects need to be addressed in order to support user-adaptable similarity search. First, appropriate flexible similarity models have to be provided

that may be modified by the user. These similarity models include representations that reflect the relevant features of complex in multimedia, image, or spatial objects. By introducing 2D and 3D shape histograms, we give an example for illustrative feature vectors. Similarity is often measured by the Euclidean distance that particularly lacks flexibility and disregards local neighborhoods in histogram spaces, see for instance [9, 27]. In contrast, quadratic form distance functions are particularly well suited to be adapted to application requirements and individual user requirements.

Second, efficient query processing is a crucial task due to the large and still increasing size of image databases. For high-dimensional data including digital images or complex feature vectors, several techniques to reduce the dimensionality have been proposed. We extend these techniques that previously were used for the Euclidean distance only and apply them to quadratic forms. In addition, conservative approximations are used for similarity range queries as another approach to reduce the complexity of quadratic form-based similarity query processing. For *k*-nearest neighbor queries, the approximations are generalized to filter distance functions. When used in the filter step of a multistep similarity query processing architecture, all the methods yield an optimal filter selectivity and minimize the number of expensive exact evaluations.

This chapter is organized as follows: In Section 2, we introduce flexible models for user-adaptable similarity search in image and spatial databases. In Section 3, we present efficient algorithms for similarity query processing and give particular attention to high-dimensional data spaces and approximation techniques. The experiments in Section 4 demonstrate the good performance, and Section 5 concludes the paper.

14.2 Adaptable Similarity Models

In this section, we present some similarity models that are particularly adaptable to varying application requirements or to individual user preferences. The basic idea is to provide parameters for similarity distance functions that are specified by the user to the system at query time. A class of similarity distance functions that particularly well meets these requirements are quadratic forms which are well known from their application to color histograms, e.g. in IBM's QBIC project [11, 16, 21, 22]. They were already successfully used for a variety of similarity models [3, 4, 19, 23].

14.2.1 Pixel-based Shape Similarity

Image databases used for marketing purposes or in patent agencies may contain trade marks, clip arts, or pictograms that contain single objects or small groups of objects. Since errors by scanning, sampling, or segmentation may in-

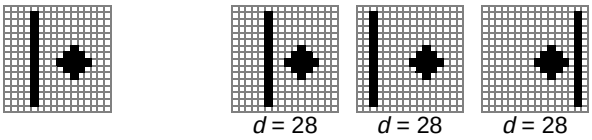


Figure 4.1. Given a reference image containing a vertical bar and a bullet (left), the Euclidean distance *d* does not distinguish between minor or major changes of the bar location.

duce slight displacements including minor shifts or rotations, similarity search faces new challenges which are not conquered by classic methods such as the Euclidean distance of digital images. An example is provided in Figure 4.1 where the bar in the image is shifted horizontally by a +1, –2, or +10 pixels to the right. The Euclidean distance to the original image cannot distinguish between slight and far translations.

We approach the problem of slightly displaced objects by taking the neighborhood of each pixel into account when computing the distance pixel by pixel. Similar images are recognized even if parts of the images are shifted by some pixels within the considered neighborhood. By specifying appropriate *neighborhood influence weights w*: dom(w) $\rightarrow$ [0, 1], the similarity model is adapted to varying application requirements or individual user preferences (Fig. 4.2).

DEFINITION (ADAPTABLE IMAGE SIMILARITY DISTANCE).

Given neighborhood influence weights *w*, the *local similarity distance* d'_w of two images F and G at a pixel p is defined as:

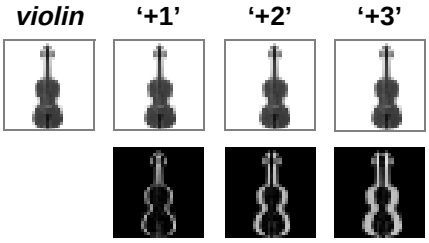
$$d'_w(F, G, p) \;=\; \sum_{\text{pixel } p'} w(p - p') \cdot (F(p') - G(p'))$$

and the *adaptable image similarity distance* d_w of F and G is defined as:

$$d_w(F, G) \;=\; \left(\sum_{\text{pixel } p} (F(p) - G(p)) \cdot d'_w(F, G, p)\right)^{1/2}$$

A straightforward algebraic transformation of the formula reveals that $d_w(F, G) = \sqrt{(F - G) \cdot W \cdot (F - G)^T}$ is a quadratic form, thereby considering the images F and G as highdimensional vectors. The neighborhood influence weights

Figure 4.2. Neighborhood weights for the adaptable image similarity model.



Neighborhood influence area	Rank of *violin* for query '+1'	'+2'	'+3'
None	**1**	**9**	**328**
'5-1'	**1**	**3**	**92**
'12-1'	**1**	**3**	**10**

Figure 4.3. Results of similarity ranking for the adaptable pixel-based model. The difference images in the bottom row illustrate the shifts of the original *violin* (left) to the right by +1, +2, or +3 pixels, respectively. The table indicates the position at which the original *violin* ranks for different neighborhood weights if querying with the shifted images +1, +2, or +3.

determine the matrix W by $W_{p,p'} = w(p - p')$ for each pair of pixels p and p'. The Euclidean distance function is generated by this schema simply by regarding no neighborhood influences, i.e. $w(p - p') = 0$ if $p \neq p'$ and $w(0) = 1$. The matrix W then is the identity.

To illustrate the desired effect for image retrieval, we consider a sample database containing 10,000 images taken from a commercially available CD-ROM. Figure 4.3 illustrates the sample image *violin*. The aim is to rank the original *violin* from the database at a top position even for shifted *violins* in query images. Whereas for a slight shift by one pixel to the right, the Euclidean distance (no neighborhood influence area) ranks the original *violin* at top, it fails in ranking the original *violin* at position 328 for a query violin shifted three pixels to the right. A neighborhood influence area of '12-1' (12 pixels to the right), ranks the *violin* among the top ten most similar images in the database.

14.2.2 2D and 3D Shape Histograms

Applications in molecular biology, mechanical engineering, or medical imaging are faced with objects that are larger and more complex than small pixel images. For searching similar spatial objects in large medical, biomolecular, or CAD databases, new models are required to represent and query the contours of 2D or 3D shapes [9, 19]. A successful approach is the concept of shape histograms. Having partitioned the 2D or 3D space into disjoint cells, the occupancy of each cell by the object of interest is measured. Figure 4.4 provides examples for 3D shape histograms, based on a partitioning of the 3D space into shells (a), into sectors (b), or into a combination of shells and sectors (c). Provided with the original representation of a molecule by a set of surface points, the fraction of points in each cell is measured and stored in the respective histogram bin. Figure 4.4d illustrates the application to X-ray images by partitioning the 2D space into a cell structure of shells and sectors. Rather than the distribution of surface points, the distribution of gray values over the cells is encoded by the histogram in this case.

