Ira Assent, Marc Wichterich, Thomas Seidl

Adaptable Distance Functions for Similarity-based Multimedia Retrieval

Today's abundance of storage coupled with digital technologies in virtually all scientific or commercial applications such as medical and biological imaging or music archives deal with tremendous quantities of images, videos or audio files stored in large multimedia databases. For content-based data mining and multimedia retrieval purposes, suitable similarity models are crucial. Adaptable distance functions are particularly well-suited to match the human perception of similarity. Quadratic Forms (QF) were introduced to capture the notion of inter-feature similarity which sets them apart from the more traditional feature-by-feature measures from e.g. the Euclidean or Manhattan dissimilarity functions. The Earth Mover's Distance (EMD) was adopted in Computer Vision to better approach human perceptual similarities by allowing feature transformation under a number of restrictions. After recapping the concepts of distance-based similarity search in databases, we familiarize the reader with the flexible building stones behind Quadratic Forms and the EMD. These enable their application to a large variety of multimedia retrieval problems. Unfortunately, the flexibility comes at a cost. Their computation is relatively time-consuming, which severely limits its adoption in interactive multimedia database scenarios. Therefore, we research methods to speed up the retrieval process and show some encouraging recent results to achieve just that via an index-supported multi-step algorithm based on new lower bounding approximation techniques.

1 Introduction to Similarity Search

Multimedia similarity search is a new challenge which arises from the diversity of applications generating images, videos and other non-text data.

For example, for today's digital production of advertisements based on large stock footage collections, it is essential to search for images by their content. When looking for e.g. a sunset, it is desirable to search for images similar to a given sunset or a sketch by the user. Searching for images by metadata, such as the title or manually annotated keywords is not feasible for larger image databases where millions of images have to be labeled. Moreover, manual annotations are necessarily subjective and not complete.

Therefore, in content-based similarity search, the idea is to derive the image content description automatically [Smeulders et al. 2000]. For images, characteristics such as the color distribution, the texture or objects contained might be compared for similarity.

In science, the search for similarity among gene sequences or protein structures has opened up a new field for thinking about which biological objects are similar. By detecting these similarities, drug generation is facilitated and the function of genes or proteins can be better understood.

Many more examples from sensors generating large amounts of scientific data, news broadcasting being stored in large video databases and so on demonstrate the need for meaningful and interactive similarity search.

For all these application areas, appropriate similarity models have to be created. This involves deciding which characteristics are relevant for the task at hand, how they can be compared, and last, but not least, how this comparison can be done in reasonable time even for the large databases typically found in these applications.

1.1 Query Types

Similarity Search can be further categorized into several query types that are useful for different applications. For some users, only the most similar object compared to a query object might be of interest. Others might not know how many results they require but might instead know what they do not accept as similar anymore. Three such query types are frequently investigated and formulated below.

Range Queries: Given a tolerance parameter ε that describes the maximum acceptable dissimilarity, a range query $range(q,\varepsilon)$ retrieves all objects from the database which are not more dissimilar to q than ε allows for. The number of returned objects is not known a priori.

Nearest Neighbor Queries: Given a natural number k, a Nearest Neighbor query $nn(q,k)$ returns the k most similar objects compared to q. The object returned for $nn(q,1)$ is said to be the (or a) nearest neighbor. Unlike the range query, the cardinality of the returned set is given through k. It is not known a priori, how dissimilar the results are compared to q.

Incremental Ranking Queries: Given a query object q, a ranking query $rank(q,i)$ in iteration i returns the object that was not already returned by any $rank(q,j)$ with $1 = j \leq i$ where $rank(q,1) = nn(q,1)$. Thus, a ranking query follows the »give me more« concept, giving more control to the user.

1.2 Feature Extraction and Similarity

To mathematically capture the similarity between any two objects in a database, the most common approach is to first use a preprocessing step to extract the most significant features of each object.

In this context, histograms partition the feature space into regions or 'bins'. E.g., a color histogram partitions the HSV, HLS, CIE Lab or any other suitable color space into a number of color regions and counts the pixels of a picture that are assigned to each of those color regions [Smith 1997]. Similarly, histograms can be defined for other feature spaces such as spatial histograms that are used for shape similarity (see fig. 1) or frequency histograms which can be extracted from audio data.

