Proc. Int'l Conf. on Data Warehousing and Knowledge Discovery (**DaWaK'01**).
Lecture Notes in Computer Science, © Springer Verlag. 2001

Adaptable Similarity Search Using Vector Quantization

Christian Böhm, Hans-Peter Kriegel, and Thomas Seidl

University of Munich, Oettingenstr. 67, D-80538 Munich, Germany
{boehm,kriegel,seidl}@informatik.uni-muenchen.de

Abstract. Adaptable similarity queries based on quadratic form distance functions are widely popular in data mining applications, particularly for domains such as multimedia, CAD, molecular biology or medical image databases. Recently it has been recognized that quantization of feature vectors can substantially improve query processing for Euclidean distance functions, as demonstrated by the scan-based VA-file and the index structure IQ-tree. In this paper, we address the problem that determining quadratic form distances between quantized vectors is difficult and computationally expensive. Our solution provides a variety of new approximation techniques for quantized vectors which are combined by an extended multistep query processing architecture. In our analysis section we show that the filter steps complement each other. Consequently, it is useful to apply our filters in combination. We show the superiority of our approach over other architectures and over competitive query processing methods. In our experimental evaluation, the sequential scan is outperformed by a factor of 2.3. Compared to the X-tree, on 64 dimensional color histogram data, we measured an improvement factor of 5.6.

1. Introduction

Similarity search in large databases has gained much attention during the last years. A successful approach is provided by feature-based similarity models where appropriate properties of the objects are mapped to vectors of a usually high-dimensional space. The similarity of objects is defined in terms of their distance in the feature space. In addition to the Euclidean distance function which is a very simple but inflexible similarity measure, quadratic forms often provide more appropriate similarity models. Given a positive definite so-called similarity matrix A, the distance of two vectors p and q is defined to be

$$\mathrm{dist}_A(p, q) = \sqrt{(p-q) \cdot A \cdot (p-q)^T}.$$

As the locus of all points p having a distance $\mathrm{dist}_A(p, q) \leq \varepsilon$ is an ellipsoid centered around q, the quadratic form-based queries are called *ellipsoid queries* [12].Quadratic form distance functions have been successfully employed for a variety of similarity models in different domains. Examples include the color histogram model for color images in IBM's Query By Image Content (QBIC) system [7, 8], histogram and non-histogram distances for images [13], the shape similarity model for 3D surface segments in a biomolecular database [10], a 2D shape similarity model for clip arts [3], a 3D shape histogram model for proteins [2], or a relevance feedback system [9].

It has been widely recognized that in many application domains, there is not simply one valid measure for the similarity of objects. Instead, the notion of similarity changes with the user's focus of search. This observation has led to the need of user-adaptable similarity models where the user may adapt the similarity distance function to changing application requirements or even personal preferences. As an example, the color histogram approach was extended to become user-adaptable [12]. The query system of [9] based on relevance feedback through multiple examples relies on iterated modifications of the query matrix thus approaching the 'hidden' distance function in the user's mind.

For an efficient query evaluation, feature vectors are often managed in a multidimensional index. Various index structures have been proposed for this purpose [11, 14, 6]. Due to a bunch of problems usually called the '*curse of dimensionality*', even specialized index structures deteriorate with increasing dimension of the feature space. It has been shown [15] that in many situations, depending on parameters such as the dimension, the data distribution and the number of objects in the database, indexes often fail to outperform simpler evaluation methods based on the sequen-

tial scan of the set of feature vectors. The solution proposed by Weber, Schek and Blott is therefore an improvement of the sequential scan by quantizing the feature vectors, called *VA-file* [15]. The general idea of the VA-file is to store the features not with the full precision of e.g. 32 bit floating point values but to use a reduced number of bits. For this purpose an irregular, quantile based grid is laid over the data space. Following the paradigm of multistep query processing, candidates produced by this filter step are exactly evaluated by a subsequent refinement step. The gain of the lossy data compression is a reduced time to load the feature vectors from secondary storage. Queries using Euclidean distance metric can be accelerated by factors up to 6.0 with this technique, if the search is I/O bound.

While the VA-file is an excellent choice for Euclidean and weighted Euclidean distance metric, several problems come up when the similarity measure is a quadratic form distance function. First, the scan based processing of ellipsoid queries is typically CPU bound, because the determination of the distance function has quadratic time complexity with respect to the dimension of the feature space. Therefore, a pure acceleration of the I/O time will hardly improve the answer time. A second problem is that it is generally difficult to determine the distance between the query point and the grid approximation of the feature vector if the distance measure is a general (i.e. not iso-oriented) ellipsoid. The reason is that the intersection point between the ellipsoid and the approximation cell can only be determined by expensive numerical methods.

Our solution to these problems is a series of three filter steps with increasing evaluation cost and decreasing number of produced candidates. These filter steps approximate both, the query ellipsoid and the grid cells at different levels of accuracy. Our first filter step performs a new kind of approximation of the query ellipsoid, the approximation by an axis-parallel ellipsoid. As it corresponds to a weighted euclidean distance calculation, this approximation can be evaluated very efficiently for vector approximations. Unfortunately the selectivity of this filter step is for some query ellipsoids not good enough as desired. The selectivity can be improved by large factors, however, if the first filter is followed by a novel technique approximating the grid cells. This technique determines the distance between the query point and the center of the grid cell, additionally considering a maximum approximation error. The maximum approximation error represents the maximum (ellipsoid) distance between the center point of the grid cell and its most distant corner point. This will be our last filter step. As the determination of the maximum approximation error has quadratic time complexity, we propose as an intermediate filter step another conservative approximation of the maximum approximation error which can be determined in linear time.