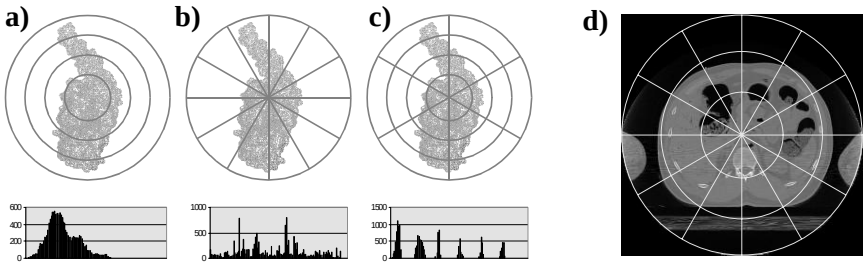


Figure 4.4. Shape histograms based on a shell model (a), a sector model (b) and a combined model (c) for a 3D molecule (*top row:* schematical illustrations, *bottom row:* actual histograms for 120 shells, 122 sectors, and 6x20 cells, resp.). d) A histogram model for 2D X-ray images.

For similarity search, histograms are simply considered to be high-dimensional feature vectors encoding relevant properties of complex objects, the geometric shape in our case. What needs to be complemented is the definition of a similarity distance measuring the dissimilarity of two objects by their distance values. Figure 4.5 demonstrates the limitations of the Euclidean distance when applied to shape histograms. Similar distributions in adjacent cells are not recognized since the Euclidean distance neglects any correlations between the dimensions. In order to take these cross-relationships into account, we encode the relationship of every two cells i and j by entries a_{ij} in a similarity matrix A and use quadratic forms $d_A(p, q) = \sqrt{(p-q) \cdot A \cdot (p-q)^T}$ as appropriate distance functions.

We applied our 3D shape histogram models to nearest neighbor classification for a 3D protein database containing molecules from the Protein Data Bank [1] that are classified in the FSSP database [17]. Taking care that for every class, at least two molecules are available, we have got 3,422 proteins which in total were assigned to 281 classes containing some 2 to 185 molecules. In order to measure the classification accuracy as the ratio of correctly predicted classes to the overall number of predictions, we performed *leave-one-out* experiments, thereby assigning the class label of the nearest neighbor to the query molecule.

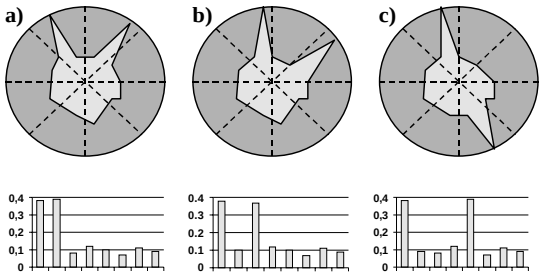


Figure 4.5. The Euclidean distance of 2D shape histograms does not reflect local neighborhoods of spatially adjacent bins. Objects a) and c) may count for being more similar than a) and b).

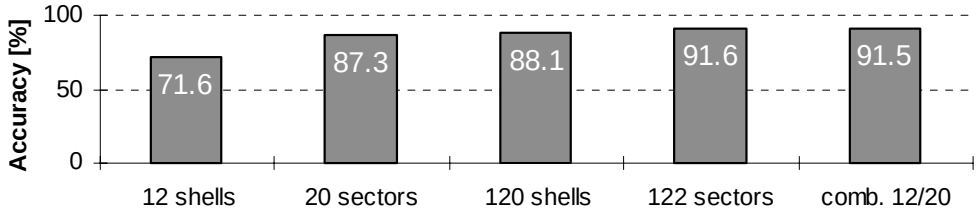


Figure 4.6. Classification accuracy of various histogram models on a database of 3,422 proteins.

The entries in the similarity matrix A are computed by the formula $a_{ij} = \exp(-\sigma \cdot d(i, j))$ with $\sigma = 10$ in our experiments. As distance $d(i, j)$ of two cells i and j, we use the difference of the corresponding shell radii or the angle between sectors, respectively. Figure 4.6 illustrates the results for various shape histogram models. As analyzed in [3] in more detail, the sector model yields a better classification accuracy than our shell model since there is more variation in the occupancy of the cells in space. The observed accuracy of about 90% is comparable to competing biomolecular approaches based on expert knowledge and manual verification. The advantage of our system is the high efficiency in addition to the high accuracy, and single queries are processed in less than a second.

14.3 Efficient Similarity Query Processing

Focusing to quadratic form distance functions as particularly adaptable similarity models, the problem of efficient query processing emerges. The evaluation of a quadratic form, $d_A(p, q) = \sqrt{(p-q) \cdot A \cdot (p-q)^T}$, of dimension n requires $O(n^2)$ floating point operations. For databases containing hundreds of thousands to millions of objects, sequentially scanning the database is prohibitive. Two ways help to improve the efficiency of similarity search in large databases. First, the use of multidimensional index structures and, second, the introduction of conservative approximation techniques.

14.3.1 Ellipsoid Queries on Indexes

Due to the geometric shape of its iso-surfaces, quadratic form-based queries are called *Ellipsoid Queries* [23]. These ellipsoids may have an arbitrary orientation in space. For weighted Euclidean distances which are represented by diagonal matrices, the ellipsoids are iso-oriented, i.e. the principal axes coincide with the coordinate axes of the data space. The isosurface of the most basic case, the Euclidean distance, represented by the identity matrix, is a sphere. For the Euclidean distance, several solutions employing multidimensional index structures are available [14]. These techniques are easily extended to weighted Euclidean distance functions simply by weighing the individual dimensions in-

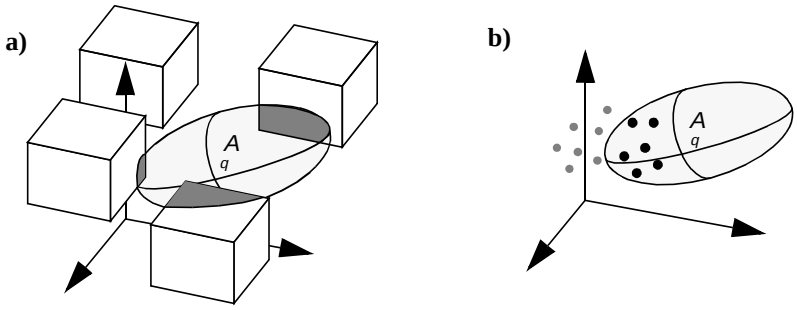


Figure 4.7. Intersection test for a query ellipsoid and some bounding boxes from a directory node of the index (*a*) and containment test for some data points from a leaf node of the index (*b*).

dependent of each other. For the general case of arbitrary positive definite similarity matrices, new solutions are required.