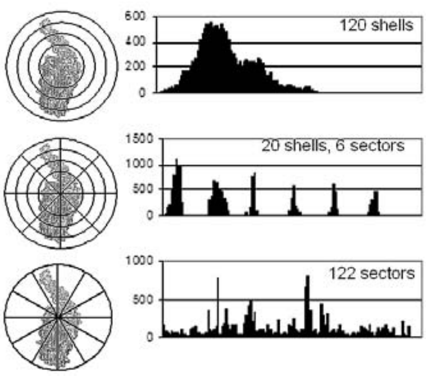


Fig. 1: Three shape histograms for molecule data

Instead of partitioning the feature space in the same manner for all objects, this process can also be realized in an adaptive way. For example, a picture containing a sunset could differentiate between multiple shades of red, blue and yellow while putting less emphasis on the various green shades. Adaptive binning potentially increases the quality of the feature extraction process but makes the comparison of different objects more complicated.

After extracting the features x and y of two objects, a distance function *dist* compares x and y. For metric distance functions, the smallest possible value of $dist(x, y) = 0$ is reserved for x equal to y and tells us that the according objects are to be considered identical for the purpose of the application at hand. Larger values tell us that they are different or dissimilar to some degree.

Of course, the distance function can only utilize the features extracted in the preprocessing step. In addition, there generally is no way to devise a formal model that perfectly reflects the human sense of similarity. Any similarity search in this context is intrinsically subjective. However, some families of distance functions allow for a closer match than others through multiple degrees of freedom of adaptation that can be specified by the user or a domain specialist.

1.3 Types of Distance Functions

We give an overview over commonly used distance functions, focusing on adaptable models. For additional reading on similarity models see [Schmitt 2005].

The most commonly encountered distance functions probably are the L_p norm-based ones of which the Euclidean Distance $dist_{L2}$ reflects our notion of distance in space. They can be applied to any two d-dimensional feature vectors $x = (x_1, \ldots x_d)$, $y = (y_1, \ldots, y_d)$ via

$$dist_{L_p}(x, y) = (\sum_i |x_i - y_i|^p)^{1/p}.$$

As evident from the formula, there is little freedom to adapt the distance between two objects outside of the feature extraction step. One could introduce weights per dimension to emphasize some feature over another. However, a second and possibly more severe limitation remains: The Euclidean Distance and its variants compare two feature vectors in a component by component manner. This makes sense when the components are orthogonal (or not comparable) to another. In many cases, extracted feature components are not orthogonal though. For example, the extracted features of a color histogram represent regions in a color space where some regions are more similar to another than others. A crimson red color region is closer to a scarlet red than to a royal blue region. While the first possible shortcoming can be remedied by e.g. introducing weights per dimension, the second one requires a move to fundamentally different distance functions.

There are a number of distance functions that are both user-adaptable in an intuitive way and not limited to feature component by feature component computation. Two functions successfully applied in a multitude of practical scenarios and of ongoing interest in research shall be the focus of the remainder of the article, Quadratic Forms and the Earth Mover's Distance.

2 Quadratic Forms

As mentioned above, incorporating perceived similarities between features leads to more appropriate and flexible distance functions. Quadratic Forms adopt a similarity matrix which encodes these inter-feature similarities. By formalizing that crimson is very similar to scarlet red, but less similar to royal blue, knowledge about features is taken into account.

More formally, the quadratic form distance between two feature vectors x and y according to a similarity matrix A is

$$dist_{QF}(x, y) = \sqrt{(x-y)A(x-y)^T} \;.$$

Quadratic Forms can be tailored to specific applications and even individual user preferences by modifying the similarity matrix. Especially when the notion of similarity changes, such a flexibility is very helpful. In the image similarity scenario, users may be interested in different aspects of similarity for different similarity queries.

2.1 Applications of Quadratic Forms

In IBM's QBIC (Query By Image Content) similarity search, Quadratic Forms were used to compare features such as color or texture histograms. To ensure that the overall similarity is in agreement with human perception, great care was taken to determine a similarity matrix consistent with human cognition as found in psychological experiments [Hafner et al. 1995].

Another application using Quadratic Forms as a distance function for retrieval is the MindReader framework [Ishikawa et al. 1998]. To grasp correlations in user queries, Quadratic Forms are chosen to incorporate multiple aspects for similarity retrieval. Moreover, for relevance feedback, adapting the similarity matrix to user input allows the system to iteratively refine the output.

In bioinformatics, shape histograms were used to model the geometric properties of molecules which were then compared using Quadratic Forms [Ankerst et al. 1999]. Shape histograms divide the feature space into cells. Features are extracted by recording the occupancy of a cell by each molecule. The spatial neighborhood relationship of cells was formalized by the similarity matrices of Quadratic Forms. This results in a far better similarity assessment than e.g. by applying Euclidean distance. Figure 1 illustrates some models for shape histograms.