The benefits of our technique are not limited to the extension of the VA-file approach. Our filter steps can always be applied when ellipsoid distances to rectilinear rectangles have to be efficiently estimated. For instance, the page regions of most index structures are rectangular. Therefore, processing the directory pages of a multidimensional index structure can be improved by our technique. Recently, the idea of vector approximation of grid cells was also applied to index based query processing in the so-called IQ-tree [4]. Our multi-step query processing architecture enables efficient processing of ellipsoid queries in the IQ-tree. As one concrete example, however, we put our focus in this paper to processing of adaptable similarity queries by the VA-file. Out of the focus of this paper is dimensionality reduction [7, 12]. Reduction techniques based on the principal components analysis, fourier transformation or similar methods can be applied to our technique and all competitive techniques as a preprocessing step.

The rest of this paper is organized as follows: In section 2 (related work) we introduce various techniques for evaluating ellipsoid queries and the vector quantization. In section 3 we describe our novel techniques for approximating grid cells by minimum bounding ellipsoids in quadratic time, our linear approximation of the minimum bounding ellipsoid and our new approximation of the query ellipsoid by an iso-oriented ellipsoid. Section 4 is dedicated to the filter architecture. Here, we demonstrate in what situation what filter has particular strengths and how our three filters complement each other. Thus, it is useful to combine our three proposed filters, and each of the filters yields additional performance gains. The most efficient order of filter steps is also determined. Finally, we perform a comprehensive experimental evaluation in section 5.

2. Related Work

The evaluation of ellipsoid queries requires quadratic time in the data and directory nodes of the index. In order to reduce this effort, approximation techniques for ellipsoid queries have been developed that have a linear complexity thus supporting linear filter steps [1]. Conservative ap-

proximations of query ellipsoids such as the minimum bounding sphere, the minimum bounding box or the intersection of both guarantee no false dismissals for range queries. As a more general concept, equivalent lower-bounding distance functions were presented that guarantee no false dismissals for (*k*-)nearest neighbor queries in addition to similarity range queries.

Sphere Approximation. The *greatest lower-bounding sphere distance function* $d_{\text{glbs}(A)}$ of an ellipsoid is a scaled Euclidean distance function defined as follows. Let A be a similarity matrix, and $w_{\min}^2$ be the minimum eigenvalue of the matrix A, then

$$d_{\text{glbs}(A)}(p, q) = w_{\min} \cdot |p - q| .$$

In [1], it is shown that the function $d_{\text{glbs}(A)}$ is the greatest scaled Euclidean distance function that lower bounds the ellipsoid distance function d_A, i.e. for all $p, q \in \Re^d$:

$$d_{\text{glbs}(A)}(p, q) \leq d_A(p, q) .$$

Note particularly that $d_{\text{glbs}(A)}$ is well-defined since the eigenvalues of the positive definite similarity matrix are positive.

In addition to the minimum bounding sphere, also the minimum bounding box approximation has been generalized to an equivalent distance function:

Box Approximation. The *greatest lower-bounding box distance function* $d_{\text{glbb}(A)}$ of an ellipsoid is a weighted maximum distance function. For any similarity matrix A, the inverse A^{-1} exists, and we define:

$$d_{\text{glbb}(A)}(p, q) = \max\left\{ \frac{|p_i - q_i|}{\sqrt{(A^{-1})_{ii}}}, i = 1 \ldots d \right\}$$

The function $d_{\text{glbb}(A)}$ is the greatest weighted maximum distance function that lower bounds the ellipsoid distance function d_A, i.e. for all $p, q \in \Re^d$:

$$d_{\text{glbb}(A)}(p, q) \leq d_A(p, q)$$

Vector Quantization. The general idea of the VA-file [15] is to store the components of the feature vectors not with full (e.g. 32 bit) precision, but with a reduced precision of e.g. 5-6 bits. For this purpose, the data space is partitioned by an irregular grid which is determined according to the quantiles of the components of the feature vectors. Instead of the exact position of each point in the database, only the grid cell is stored in the VA-file. Scanning the reduced VA-file instead of the file containing the full precision data saves I/O time proportional to the compression rate. If a cell is partially intersected by the query region, it cannot be decided whether the point is in the result set or not. In this case, an expensive lookup to the file containing the full precision point data is due. Therefore, this approach corresponds to the paradigm of multistep query processing where the distance to the grid cell is a lower bound of the distance to the point. For Euclidean distance measures, the distance to a given cell can be determined very efficiently by precomputing the squared distances between the query point and the quantiles. Therefore, such a distance calculation can even be slightly cheaper than determining the distance between two points.

Recently, the idea of vector quantization was adopted to index based query processing. In the IQ tree [4] a separate regular grid is laid over each page region. Each data page has two representations: one containing the compressed (quantized) representation of the data points and the second containing the points in full precision. The optimal precision for the compressed data page as well as an optimal page schedule using a "fast index scan" operation are determined according to a cost model [5].

3. Ellipsoid Queries on Quantized Vectors

Compared to axis-parallel ellipsoids and spheres, it is much more difficult to determine whether a general ellipsoid intersects a rectangle. The reason for this difference is depicted in figure 1: For the axis-parallel ellipsoid, the closest point to the center of the ellipsoid can be easily determined by projecting the center point. In contrast, the general ellipsoid may in-

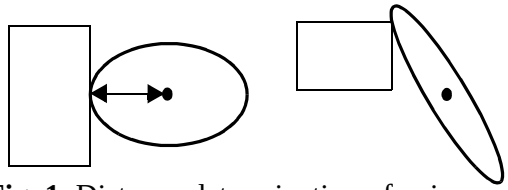

Fig. 1. Distance determination of axis-parallel and general ellipsoids to a rectangle

tersect the rectangle at its upper right corner although the center of the ellipsoid is underneath the lower boundary of the rectangle. This is impossible for unskewed ellipsoids and this property facilitates distance determination for Euclidean and weighted Euclidean metric.