In IBM's QBIC project, a technique to efficiently support quadratic form-based distance functions was developed [11, 16, 22]. By an affine transformation of the data space obtained from a diagonalization of the similarity matrix, $A = VWV^T$ with W being a diagonal matrix and V being orthogonal, the original quadratic form translates to a weighted Euclidean distance in the transformed space, $d_A(p, q) = \sqrt{(pV - qV) \cdot W \cdot (pV - qV)^T} = d_W(pV, qV)$. The rich variety of multidimensional index structures may then be used for efficient query processing. The major limitation of this technique is the lack of flexibility and adaptivity to varying similarity matrices. The matrix has to be available at index creation time when the transformation of the data takes place. Later modifications of the similarity matrix, e.g. at query time, are not supported.

A quite more flexible approach is provided by the concept of ellipsoid queries in index structures [23, 24]. Rather than relying on a single similarity matrix and transforming the data depending on it, the matrix becomes a query parameter. The user may now specify any arbitrary similarity matrix for each individual query. The index does not have to be rebuilt when changing or modifying the matrix, and it immediately supports query processing for any similarity matrices given by the user at query time.

Any arbitrary multidimensional index structure which hierarchically organizes the data by rectilinear hyperrectangles may be employed. In our implementation, we focused on R-trees [15], R^+-trees [26], R*-trees [5] and X-trees [6, 8]. The algorithm recursively descends from the root node down to the leaf nodes while pruning subtrees whose bounding box does not intersect with the query ellipsoid in case of a similarity range query. Figure 4.7 illustrates the intersection test of a query ellipsoid with the bounding boxes of some subtrees and the containment test for some points from leaf nodes of the index. For *k*-nearest neighbor queries, the minimum distances of the query point and the bounding boxes are used as the best available heuristics to guide a best-first search without loosing completeness.

14.3.2 Approximations and Multistep Query Processing

Though the basic algorithm for ellipsoid queries on multidimensional index structures has shown its high efficiency in many cases, it performs very poor in high- and ultra-highdimensional data spaces. An effect known as the *curse of dimensionality* is that the performance of index structures generally degrades with increasing dimension [7, 8, 13]. A solution to overcome this severe problem is to employ approximation methods that reduce the complexity either of the data space or of the query. The complexity of a highdimensional data space is decreased by reducing the dimensionality. Addressing the complexity of ellipsoid queries, approximations for which a single evaluation takes $O(n)$ time in n dimensions help to avoid expensive $O(n^2)$ quadratic form evaluations.

In general, approximations introduce a trade-off between accuracy and efficiency, and completeness as well as soundness are important criteria. We address these aspects by following the paradigm of multistep query processing. A filter step that is supported by a multidimensional index structure is based on an appropriate approximation and produces a set of candidate answers. In a subsequent refinement step, the candidate answers are evaluated according to the exact complex query criterion. Whereas the soundness of the result is guaranteed by the exact evaluation in the refinement step, additional assumptions have to be fulfilled to ensure completeness as well. The illustration in Figure 4.8 sketches our multistep architecture for similarity query processing.

For similarity range queries, *conservative approximations* which completely enclose the original object guarantee no false dismissals in the result set. The approximation error in the filter step is thus constrained to false answers due to some surplus approximation space, but no deficit space may cause the loss of any true result. For k-nearest neighbor search, the geometric approximations have to be extended to distance functions since there is no query geometry that has a finite extension. Available algorithms guarantee completeness if the distance function in the filter step fulfills the *lower-bounding* property. For any two objects p and q, a lower-bounding distance function d_{lb} in the filter step has to return a value that is not greater than the exact distance d_{e} of p and q, i.e. $d_{\mathrm{lb}}(p, q) \leq d_{\mathrm{e}}(p, q)$.

The complete algorithm of [18] first retrieves the k-nearest neighbors according to the filter distance d_{lb} from the index, determines the maximum object distance $d_{\mathrm{e}}^{\max}$ for the answers and, then, reports all objects p whose filter distance $d_{\mathrm{lb}}(p, q)$ to the query object q is less or equal $d_{\mathrm{e}}^{\max}$ by a range query. Our new solution [25] is optimal in the sense that it produces the minimal number of candidates in the filter step and, therefore, the number of exact evaluations in the refinement step is minimal. The key to this optimization is that the filter step reports new candidates in ascending d_{lb}-order, and termination is controlled by a feedback from the refinement step.

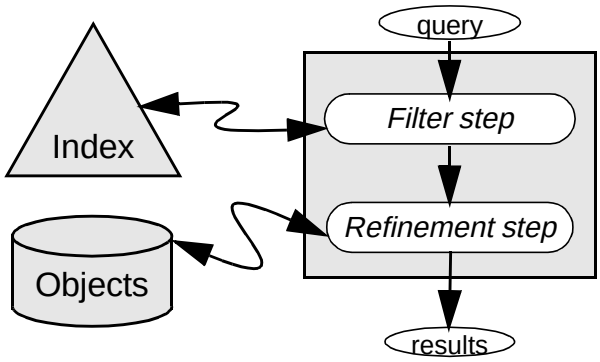


Figure 4.8. Multi-step similarity query processing.

14.3.3 Ellipsoid Queries in High Dimensions

In high-dimensional data spaces, the mentioned curse of dimensionality causes a serious degradation of multidimensional index structures. In addition, the $O(n^2)$ evaluation time for a quadratic form of dimension n yields unacceptable runtimes for high dimensional ellipsoid queries. Reduction of dimensionality helps to overcome these problems. The objects are projected from their high-dimensional data space to a space having less dimensions and for which index structures and ellipsoid distance evaluation show a significantly better efficiency. A variety of reduction techniques has been developed including Karhunen-Loève Transform (KLT), Discrete Fourier or Cosine Transform (DFT, DCT), Wavelet Transforms, Multidimensional Scaling [20], or FastMap [12]. Any linear reduction from n down to r dimensions is represented by a $n \times r$-dimensional reduction matrix R. Nevertheless, the reduction may be evaluated more efficiently e.g. by FFT algorithms. Conceptually, an n-dimensional point p is first rotated by a (complemented) $n \times n$-matrix R' and, afterwards, the last $(n - r)$ dimensions are cut from pR', resulting in an r-dimensional point pR. The complemented $n \times n$-matrix R' is obtained by extending the $n \times r$-matrix R with $(n - r)$ base vectors of the nullspace of R.