3 The Earth Mover's Distance

Another approach towards adaptable similarity search, the so-called Earth Mover's Distance (EMD), has been recently proposed in Computer Vision. A cost matrix encoding the dissimilarity between individual features is used to adapt to application specific similarity. Unlike Quadratic Forms, the idea is not to calculate weighted differences, but to find the best match between two features as a measure for how similar they are.

The name »Earth Mover's Distance« stems from an intuitive explanation of the underlying similarity model. Think of one feature set as a collection of earth hills, and the other one as earth holes. By moving the earth from the hills to the holes, a possible match between the features is found. The distance is the overall cost of the best match between hills and holes based on the individual costs between features.

More formally, the Earth Mover's Distance between two feature vectors

x and y according to a cost matrix $C=[c_{ij}]$ is

$$dist_{EMD}(x,y)=\min_{F\in AF}\{\sum_i\sum_j f_{ij}c_{ij}/\sum_{i,j}f_{ij}\},$$

where AF denotes the set of all admissible flows $F=[f_{ij}]$ between the individual feature components of x and y. This means, as depicted in figure 2, that the Earth Mover's Distance is the minimal overall cost incurred when moving earth from x to y, ensuring that

a. no more earth is moved from any one feature component of x than available in that feature component,
b. no more earth is moved to any one feature component of y than allowed by that feature component,
c. only non-negative amounts of earth are moved and
d. the entire earth is moved. (In case of unequal mass, at least the total amount of earth in the smaller of the two features x and y is moved.)

3.1 Computation

When phrased in this way, the EMD can be viewed as a linear optimization problem. The target function to be minimized is the linear combination of all components in terms of »flow times cost«. The restrictions regarding the possible flow of the earth can be modeled as a number of linear constraints in a straightforward way. Thus, the simplex algorithm or other linear programming techniques can be used to calculate the EMD between two objects represented by their features. However, the special structure of the linear optimization problem allows for a more efficient computation. Through using the intuition of moving earth to interpret the optimization as a transportation problem, the so-called streamlined simplex algorithm can be applied to calculate the same distance [Hillier & Lieberman 1990]. This algorithm has previously been of much interest in other fields such as operations research.

Analogously to the similarity-matrix of the quadratic forms, the cost matrix can be adapted to reflect the dissimilarity among features. A large value for c_{ij} discourages earth to be moved from feature i to feature j, while a value close to zero encourages this combination.

3.2 Applications of the EMD

The Earth Mover's Distance can be used to capture similarity in a variety of multimedia settings. As expert knowledge on the underlying differences in feature space can be easily formalized in a cost matrix, this similarity model has recently caught a lot of attention. EMD can be used to match objects of different sizes and even incorporates partial matching (e.g. searching for images that contain a similar area to a query area).

Most prominently, EMD has been thoroughly studied in Computer Vision, where several feature extraction models have been successfully employed and analyzed. Besides color distributions, combinations of color and location, textures of various kinds have been shown to fit the EMD well. By visualizing the results using Multi-Dimensional Scaling (MDS), users can easily grasp the power of the model. The images seem to automatically fall into place according to directionality and »scality«, i.e. texture size (see fig. 3 [Rubner & Tomasi 2001]).

Several region-based image retrieval frameworks have been suggested, focusing on different aspects of similarity retrieval, such as navigating through image databases [Rubner & Tomasi 2001], learning user interest in a relevance feedback scheme for region-based image retrieval [Jing et al. 2004], sparse texture representation [Lazebnik et al. 2003], etc.

EMD has been studied in other application areas as well. In contour matching for shape similarity low-distortion embeddings of the EMD are used to grasp the shape of objects and in a graph-based approach the similarity between different gesture images is well modeled using the EMD-matching approach [Grauman & Darrell 2004, Demiric et al. 2004]. In physics, vector fields may be characterized using critical points. Combined with their associated parameters, these features can be compared using EMD [Lavin et al. 1998]. Another example stems from music retrieval, where features are based on time, pitch and note durations which are then compared using the EMD and a less-complex variation thereof [Typke et al. 2004].

4 Speeding things up

4.1 Indexing

While adaptable distance measures like Quadratic Forms and the Earth Mover's Distance have proven their usefulness in a number of applications, their complexity hinders direct applicability to settings with higher dimensionality and / or large databases.