The exact ellipsoid distance between a point and a rectangle can therefore only be determined by a time consuming numerical method [12], which is not suitable as a filter step. An approximation of the exact distance is needed which fulfills the lower bounding property. Our idea for this filter step is not to approximate the query (which is done in another filter step) but to approximate the rectangle. Unfortunately, the most common approximation by minimum bounding spheres suffers from the same problem as the rectangle itself: It is numerically difficult to determine the ellipsoid distance between the query point and the sphere. A suitable approximation, however, is an ellipsoid which has the same shape as the query ellipsoid, i.e. the same lengths and directions of the principal axes. We will show in the following that this approximation can be determined with low effort and that it can be used to define a lower bounding of the exact ellipsoid distance.

3.1 The Minimum Bounding Ellipsoid Approximation (MBE)

We want to determine the minimum ellipsoid enclosing a given rectangle R which is geometrically similar to the query ellipsoid. Due to the convexity of an ellipsoid, the rectangle is contained in the ellipsoid if all of its corners are contained. But we cannot investigate all 2^d corners of the rectangle to determine the minimum bounding ellipsoid. We need to determine the corner of the rectangle with maximum (ellipsoid) distance from the center of the rectangle. The following lemma will show that it is the corner which is closest to the eigenvector corresponding to the largest eigenvalue.

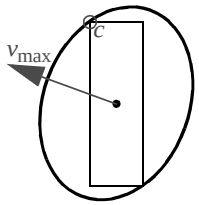

Fig. 2. Minimum bounding ellipsoid

Lemma 1. A rectangle is contained in an ellipsoid with the same center iff the corner which is closest to the eigenvector corresponding to the largest eigenvalue is contained in the ellipsoid.

Proof. Let $A = VW^2V^T$ with $VV^T = \mathrm{Id}$, $W^2 = \mathrm{diag}(w_i^2)$ be the diagonalization of A where V_i denotes the eigenvector corresponding to the eigenvalue w_i^2, and let c be any of the corners of a rectangle centered at the origin. To compute the ellipsoid distance of the corner c to the origin 0, consider the following transformation: $c \cdot A \cdot c^T = c \cdot VW^2V^T \cdot c^T = (cV) \cdot W^2 \cdot (cV)^T = (cVW) \cdot (cVW)^T$. These terms are quadratic forms for the matrices A, W^2 and Id applied to the vectors c, cV and cVW, respectively, i.e. $d_A(c, 0) = d_{VW^2V^T}(c, 0) = d_{W^2}(cV, 0) = d_{Id}(cVW, 0) = |cVW|$.

Due to the orthogonality of V, all the original corner points c and transformed corners cV have the same Euclidean distance $|c| = |cV|$ to the origin. Only after the iso-oriented weighting by W, i.e. stretching the axis e_i by the weight w_i, the lengths $|cVW|$ differ. Let e_{max} be the axis corresponding to the maximum weight $w_{max} = \max\{w_i\}$. The maximum value of $|cVW|$ is obtained for the corner cV that is closest to the axis e_{max}, i.e. the angle between cV and e_{max} is minimal:

$$\mathrm{angle}(cV, e_{max}) = \mathrm{acos}\frac{cV \cdot e_{max}}{|cV| \cdot |e_{max}|} = \mathrm{acos}\frac{c \cdot v_{max}}{|c|}$$

Proof. At all, the corner c_{max} having the maximal distance value $d_A(c_{max}, 0)$ is the corner that is closest to v_{max}, i.e. the eigenvector corresponding to the maximal eigenvalue. ❑

The corner closest to the largest eigenvector can be determined in linear time by considering the sign of each component of the eigenvector: If the sign is positive, we take the upper boundary of the rectangle in this dimension, otherwise the lower boundary. If a component is 0, it makes no difference whether we take the lower or upper boundary because both corners yield equal distance:

```
Algorithm A1:
    closest (e: vector [d], l: vector [d], u: vector[d]): vector[d]
        for i:=1 to d do
            if e[i] >= 0 then closest[i] := (u[i] - l[i])/2 ;
                         else closest[i] := (l[i] - u[i])/2 ;
```

If c is the closest corner determined by algorithm A_1, the distance d_{MBE} is the maximum ellipsoid distance between the center of the grid cell and an arbitrary point in the cell where d_{MBE} is given:

$$d_{MBE} = \sqrt{c \cdot A \cdot c^T}$$

For the distance between an arbitrary point p in the cell with center point p_c and the query point q, the following inequality holds:

$$d_A(p, q) \geq d_A(p_c, q) - d_{\text{MBE}}$$

This is simply a consequence of the triangle inequality which is valid for ellipsoid distances. The computation of d_{MBE} has a quadratic time complexity, because the multiplication of the matrix and a vector is quadratic. In the next section, we will develop an additional approximation which is an upper bound of d_{MBE}. Given a query object and a database object, these approximations can be evaluated in time linear in the number of dimensions.

3.2 The Rhomboidal Ellipsoid Approximation (RE)

The computation of the approximated distance $d_A(p_c, q) - d_{\text{MBE}}$ is quadratic in the number of the dimensions. This is already much better than determining the exact distance between query point and approximation cell which can only be done by expensive numerical methods. In order to further reduce the computational effort, we develop in this section an additional filtering step which is *linear* in the number of dimensions. It would be possible to avoid the computation of d_{MBE} for each item stored in the VA-file, if we would determine $d_{\text{MBE,max}}$ i.e. the ellipsoid approximation of the largest cell stored in the VA-file. Unfortunately, the sizes of the grid cells vary extremely for real data sets due to the quantile concept. Therefore, $d_{\text{MBE,max}}$ is a bad approximation of most of the grid cells. The size of an individual cell must be considered in a suitable way.