For quadratic forms, the question emerges which filter distance functions guarantee completeness by fulfilling the lower-bounding property and, in addition, which minimize the number of candidates to be evaluated in the refinement step. From a geometric point of view, the solution is a projection of the high-dimensional ellipsoid to the low-dimensional data space. This projection is obtained from a transformation of the n-dimensional similarity matrix A to an r-dimensional similarity matrix A'. Obviously, the projection has to respect the reduction matrix R in order to produce correct results.

The algorithm proceeds in two steps [23, 24]. First, the $n \times n$-matrix A is transformed by a multiplication with the complemented $n \times n$-matrix R', resulting in $R'^T A R'$. Second, performing a step by step reduction from n dimen-

sions down to r dimensions, the scaled outer product of the last column and the last row of the matrix is added to the remaining part of the matrix in each step.

As desired, the resulting quadratic form $d_{A'}(p, q) = \sqrt{(p-q) \cdot A' \cdot (p-q)^T}$ is a lower bound of the original quadratic form $d_A(p, q)$. On top of it, $d_{A'}(p, q)$ is the greatest of all lower-bounding distance functions in the reduced space and, therefore, is optimal since no other complete filter distance function in the reduced space produces less candidates.

14.3.4 Geometric Approximations of Ellipsoids

A variety of conservative approximation techniques has been developed for 2D spatial database systems and Geographic Information Systems [10]. From these methods, we found the *Minimum Bounding Box* (MBB) and the *Minimum Bounding Sphere* (MBS) to be best suited for ellipsoid queries [2, 23]. Both approximations, the MBB and the MBS, require only $O(n)$ time for testing intersections and containments. In addition, combining the methods exploits the advantages of both.

All approximations are applied to both query types, similarity range queries and k-nearest neighbor queries. For this purpose, we have to provide two instances for each model: First, a conservative approximate query region which completely encloses the original query ellipsoid. Whereas the orientation and the location of the original query ellipsoid is specified by the similarity matrix A and the center point q, the extension is derived from the range query parameter ε. For convenient reference of ellipsoid range queries, let us introduce the following symbol:

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

Second, an approximate distance function is required for each of the approximation types. Due to their close relationship to the corresponding solids, we call the respective distance functions box distance function and sphere distance function. In order to guarantee the completeness of the filter step as well as the minimality of the produced candidate sets, our particular filter distance functions are designed to be the greatest lower bounds within their class of box or sphere distance functions.

14.3.4.1 Minimum Bounding Box Approximation

The *Minimum Bounding Box* (MBB) of a spatial object is the smallest rectilinear hyperrectangle totally enclosing the object. The MBB is a favorite approximation technique due to its compact representation which requires only $2 \cdot n$ parameters in an n-dimensional space since it suffices to store the lower and upper bound in each dimension. It is highly compatible to rectilinearly or-

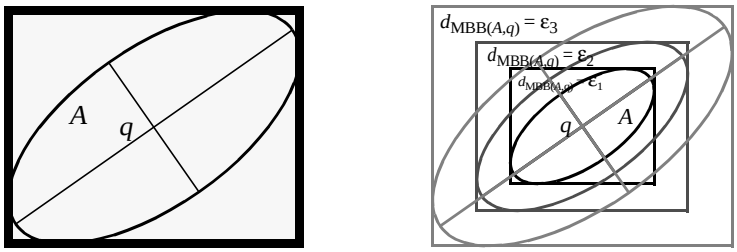


Figure 4.9. Minimum Bounding Box MBB(A,q,ε) of a 2D ellipsoid of level ε and greatest lower-bounding box distance function $d_{\mathrm{MBB}(A,q)}$.

ganized multidimensional access methods and, typically, easy to determine. Figure 4.9 provides an example for the MBB of a 2D ellipsoid. Since the geometry of the MBB can be derived from the corresponding lower bounding distance function, we first introduce the more general MBB distance function.

Lower-Bounding Box Distance Function. The generalization of a box to a distance function yields a weighted maximum norm L_∞ whose isosurfaces are rectilinear hyperrectangles. The weighting factors for individual dimensions represent non-square rectangles. What we get in analogy to the minimum bounding box of an ellipsoid is the greatest box distance function providing a lower bound of the corresponding ellipsoid distance function.

DEFINITION AND THEOREM (MBB DISTANCE FUNCTION).

Let A be a similarity matrix, and A^{-1} its inverse. The weighted maximum norm distance function

$$d_{\mathrm{MBB}(A)}(p, q) \;=\; \max \{| p_i - q_i | / \sqrt{A^{-1}_{ii}} \;: i = 1, \ldots, d\}$$

is called the *minimum bounding box distance function* of A. It is a lower bound of the ellipsoid distance function d_A:

$$d_{\mathrm{MBB}(A)}(p, q) \;\leq\; d_A(p, q) \quad \text{for all } p, q \in \Re^d$$

PROOF.

First, $d_{\mathrm{MBB}(A)}$ is well-defined since A^{-1} exists for every positive definite matrix A, and all diagonal elements, A^{-1}_{ii}, are positive. Second, we show that for every $p, q \in \Re^d$, an auxiliary point p_t exists such that the following formula is true which immediately implies the proposition:

$$d_{\mathrm{MBB}(A)}(p, q) \;=\; d_A(p_t, q) \;\leq\; d_A(p, q)$$

Consider the box on which p is located and the largest ellipsoid enclosed in this box. The tangential point p_t of the box and the ellipsoid shares its box dis-

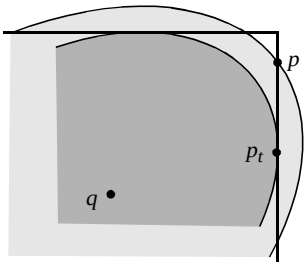


Figure 4.10. The tangential point p_t shares its box distance with p but is located on a smaller ellipsoid than p. At p_t, the box and ellipsoid distances are equal: $d_A(p_t, q) = d_{\mathrm{MBB}(A)}(p_t, q)$.

tance with p, $d_{\mathrm{MBB}(A)}(p_t, q) = d_{\mathrm{MBB}(A)}(p, q)$, whereas the ellipsoid of p_t is smaller than the ellipsoid on which p is located, i.e. $d_A(p_t, q) \leq d_A(p, q)$. As shown in [2], the box distance and the ellipsoid distance of p_t to q are equal, $d_{\mathrm{MBB}(A)}(p_t, q) = d_A(p_t, q)$, and the proposition holds. ◊

The equality of the distance values at p_t, i.e. $d_{\mathrm{MBB}(A)}(p_t, q) = d_A(p_t, q)$, indicates that $d_{\mathrm{MBB}(A)}$ represents the greatest of all box-shaped lower-bounding distance functions. As a consequence, $d_{\mathrm{MBB}(A)}$ guarantees the best filtering quality that can be achieved for lower-bounding distance functions that are based on a weighted maximum norm.