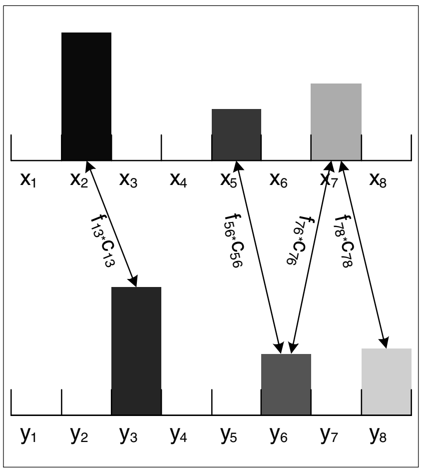


Fig. 2: Earth Mover's Distance

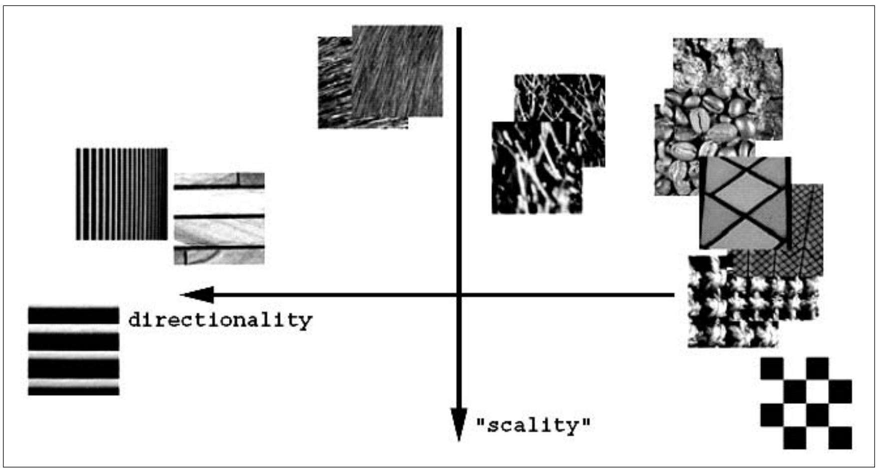


Fig. 3: EMD-based MDS Display of image textures

For multimedia database retrieval, some speed-up techniques have been suggested. In *multidimensional index structures*, the main idea is to use the information contained in the individual dimensions [Böhm et al. 2001]. *Hierarchical index structures* create a small-size directory of the entire database based on those dimensions. This directory is used to direct the similarity search process such that only a small, relevant, part of the database needs to be accessed in order to determine the result set.

To do so, multidimensional features are grouped together according to their respective values. They are summarized by bounding geometries kept at the next higher levels in the directory. During retrieval, the query is compared to these bounding geometries to direct the search towards the answer (e.g. R-Tree or X-Tree [Guttman 1984], [Berchtold et al. 1996]). In figure 4, two-dimensional points are grouped in a hierarchy of minimum bounding rectangles. These rectangles are stored in a tree (fig. 5). Starting at the root, the query is compared to the rectangles, thus choosing only paths down to the data that are relevant for the query.

While index structures have been proven to enable efficient query processing in numerous applications, it has been shown that their performance drops as dimensionality increases. Eventually, they fall prey to the so-called »curse of dimensionality«. This refers to the fact that in higher dimensional data sets, it is eventually impossible to find small bounding geometries: As large geometries tend to overlap, several query paths have to be used, which in turn means that large portions of the database have to be accessed.

In summary, index structures are helpful in reducing response times only for smaller dimensional problems.

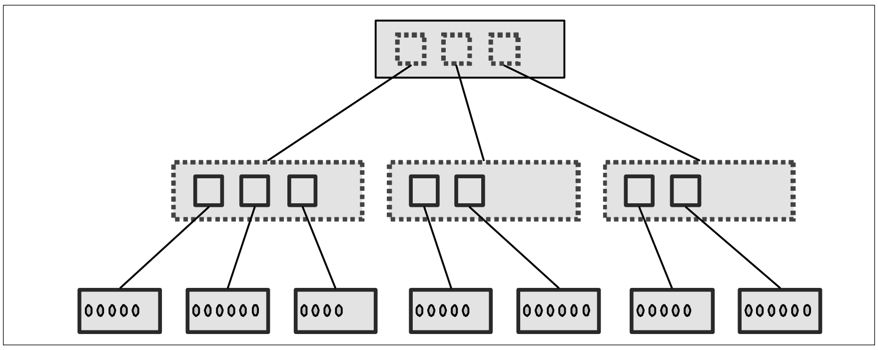


Fig. 5: Directory of hierarchical multi-dimensional index

4.2 Dimensionality Reduction

To benefit from index support for higher dimensional settings, multistep query processing has been proposed. The idea is to efficiently pre-sort the data in terms of relevance to the query using a filter distance function.

The first variant thereof is called dimensionality reduction. The basic idea is to determine a lower-dimensional feature representation of the original high-dimensional dataset. On these lower dimensional features index support can be used to efficiently determine a set of candidates. This candidate set is then refined using the original, high-dimensional data to obtain the final result.