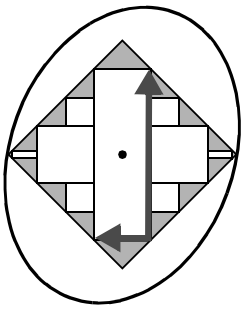

Fig. 3. Rhomb. ellips. app.

Our way to consider the cell size is based on the sum of the side lengths of the cell. This sum can obviously be determined in O(d) time. The idea is to determine in a preprocessing step an ellipsoid which contains all cells having a standardized sum of the side lengths. If the actual sum of the side lengths of a grid cell is larger or smaller than this standardized value, the ellipsoid is scaled, which is an operation of constant time complexity.

As depicted in figure 3 the hull of all rectangles having a sum $s = \sum_{i=1\ldots d} s_i$ of the extensions s_i of s forms a rhomboid with diameter s or "radius" $s/2$. The corners of the rhomboid are the unit vectors, each multiplied with $+s/2$ and $-s/2$, respectively. Due to convexity and symmetry, the minimum bounding ellipsoid of the rhomboid touches at least two of the corners, which are on opposite sides with respect to the origin.

This observation leads to the following lemma:

Lemma 2. For a similarity matrix $A = [a_{ij}]$, the minimum bounding ellipsoid over all cells that have a sum $s = \sum_{i=1\ldots d} s_i$ of extensions has a distance value of

$$d_{\text{RE}}(s) = \frac{s}{2} \cdot \max\{\sqrt{a_{ii}}, i = 1\ldots d\}$$

Proof. The minimum bounding ellipsoid touches the rhomboid at the corners. Due to symmetry, we can restrict ourselves to the positive unit vectors scaled by $s/2$, i.e. $e_i \cdot s/2$, $i = 1\ldots d$. The rhombus is touched at the unit vector which has the largest ellipsoid distance from the center, because all other corners must not be outside of the ellipsoid, and this distance equals to $d_{\text{RE}}(s)$.

Proof. The ellipsoid distance between the i-th unit vector e_i and the origin 0 is $d_A(e_i, 0) = \sqrt{e_i \cdot A \cdot e_i^T} = \sqrt{a_{ii}}$, and the maximum over the dimensions $i = 1\ldots d$ is the square root of the maximal diagonal element of A, i.e. $\max\{d_A(e_i, 0)\} = \max\{\sqrt{a_{ii}}\}$. Scaling by $s/2$ is commutative with d_A and with the maximum operator and, hence, the proposition holds. ❑

With lemma 2, we can estimate the maximum distance between a point approximated by a grid cell and the center of the grid cell. In contrast to the distance d_{MBE} introduced in section 3.1 this bound can be determined in linear time. It is an upper bound of d_{MBE}, as can be shown as follows :

Lemma 3. $d_{\text{RE}}(s)$ is an upper bound of d_{MBE}.

Proof. Let Rh(s) be the locus of all points p where $\sum_{i=1\ldots d} |p_i| \leq s/2$. By lemma 2, we know that for each point $p \in \text{Rh}(s)$ the ellipsoid distance to the origin $d_A(p, o)$ is at most $d_{\text{RE}}(s)$. Let C(s) be an arbitrary grid cell centered at the origin having a side sum of s. For every point

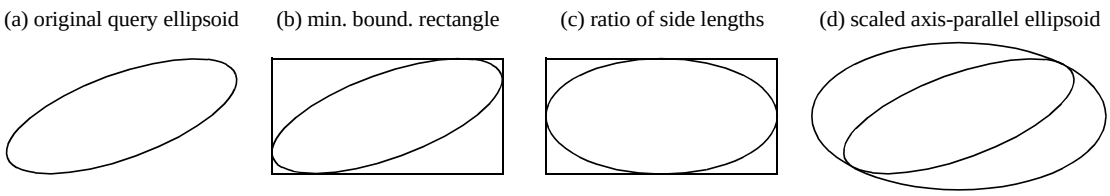

Fig. 4. Construction of the minimum axis-parallel ellipsoid for a given general ellipsoid

$q \in C(s)$ we know that $\Sigma_{i=1\ldots d} |q_i| \leq s/2$. It follows that $d_A(q, o) \leq d_{RE}(s)$. As d_{MBE} is the maximum ellipsoid distance of all points $q \in C(s)$, we have

$$d_{MBE} = \max\{d_A(q, o), q \in C(s)\} \leq d_{RE}(s) \qquad \square$$

Since the rhomboidal ellipsoid approximation is less exact than the minimum bounding ellipsoid approximation, it is likely to yield a worse filter selectivity (i.e. a higher number of candidates). However, it can be determined by far faster. We will see in section 4 that the determination of the rhomboidal ellipsoid approximation causes almost no extra cost compared to the MBE approximation, but avoids many expensive evaluations of the MBE approximation. Therefore, it is intended to apply the combination of these two filters.

3.3 Axis-Parallel Ellipsoid Approximation

In this section, we propose an additional filtering step which now approximates the query ellipsoid, not the grid cells. In the next section, we will show that it is beneficial to approximate both, queries and grid cells in separate filtering steps, because the two different approximation techniques exclude quite different false hits from processing, and, therefore, the combination of both methods yields a much better selectivity than either of the methods alone. We again propose an approximation technique which is particularly well suited for grid based quantization. In the VA-file (as well as in the IQ-tree and any grid based method) the computation of weighted Euclidean distances is very simple. Basically, it makes no difference whether to determine a weighted or an unweightet Euclidean distance. Therefore, we approximate the general query ellipsoid by the minimum axis-parallel ellipsoid that contains the query ellipsoid.