Geometry of the Minimum Bounding Box. The minimum bounding box $\mathrm{MBB}(A, q, \varepsilon)$ for a given ellipsoid $ellip(A, q, \varepsilon)$ can be computed by determining the tangential hyperplanes whose normal vectors are parallel to the coordinate axes. A simpler way is to derive the geometry from the MBB distance function $d_{\mathrm{MBB}(A)}$:

$$\mathrm{MBB}(A, q, \varepsilon) = \{p \in \Re^d : d_{\mathrm{MBB}(A)}(p, q) \leq \varepsilon\} =$$

$$= \{p \in \Re^d : \max\{|p_i - q_i| / \sqrt{A^{-1}}_{ii} : i = 1, \ldots, d\} \leq \varepsilon\} =$$

$$= \{p \in \Re^d : \forall i = 1, \ldots, d: |p_i - q_i| \leq \varepsilon \cdot \sqrt{A^{-1}}_{ii}\}$$

Thus, the i-th dimension of $\mathrm{MBB}(A, q, \varepsilon)$ covers the following range:

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

14.3.4.2 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 $n + 1$ parameters in n-dimensional spaces to store the radius and the n coordinates of the center

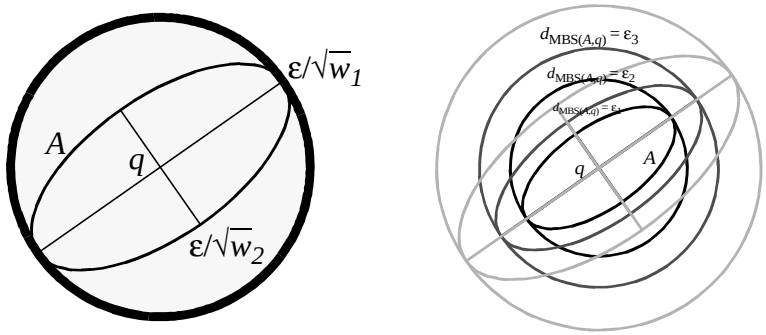


Figure 4.11. Minimum Bounding Sphere MBS(A,q,ε) of an ellipsoid *ellip*(A, q, ε) and greatest lower-bounding sphere distance function $d_{\mathrm{MBS}(A)}$. The radius of the MBS depends on the smallest eigenvalue $w_{\min}$ of A and on the level ε.

point. For ellipsoids, the center of the MBS coincides with the center of the ellipsoid. Figure 4.11 provides an example in 2D.

Lower-Bounding Sphere Distance Function. Also for the MBS approximation model, we use a distance function $d_{\mathrm{MBS}(A)}$ that lower-bounds the ellipsoid distance function d_A. The generalization of spheres to distance functions yields a scaled Euclidean distance with a scaling factor corresponding to the radius of the sphere.

DEFINITION AND THEOREM (MBS DISTANCE FUNCTION).

Let A be a similarity matrix, and $w_{\min}$ the minimum eigenvalue of A. The scaled Euclidean distance function $d_{\mathrm{MBS}(A)} = \sqrt{w_{\min}} \cdot |p - q|$ is called the *minimum bounding sphere distance function* of A. It is a lower bound of d_A:

$$d_{\mathrm{MBS}(A)}(p, q) \;\leq\; d_A(p, q) \quad \text{for all } p, q \in \Re^d$$

PROOF.

Since the matrix A is positive definite, the diagonalization $A = V\,W\,V^T$, $V\,V^T = Id$, $W = \mathrm{diag}(w_1, \ldots, w_d)$ exists, and the eigenvalues of A, $w_1, \ldots, w_d$, are positive. If denoting the minimum of these eigenvalues by $w_{\min}$, we obtain:

$$d_A(p, q) = \sqrt{(p-q) \cdot VWV^T \cdot (p-q)^T} \;=\; \sqrt{(pV-qV) \cdot W \cdot (pV-qV)^T} \;=$$

$$= \sqrt{\sum_{i=1}^{d} w_i \cdot (pV-qV)_i^2} \;\geq\; \sqrt{\sum_{i=1}^{d} w_{\min} \cdot (pV-qV)_i^2} \;=$$

$$= \sqrt{w_{\min} \cdot (p-q) \cdot VV^T \cdot (p-q)^T} \;=\; \sqrt{w_{\min}} \cdot |p-q| \;=\; d_{\mathrm{MBS}(A)}(p, q). \;\Diamond$$

Note that $d_{\mathrm{MBS}(A)}(p_t, q)$ is equal to $d_A(p_t, q)$ for the tangential point p_t and, therefore, $d_{\mathrm{MBS}(A)}$ represents the greatest lower-bounding distance function of the spherical type. This optimality ensures the best approximation quality that can be achieved for the class of scaled Euclidean distance functions.

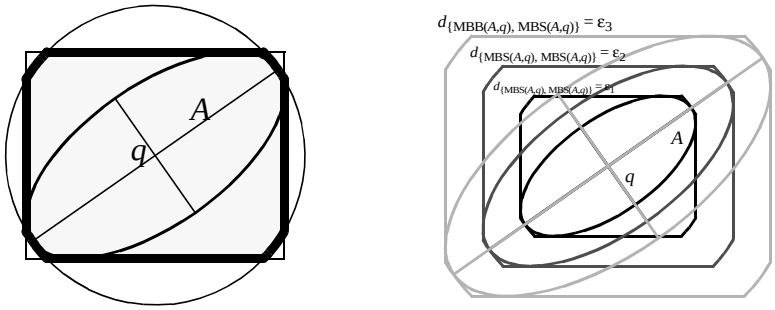


Figure 4.12. Combined approximation $\mathrm{MBB}(A,q,\varepsilon) \cap \mathrm{MBS}(A,q,\varepsilon)$ of an ellipsoid of level ε and lower-bounding combined distance function $d_C = \max\{d_{\mathrm{MBB}(A,q)}, d_{\mathrm{MBS}(A,q)}\}$.

Geometry of the Minimum Bounding Sphere. Given an arbitrary ellipsoid $ellip(A, q, \varepsilon)$, the center of the MBS obviously coincides with the center q of the ellipsoid. The radius $r = \varepsilon / \sqrt{w_{\min}}$ of $\mathrm{MBS}(A, q, \varepsilon)$ is derived from the MBS distance function $d_{\mathrm{MBS}(A)}$ as follows:

$$\mathrm{MBS}(A, q, \varepsilon) = \{p: d_{\mathrm{MBS}(A)}(p, q) \le \varepsilon\} = \{p \in \Re^d: \ |p - q| \le \varepsilon / \sqrt{w_{\min}}\}$$

Since $d_{\mathrm{MBS}(A)}(p_t, q) = d_A(p_t, q)$ for the tangential point p_t, we have got the actual minimum of all bounding spheres of $ellip(A, q, \varepsilon)$.

14.3.4.3 Combination of 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, we demonstrate how to combine conservative approximations for similarity range queries and how to combine basic lower-bounding distance functions for k-nearest neighbor search.

Combination of Approximations. Let $\mathrm{APP}_1(A, q, \varepsilon)$ and $\mathrm{APP}_2(A, q, \varepsilon)$ be two conservative approximations of $ellip(A, q, \varepsilon)$, i.e. $ellip(A, q, \varepsilon) \subseteq \mathrm{APP}_1(A, q, \varepsilon)$ and $ellip(A, q, \varepsilon) \subseteq \mathrm{APP}_2(A, q, \varepsilon)$. Then the intersection of both approximations is a conservative approximation of $ellip(A, q, \varepsilon)$, too:

$$ellip(A, q, \varepsilon) \ \subseteq \ \mathrm{APP}_1(A, q, \varepsilon) \cap \mathrm{APP}_2(A, q, \varepsilon).$$

Figure 4.12 shows a 2D example for a conservative approximation that combines the minimum bounding box (MBB) and the minimum bounding sphere (MBS) 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.

Combination of Lower-Bounding Distance Functions. Analogous to the preceding approximation techniques, a combination of lower-bounding distance functions that again lower-bounds the exact similarity distance function is desired. The following derivation shows that the maximum of the component distance functions fulfills this requirement.

DEFINITION AND THEOREM (COMBINED DISTANCE FUNCTION).

Let A be a similarity matrix and $C = \{d_i\}$ be a set of lower-bounding distance functions for d_A, i.e. $d_i(p, q) \leq d_A(p, q)$ for all $p, q \in \Re^d$. Then, the combined distance function $d_C = \max\{d_i\}$ is a lower bound of d_A, too:

$$d_C(p, q) \leq d_A(p, q) \quad \text{for all } p, q \in \Re^d$$

PROOF.

For all $p, q \in \Re^d$, the following equivalences hold: $d_C(p, q) \leq d_A(p, q) \Leftrightarrow \max\{d_i(p, q)\} \leq d_A(p, q) \Leftrightarrow \forall d_i: d_i(p, q) \leq d_A(p, q)$. The final inequality represents the precondition, and the proposition is true. ◊

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

14.4 Performance Evaluation

We evaluated our algorithms on various test databases containing 10,000 grayscale images, 112,000 color images and 1,000,000 synthetic vectors for different similarity models.

14.4.1 Varying Ellipsoids in High Dimensions

The first experiments ran on a database of 10,000 grayscale clip arts of resolution $32 \times 32 = 1024$ pixels each. The 1024-dimensional image vectors were reduced to r dimensions for $r \in \{16, 32, 48, 64\}$, and the r-vectors are managed by X-trees [8] on an HP 9000/780 machine. We performed a sample of k-nearest neighbor queries for different neighborhood influence areas '1-1', '3-1', '6-1', and '9-1' and measured the number of I/O operations as well as the overall response time.

We observed that with increasing dimension of the index, the number of candidates significantly decreases since the index provides more information to the filter step. On the other hand, the number of accessed index pages increases due to the higher space requirement and the curse of dimensionality. An optimal dimension for the index yielding a minimum overall runtime therefore

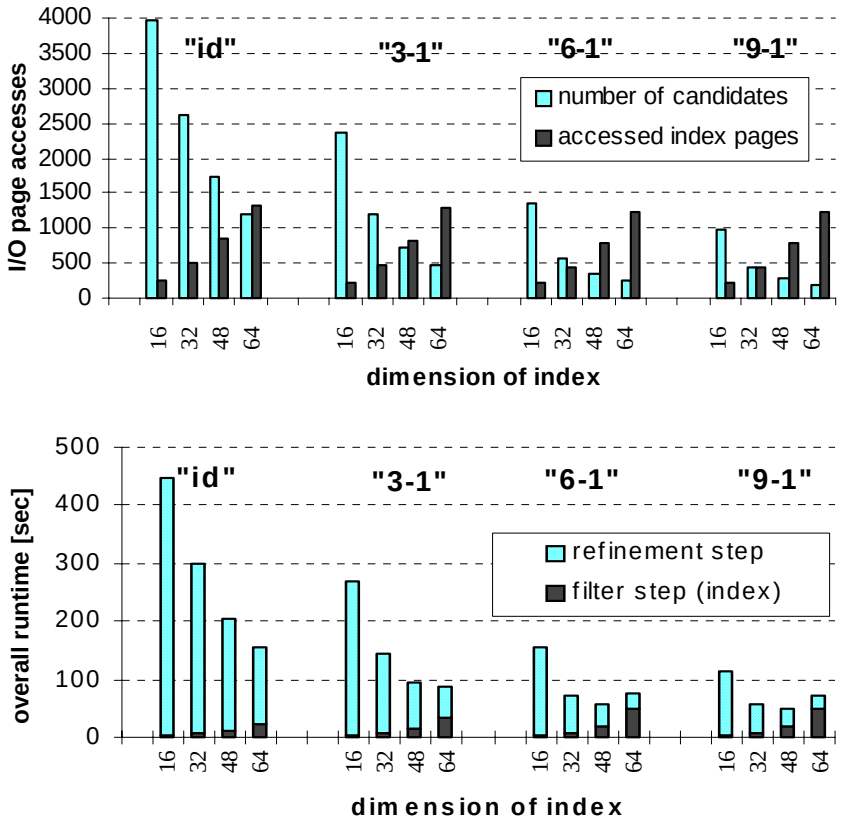


Figure 4.13. For various neighborhood influence areas (‘id’, ‘3-1’, ‘6-1’, ‘9-1’), the number of candidates and accessed index pages (top diagram) and the overall runtime (bottom diagram) for index dimensions 16, 32, 48, 64 is depicted for a sample of 10-nn queries (selectivity 0.1%).

exists due to this trade-off. The diagram in Figure 4.13 shows that the optimum dimension for the index depends on the neighborhood influence area. In particular, for ‘9-1’ and ‘6-1’, the optimum is approximately 48 whereas for ‘3-1’ and ‘1-1’ (Id), the optimum is greater than 64.