The axis parallel ellipsoid is constructed in two steps. In the first step, the ratio of the side lengths of the ellipsoid is determined. It corresponds to the ratio of the side lengths of the minimum bounding rectangle of the ellipsoid. In the second step, the axis-parallel ellipsoid is scaled such that it is a minimum bound of the original query ellipsoid. This is done by scaling both the original and the axis-parallel ellipsoid non-uniformly such that the axis-parallel ellipsoid becomes a sphere. The matrix corresponding to the scaled original query ellipsoid is then diagonalized to determine the smallest eigenvalue w_{min}. This eigenvalue corresponds to the factor by which the axis-parallel ellipsoid must be (uniformly) scaled such that it minimally bounds the original query ellipsoid. The process of constructing the minimum bounding axis-parallel ellipsoid is shown in figure 4.

Lemma 4. Among all axis-parallel ellipsoids our solution is a minimum bounding approximation of the query ellipsoid.

Proof. Follows immediately from the construction process. ❑

4. Analysis of the Filter Architecture

In this section, we will develop a filter architecture for adaptable similarity queries which is adjusted to the VA-file and other grid-based quantization methods and which optimizes query processing based on vector quantization. We have proposed three new approximation techniques for this purpose in section 3, the minimum bounding ellipsoid approximation, the rhomboidal ellipsoid approximation, and the minimum bounding axis-parallel ellipsoid. The first and second technique approximates the quantization cells by ellipsoids. The shape of these approximating ellipsoids is defined by the similarity matrix *A*. In contrast, the third technique approximates the query ellipsoid by an axis-parallel ellipsoid which corresponds to a weighted euclidean distance calculation. We are now going to discuss the various approximation techniques according to their most relevant parameters, computational effort and filter selectivity.

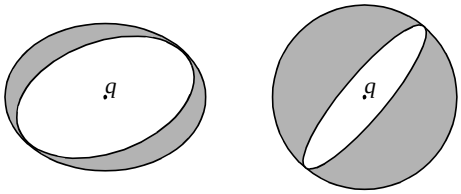

Fig. 5. Selectivity of axis-parallel approx.

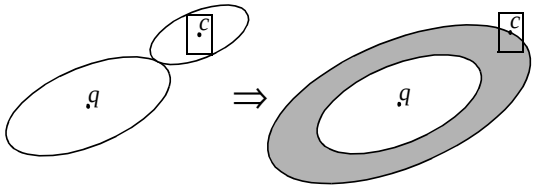

Fig. 6. Selectivity of the MBE approx.

4.1 Basic Approximations

The axis-parallel approximation of the query ellipsoid requires the determination of the smallest eigenvalue which is done in a preprocessing step when reading the similarity matrix for the query. Although this operation has a bad time complexity, preprocessing operations are still considered to be negligible as we generally assume $d \ll N$. Once the smallest eigenvalue is determined, we have only a linear effort O(d) per considered object. As discussed in section 2, the VA-file allows a particularly efficient determination of the weighted Euclidean distance between the query point and the approximation cell, because the squared differences between the quantiles and the query point can also be precomputed outside the main loop iterating over all objects. Therefore, d additions are the only floating point operations performed per object. To assess the selectivity of this filter step, refer to figure 5. Here, we distinguish between queries which are more or less axis-parallel. The left side of figure 5 shows an ellipsoid which is very well approximable. In contrast, the right side presents an ellipsoid with an angle close to 45°. The axis-parallel approximations are also depicted. Underlaid in gray is the volume which represents the approximation overhead. Objects from this area are candidates produced by the filter step, however, they are no hits of the similarity query. The smaller the gray overhead volume, i.e. the more axis-parallel the original query ellipsoid, the better is the filter selectivity. Assuming a uniform distribution of points in the data space, we would have a few percent overhead for the left query ellipsoid, but several hundred percent overhead for the right query ellipsoid. For less axis-parallel original ellipsoids, the filter selectivity is not sufficient.

The MBE approximation of the grid cells requires the highest CPU time of all considered filters. To determine whether a grid cell contains a candidate point or not, two ellipsoid distance computations must be performed: First, the distance between the point and the center of the cell and second, the distance between the center of the cell and the corner closest to the "largest" eigenvector have to be computed. To assess the selectivity of this filter, we apply a problem transformation. For our analysis, we assume that all grid cells have identical shapes and, therefore, the minimum bounding ellipsoids are all identical. This allows us to transform the problem in such a way that we add the range of the MBE to the range of the query, as depicted in figure 6: In the left part, the cell is a candidate if the query ellipsoid intersects the MBE approximation of the cell. In the right part, we have transformed the query ellipsoid into a larger ellipsoid, where the range is the sum of the two ranges of the query and the MBE. The cell is a candidate whenever the center of the cell is contained in the enlarged ellipsoid. This concept of transformation is called Minkowski sum [5]. Again, in the right part of figure 6, the approximation overhead is marked in gray. Using this approximation, the overhead heavily depends on the size of the MBE approximation.

The RE approximation of the grid cells is very similar in both, computation time and selectivity. In contrast to the MBE approximation, this techniques requires only one distance calculation: The distance between the query point and the center of the grid cell. That means, one application of this filter step causes about half the expenses compared to one application of the MBE filter, assuming that all other cost are negligible. For this technique, we have a similar selectivity assessment as for the MBE approximation depicted in figure 6. The only difference is that the ellipsoid approximating the cell is not minimal. Therefore, also the Minkowski sum and the implied approximation overhead are larger than the MBE overhead in figure 6.

Last but not least, we have to consider the cost of the refinement step. The computational cost is limited to a single ellipsoid distance computation, which is even less expensive than an application of the MBE filter and comparable to the cost of the RE filter. In contrast to all described filter steps, the refinement step works on the exact coordinates of the data point and not on the grid approximation. Therefore, the point must be loaded from background storage which usually caus-

es one costly seek operation on the disk drive. The refinement step is the most expensive step unless the distance calculation cost exceed the cost for a random disk access. Table 1 summarizes the results of this section.

4.2 Combined Approximations

Table 1: Cost and selectivity of filters

Technique	Cost	Selectivity
axis parallel ellip. approx.	O(d)	low
MBE approx.	$2 \cdot O(d^2)$	fair
rhomb. ell. ap.	$1 \cdot O(d^2)$	medium
exact distance	$1 \cdot O(d^2) + 1$ random I/O	exact

In this section, we show that the combined application of the developed filters is useful and results in systematic performance advantages compared to the application of a single filter. First we consider the combined application of the MBE filter and the rhomboidal ellipsoid filter. We first show that the combined application of MBE and RE does not cause any considerable extra cost compared to the application of the MBE filter alone. The application of the MBE filter requires the distance calculation between the query point and the center of the cell, and the distance calculation between the cell center and a cell corner. The RE filter requires the distance calculation between the query point and the center of the cell only. All other cost are negligible. If we first apply the RE filter and then the MBE filter, the MBE filter may reuse the distance evaluated by the RE filter. The combined application performs one distance calculation for all points and a second calculation for those points which are not excluded by the RE filter. Whether the filter combination is more efficient than the application of the MBE filter alone depends on the selectivity of the two filters. We know that the MBE filter is at least as selective as the RE filter. However, if the MBE filter does not save additional I/O accesses compared to the RE filter, the additional distance calculations lead to a performance deterioration. Since I/O accesses are usually much more expensive than distance calculations, the breakeven point can be reached when the MBE filter is only slightly more selective than the RE filter. Our experimental evaluation will show, that the selectivity of the MBE filter is significantly better. It is beneficial to apply the RE filter first and then the MBE filter, because the cost of the RE filter are lower.

Next we are going to show that it is beneficial to combine the axis-parallel approximation and the MBE approximation. In figure 8, we can compare the selectivities of the two filters. In contrast to the previously described combination, the filters are not lower bounds of each other, but access rather different parts of the data space. The area of the main overhead of the axis-parallel approximation is where the query ellipsoid is narrow (depicted on the left side of fig. 8). The MBE approximation, however, yields most of its overhead at the front and the bottom of the ellipsoid, where the axis-parallel approximation yields no overhead at all. Therefore, the combined filter yields a dramatically improved selectivity compared to the axis-parallel approximation as well as compared to the MBE filter, as depicted in the rightmost illustration in figure 8. As the axis-parallel approximation is by far cheaper than the MBE filter, it is natural to apply the axis-parallel approximation first. For the order of these filters, we consider the cost of the filter step per evaluated point:

$$C_{\text{axis-par}} << C_{\text{RE}} << C_{\text{MBE}} << C_{\text{exact}}.$$

Therefore, we apply the axis-parallel approximation as the first filter step. The second filter step is the RE filter. The MBE approximation is the third filter followed by the refinement step.

5. Experimental Evaluation

To evaluate our filter techniques experimentally, we have implemented a VA-file extension with our four-step query processing architecture. Our implementation in C was tested on a HP C-160

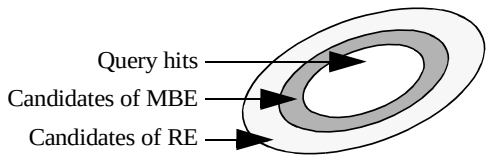

Fig. 7. Combination of RE and MBE filters

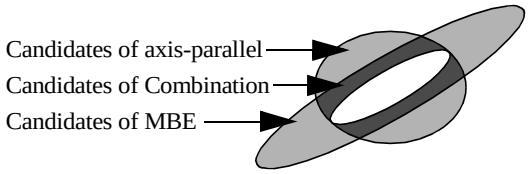

Fig. 8. Combined axis-parallel and MBE filter

workstation under the HPUX operating system. Both, vector quantizations and exact point data were stored on the same disk drive with an average seek time of 8 ms and a transfer rate of 4 MByte/s for continuous data transfer. The vector approximations were scanned in large blocks of 1 MByte to minimize the influence of rotational delays or disk arm repositionings between subsequent I/O requests.

Our reference application is a similarity search system for color images which allows a user-adaptable specification of the similarity measure based on color histograms [12]. Our data set contains 112,363 color images, each represented by a 64-dimensional color histogram. We separated our file into a large data file and a smaller query file by random selection. On our hardware, an evaluation of an ellipsoid distance calculation needs 60 µs and a Euclidean distance calculation requires about 0.5 µs. Our implementation performs ellipsoid range queries. Unless otherwise mentioned our experiments determine the 2 nearest neighbors of 10 query points.

We generate the similarity matrices $A = [a_{ij}]$ by the formula $a_{ij} = \exp(-\sigma(d(i,j)/d_{\max})^2)$ from [8] where $d(i,j)$ denotes the cross-similarity of the colors i and j. Since the adaptable similarity model supports the modification of local similarities, we introduce three additional parameters σ_r, σ_g, σ_b and define $d(i,j) = \mathrm{sqrt}(\sigma_r \Delta r^2 + \sigma_g \Delta g^2 + \sigma_b \Delta b^2)$ in the RGB color space. For our queries, we use five different parameter settings for the tuple (σ, σ_r, σ_g, σ_b), namely M_1 (1, 100, 1, 1), M_2 (1, 1, 1, 100), M_3 (100, 1, 1, 1), M_4 (100, 100, 1, 1, 1) and M_5 (1, 1, 1, 1).