14.4.2 Approximations of Ellipsoids

We applied the approximation techniques to a large image database containing 8D color histograms of 112,000 images obtained from a reduction of dimensionality as well as to a database of 1,000,000 objects that are uniformly distributed in the 8D. The experiments were performed on an HP735 under HP-UX 10.20. In the diagrams, we use the symbols BOX for the box approximation, SPHERE for the sphere approximation, and COMB for the combination of BOX and SPHERE. The symbol NONE indicates the pure exact ellipsoid evaluation without using any approximation.

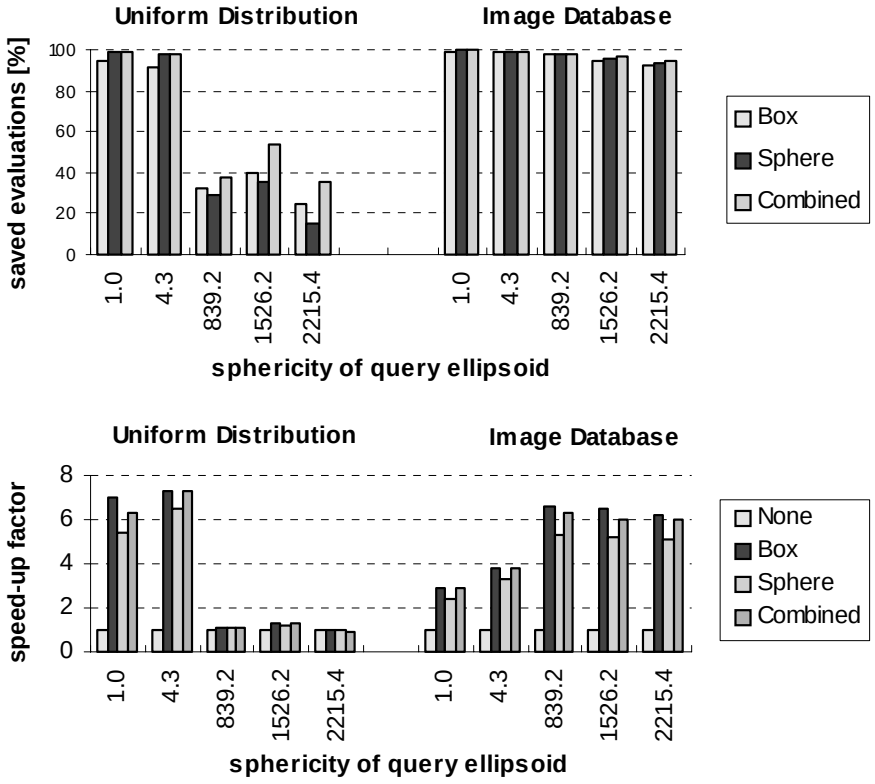


Figure 4.14. Percentage of saved evaluations for intersection and containment tests (range queries) with different matrices (top diagram). The CPU speed-up factors for different ellipsoid sphericities range from .95 to 7.3 (bottom diagram).

Speed-Up for Different Similarity Matrices. For our next experiments, we use a sample of different similarity matrices whose components a_{ij} are computed by the formula $a_{ij} = \exp(-\sigma \cdot (d_w(c_i, c_j) / d_{\max})^2)$ adapted from [16] where σ is a positive constant and $w = (w_r, w_g, w_b)$ are relative weights of red, green, blue in the RGB color space as specified by the user. We distinguish the matrices by their sphericity, i.e. the ratio of the MBS volume to the volume of the ellipsoid. The ellipsoids in the experiments have a sphericity of 1.035 up to 2,200. On both databases, the uniformly distributed data as well as the image database, we performed range queries returning between 1 and 10 results on the average.

The top diagram in Figure 4.14 indicates the percentage of exact ellipsoid evaluations that were saved due to the approximations. In case of uniformly distributed data, 90% of ellipsoid evaluations are avoided for almost spherical ellipsoids. For less spherical ellipsoids, still 20% to 60% of the exact ellipsoid evaluations are avoided. Obviously, the combined approximations yield the most savings. For the image database, more than 90% of the ellipsoid evaluations are avoided in all of our experiments.

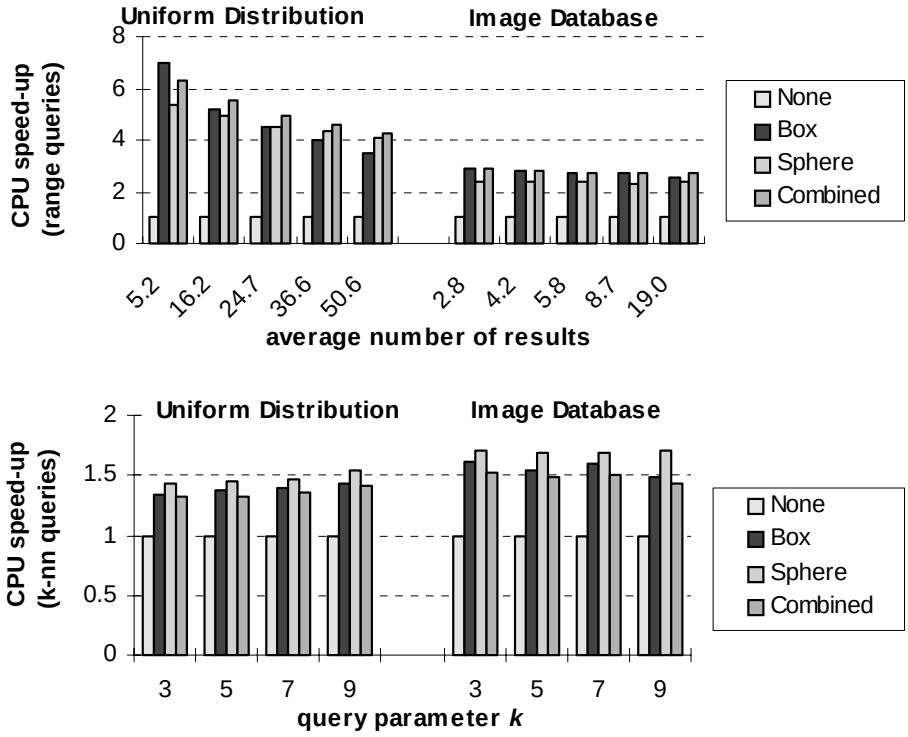


Figure 4.15. The speed-up factors for range queries range from 2.3 to 7 (top diagram). The speed-up factors for *k*-nearest neighbor queries range from 1.3 to 1.7 (bottom diagram).