In the previous sections, we have postulated several claims which require an experimental validation beyond the actual proof of superiority over competitive techniques. The most important claims to justify our four-step query processing architecture were

- the superiority of the combination of the axis-parallel approximation and the MBE filter and
- the benefit of the second filter step (RE approximation)

In our first experiment (cf. figure 9) we measure the selectivities of the axis-parallel approximation and the MBE filter both, separately as well as combined. As already pointed out in section 4, the quality of the axis-parallel approximation depends on the orientation of the query ellipsoid. Our ellipsoid M_1 has a bad selectivity (14,473 candidates equivalent to 12.5% of all points). Only ellipsoid M_3, which is almost a sphere, yields a satisfactory selectivity of 3.3 candidates which means an overhead of 1.3 candidates for 2 neighbors. The selectivity of the MBE filter, applied separately, is with 457 candidates for ellipsoid M_1 by far better than the selectivity of the axis-parallel approximation, but not satisfactorily. For the queries M_3, M_4 and M_5, the selectivity of the MBE filter is partly better, partly worse than that of the axis-parallel approximation. The combination of both filters, however, outperforms the separate applications of the approximations by large factors (up to 176 compared to the axis-parallel approximation and up to 5.6 compared to the MBE filter). Only for ellipsoid M_3 where the axis-parallel approximation is almost optimal the combination does not yield further selectivity improvements.

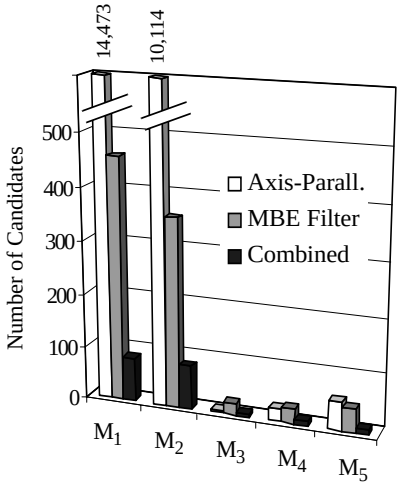

Fig. 9. Selectivity of filters.

For the ellipsoids M_1 and M_2, the RE filter yields around 12,000 candidates, and thus a selectivity in the same order of magnitude as the axis-parallel approximation. Now one may wonder whether the second filter step is useful at all. Moreover, for ellipsoid M_5, the number of candidates of the RE filter is even by a factor of 250 worse than the axis-parallel approximation. Our next experiment depicted in figure 10, however, shows that the RE filter, like the MBE filter, is beneficially combinable with the axis-parallel approximation. As we have pointed out in section 4, the RE approximation as an additional filter step is justified if it yields fewer candidates than the axis-parallel approximation but substantially more candidates than the MBE filter (otherwise, the MBE filter would be unnecessary). For the first two ellipsoids every filter step reduces the candidate set approximately to 10%. Note that the scale in this figure is logarithmic. In ellipsoid M_1, for instance, the axis-parallel approximation reduces the candidate set from 112,000 to 14,473 (12.9%). The second filter step reduces this set further to 1,458 (10.1% of 14,473) and the third filter step to 82

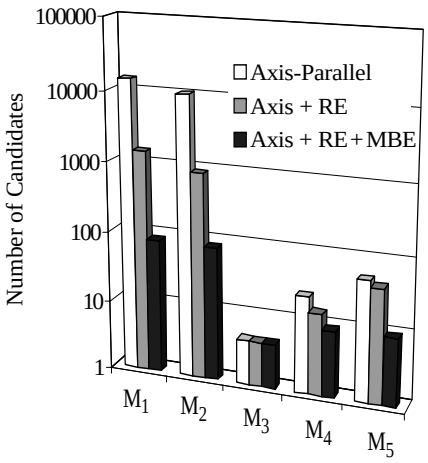

Fig. 10. Comparing architectures (# candid.)

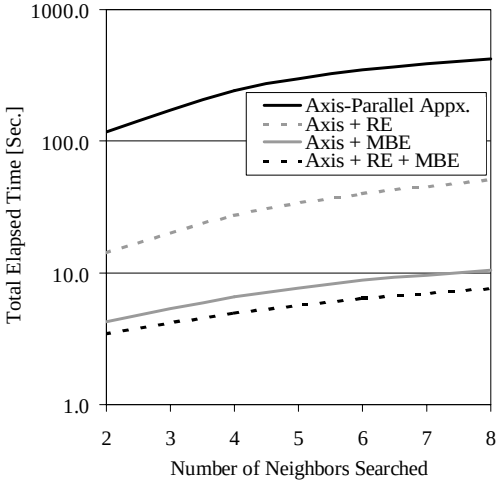

Fig. 11. Comparing architectures (time)

(5.6% of 1,458). Even for ellipsoid M_5, for which the selectivity of the RE filter yielding 13,216 candidates (cf. figure 10) is extremely bad, the combination of the axis-parallel approximation and the RE filter clearly outperforms the axis-parallel approximation by a factor 1.2.

The next experiment demonstrates the impact of the various numbers of candidates on the overall runtime of a query. Therefore, we compare the runtime of our four-step architecture with a two-step architecture (axis-parallel approximation and refinement step) and also with two architectures consisting of three steps: (1) axis-parallel approximation, RE approximation, refinement step; (2) axis-parallel approximation, MBE approximation, refinement step. Figure 11 shows the results of our experiments using ellipsoid M_1 in logarithmic scale. In this figure the selectivity is varied such that the number of retrieved neighbors ranges from 2 to 8. The performance of the two-step architecture performing merely the axis-parallel approximation upon the VA file is very bad (119 to 234 seconds of total time). It is clearly outperformed by the three-step and four-step architectures using our new types of cell approximation in combination with the axis-parallel approximation. The architecture with the additional rhomboidal ellipsoid filter requires between 14 and 51 seconds and thus is up to 8.5 times faster. The improvement factor of the three-step architecture with the MBE filter is even higher, up to 28.3. The constantly best performance shows our four-step architecture: 3.5 to 7.7 seconds total elapsed time with an improvement factor of 34 over the two-step architecture.

In a last series of experiments we compare our new technique (four-step architecture) with two competitive techniques, the sequential scan and the X-tree index. Both competitive techniques are

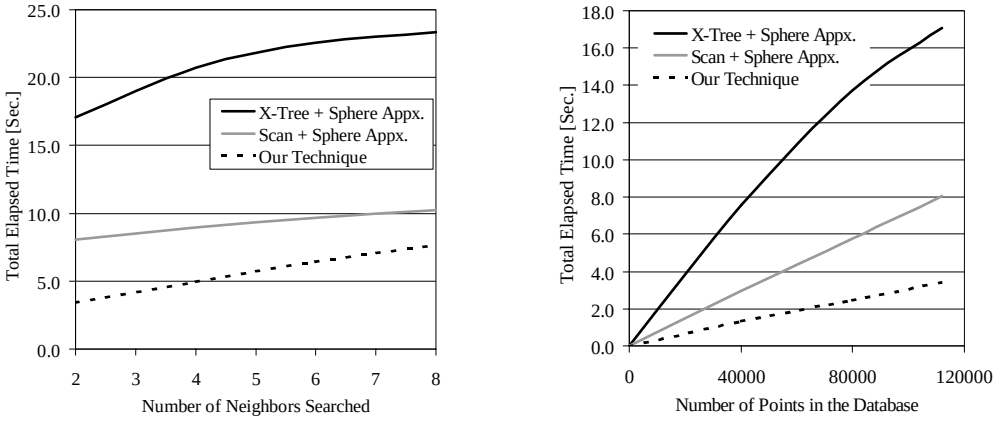

Fig. 12. Comparison with varying selectivity (l.) and scalabiltiy (r.)

allowed to use the sphere approximation as a filter step. In the left diagram of figure 12, we measure again the total processing time with varying selectivity. The sequential scan which requires between 8.1 and 10.2 seconds is outperformed by factors between 1.3 and 2.3, and the X-tree index (17.1 to 23.4 sec.) is outperformed by factors between 3.0 and 4.9. The right diagram shows the same experiment with varying database size. With increasing database size, the improvement factor over the X-tree slightly increases from 4.7 (40,000 points) to 5.0 (112,000 points) while the improvement factor over the sequential scan remains constant at 2.3.

6. Conclusions

In this paper, we have proposed three new approximation techniques to cope with the problem of efficiently processing user adaptable similarity queries on quantized vectors. The first and second filter approximate the cells of the quantization grid by ellipsoids with the same principal axes as the query ellipsoids and enable us to exploit the triangle inequality. The MBE approximation determines the minimum bounding ellipsoid for each quantized vector and, hence, requires a quadratic time complexity. In contrast, the RE filter requires linear time in query processing. Our third new filter technique approximates the query which is a general ellipsoid (corresponding to a quadratic form distance) by its minimum bounding axis-parallel ellipsoid. Axis-parallel ellipsoids correspond to weighted Euclidean distances which can be evaluated with particular efficiency on grid based query processing techniques such as the VA-file or the IQ tree. We propose a multistep query processing architecture with three filter steps (our new axis-parallel approximation and our new filters RE and MBE) and show the superiority over architectures with fewer filter steps and over competitive techniques theoretically as well as experimentally. Our anylysis demonstrates that our filters complement each other. Hence, it is useful to combine our three filters, and we determine a suitable order of the filter steps. In our experimental evaluation, the sequential scan is outperformed by a factor of 2.3. Compared to the X-tree on 64 dimensional color histogram data, we measured an improvement factor of 5.7. For our future work, we plan to integrate our new approximation techniques into the IQ-tree.

References

1. Ankerst M., Braunmüller B., Kriegel H.-P., Seidl T.: *Improving Adaptable Similarity Query Processing by Using Approximations.* Proc. 24th Int. Conf. on Very Large Data Bases (VLDB), 1998, 206-217.
2. Ankerst M., Kastenmüller G., Kriegel H.-P., Seidl T.: *3D Shape Histograms for Similarity Search and Classification in Spatial Databases*. Int. Symp. on Spatial Databases (SSD), LNCS 1651, 1999, 207-226.
3. Ankerst M., Kriegel H.-P., Seidl T.: *A Multi-Step Approach for Shape Similarity Search in Image Databases.* IEEE Transactions on Knowledge and Data Engineering (TKDE) 10(6), 1998, 996-1004.
4. Berchtold S., Böhm C., Jagadish H. V., Kriegel H.-P., Sander J.: *Independent Quantization: An Index Compression Technique for High-Dimensional Data Spaces*, Int. Conf. on Data Engineering (ICDE), 2000.
5. Berchtold S., Böhm C., Keim D., Kriegel H.-P.: *A Cost Model For Nearest Neighbor Search in High-Dimensional Data Space*. ACM PODS Symposium on Principles of Database Systems, 1997.
6. 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.
7. 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.
8. 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 (PAMI) 17(7), 1995, 729-736.
9. Ishikawa Y., Subramanya R., Faloutsos C.: *MindReader: Querying Databases Through Multiple Examples*. Proc. 24th Int. Conf. on Very Large Data Bases (VLDB), 1998, 218-227.
10. Kriegel H.-P., Seidl T.: *Approximation-Based Similarity Search for 3-D Surface Segments*. GeoInformatica Int. Journal, Vol. 2, No. 2. Kluwer Academic Publishers, 1998, 113-147.
11. Lin K., Jagadish H. V., Faloutsos C.: *'The TV-Tree: An Index Structure for High-Dimensional Data.* VLDB Journal 3(4), 1994, 517-542.
12. 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.
13. Smith J. R.: *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis.* Ph.D. thesis, Graduate School of Arts and Sciences, Columbia University, 1997.
14. White D.A., Jain R.: *Similarity indexing with the SS-tree.* Int. Conf on Data Engineering (ICDE), 1996.
15. Weber R., Schek H.-J., Blott S.: *A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces.* Int. Conf. on Very Large Databases (VLDB), 1998, 194-205.