In the bottom diagram of Figure 4.14, the impact of avoiding exact ellipsoid evaluations on the elapsed time is illustrated for the same sample of range queries as above. For the uniformly distributed data, we observed improvement factors up to 7.3. In our example, the speedup factor decreases with increasing sphericity of the ellipsoids and falls below 1.0 for the last similarity matrix. An optimizer should detect such a situation and prevent the system from using the approximation in this case. For the image database, the speed-up factors range from 2.8 to 6.7.

Speed-Up for Different Query Parameters. In our next series of experiments, we performed samples of range queries and *k*-nearest neighbor queries for various query ranges ε and query parameters *k*. The similarity matrix corresponds to an ellipsoid with sphericity 1.035. The top diagram in Figure 4.15 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 5.2 to 50.6 results from the uniformly distributed data and 2.8 to 19 results from the image database. In these experiments, the approximations outperform the pure ellipsoid evaluation by factors of 3.5 to 7.0 (uniform distribution) and of 2.3 to 2.9 (image database).

In the bottom diagram of Figure 4.15, we demonstrate the improvement that we achieved for *k*-nearest neighbor queries for a varying value of *k*. For the uniform distribution, we achieved a speed-up of 1.3 to 1.5, and for the image database a speed-up factor of 1.4 to 1.7.

14.5 Conclusions

Similarity has application-dependent and subjective user-dependent characteristics. Whereas the Euclidean distance of high-dimensional objects and feature vectors fails to recognize local similarities nor is adaptable to user requirements, quadratic forms demonstrate their support for flexible similarity search that may be adapted to individual preferences. Several similarity models based on quadratic forms have been developed and, in this paper, we presented a pixel-based model for icons and clip arts as well as a shape histogram approach for 3D spatial databases and 2D medical imaging.

We presented a multi-step architecture for similarity query processing that supports the adaptable similarity models represented by quadratic form distance functions. Multidimensional indexing methods as well as techniques to reduce the dimensionality have been extended to meet the characteristics of ellipsoid queries as a new query type in databases. Additionally, various conservative approximations have shown their successful applicability to ellipsoid queries. For all cases, completeness of the filter step is guaranteed for both, similarity range queries as well as *k*-nearest neighbor queries, and no results are missing in the answer sets. Experimental evaluations demonstrate the efficiency of the query processor even on very large databases.

References

1. Abola E. E., Sussman J. L., Prilusky J., Manning N. O.: *Protein Data Bank. Archives of Three-Dimensional Macromolecular Structures*. Methods in Enzymology 277, 1997, 556-571.
2. Ankerst M., Braunmüller B., Kriegel H.-P., T. Seidl T.: *Improving Adaptable Similarity Query Processing by Using Approximations*. Proc. 24th Int. Conf. on Very Large Data Bases (VLDB), 1998, 206-217.
3. Ankerst M., Kastenmüller G., Kriegel H.-P., Seidl T.: *3D Shape Histograms for Similarity Search and Classification in Spatial Databases*. Proc. 6th Int. Symp. on Large Spatial Databases (SSD), LNCS 1651, 1999, 207-226.
4. Ankerst M., Kriegel H.-P., Seidl T.: *A Multistep Approach for Shape Similarity Search in Image Databases*. IEEE Trans. on Knowledge and Data Engineering 10(6), 1998, 996-1004.

5. 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, 1990, 322-331.
6. 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, 1997, 1-12.
7. Berchtold S., Böhm C., Keim D., Kriegel H.-P.: *A Cost Model for Nearest Neighbor Search in High-Dimensional Data Spaces*. Proc. 16th ACM Symp. on Principles of Database Systems (PODS), 1997, 78-86.
8. 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), 1996, 28-39.
9. Berchtold S., Kriegel H.-P.: *S3: Similarity Search in CAD Database Systems*. Proc. ACM SIGMOD Int. Conf. on Management of Data, 1997, 564-567.
10. 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 (ICDE), 1993, 40-49.
11. 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, 231-262.
12. Faloutsos C., Lin K.-I.: *FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Data.* Proc. ACM SIGMOD Int. Conf. on Management of Data, 1995, 163-174.
13. Faloutsos C., Ranganathan M., Manolopoulos Y.: *Fast Subsequence Matching in Time-Series Databases*., Proc. ACM SIGMOD Int. Conf. on Management of Data, 1994, 419-429.
14. Gaede V., Günther O.: *Multidimensional Access Methods*. ACM Computing Surveys 30(2), 1998, 170-231.
15. Guttman A.: *R-trees: A Dynamic Index Structure for Spatial Searching*. Proc. ACM SIGMOD Int. Conf. on Management of Data, 1984, 47-57.
16. 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 17(7), 1995, 729-736.
17. Holm L., Sander C.: *Touring Protein Fold Space with Dali/FSSP.* Nucleic Acids Research 26, 1998, 316-319.
18. Korn F., Sidiropoulos N., Faloutsos C., Siegel E., Protopapas Z.: *Fast and Effective Retrieval of Medical Tumor Shapes*. IEEE Trans. on Knowledge and Data Engineering 10(6): 1998, 889-904.

19. Kriegel H.-P., Seidl T.: *Approximation-Based Similarity Search for 3-D Surface Segments*. GeoInformatica 2(2), 1998, 113-147.
20. Kruskal J. B., Wish M.: *Multidimensional Scaling*. SAGE publications, Beverly Hills, 1978.
21. Niblack W. et al.: *The QBIC Project: Querying Images by Content Using Color, Texture, and Shape*. SPIE Int. Symp. on Electronic Imaging: Science and Technology Conf. 1908, Storage and Retrieval for Image and Video Databases, 1993.
22. Sawhney H., Hafner J.: *Efficient Color Histogram Indexing*. Proc. Int. Conf. on Image Processing, 1994, 66-70.
23. Seidl T.: *Adaptable Similarity Search in 3-D Spatial Database Systems*. Ph.D. Thesis, University of Munich, 1997. Herbert Utz Publishers, Munich, 1998.
24. Seidl T., Kriegel H.-P.: *Efficient User-Adaptable Similarity Search in Large Multimedia Databases*. Proc. 23rd Int. Conf. on Very Large Data Bases (VLDB), 1997, 506-515.
25. Seidl T., Kriegel H.-P.: *Optimal Multi-Step k-Nearest Neighbor Search*. Proc. ACM SIGMOD Int. Conf. on Management of Data, 1998, 154-165.
26. Sellis T., Roussopoulos N., Faloutsos C.: *The R+-Tree: A Dynamic Index for Multi-Dimensional Objects*. Proc. 13th Int. Conf. on Very Large Data Bases (VLDB), 1987, 507-518.
27. White D. A., Jain R.: *Similarity Indexing with the SS-tree*. Proc. 12th Int. Conf. on Data Engineering (ICDE), 1996, 516-523.