The second version is similar in nature in that a candidate set is efficiently determined in a first step. To determine this set, any filter distance function may be used.

Intuitively, one can think of the filter as a rough, easy estimate at similarity. This estimate is verified using the exact distance. If the estimate is good – in the sense detailed below – the same result is obtained at reduced cost (see fig. 7).

To ensure the filter actually leads to a considerable speed-up, the following criteria should be met (summarized in fig. 6):

Index: as outlined above, multidimensional indexing structures such as R-Tree or X-Tree are useful for efficient query processing. To benefit from indexing structures in multistep query processing, filter distances have to be applicable to multidimensional indexing structures as well. To ensure this, dimension-wise distance computation is essential in that multidimensional indexing structures base their directory support on this information.

Completeness is important to ensure that there are no false drops in multistep query processing, i.e. in the filter step no actual result object should be discarded. For multistep algorithms as outlined below in Section 4.3, completeness can be assured by proving a lower bounding property of filter and exact distance (i.e. the value of the filter is always less than or at most equal to that of the exact distance).

Efficient single object pair filter distance computation is important for acceleration of the total response time.

Good *selectivity* means that the filter distance should produce a candidate set which is as small as possible. Good selectivity is achieved by a filter which approximates the true distance as closely as possible, i.e. which returns preferably

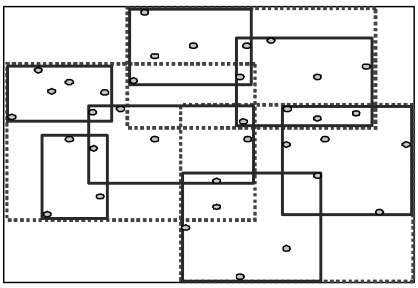


Fig. 4: Minimum bounding rectangles

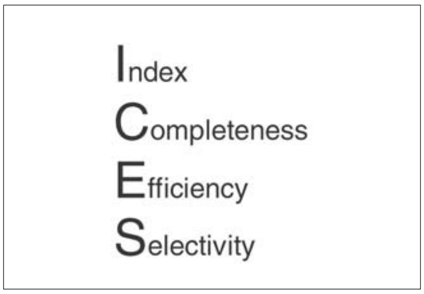


Fig. 6: ICES quality criteria for filters in a multistep

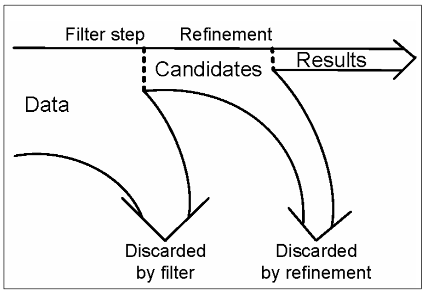


Fig. 7: Multistep Query Processing

large filter distance values without violating the lower bounding property (completeness).

4.3 Query Processing

A filter function, once developed, has to be integrated properly with query processing. For range queries, the algorithm triggers a range query on the index with the same range value for the filter distance as for the actual distance function.

For k nearest neighbor queries, where the distance of the k nearest neighbor is unknown, the GEMINI (*GE*neric *M*ultimedia object *IN*dex*I*ng [Faloutsos 1996]) algorithm first retrieves the k nearest neighbors according to the filter distance from the index. It determines the maximum actual distance for the answers and reports all objects whose filter distance to the query object is less than or equal to this maximum by a range query.

This algorithm can be further improved by changing the order in which the objects are processed. In our *K N*earest neighbor *O*ptimal *P*rocessing method, or KNOP for short, the idea to this optimization is that the filter step reports new candidates in ascending filter distance order, and termination is controlled by a feedback from the refinement step [Seidl & Kriegel 1998].

This solution 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.

5 Efficient similarity search with Quadratic Forms

5.1 Dimensionality Reduction for Quadratic Forms

Computation for a Quadratic Form is quadratic in the number of dimensions which is unacceptable for high dimensional settings and large databases.

Reducing the dimensionality of Quadratic Forms, from a geometric perspective, is a projection of the high-dimensional ellipsoid (given by the similarity matrix A) iso-distance contours to the low-dimensional data space. This projection is obtained from a transformation of the d-dimensional similarity matrix A to an r-dimensional similarity matrix A'. Obviously, care must be taken in the reduction in order to produce the results (completeness).

We develop a solution which proceeds in two phases [Seidl & Kriegel 1997]. First the matrix A is transformed by a multiplication with the complemented reduction matrix R', resulting in $R'^T A R'$. Second, performing a step by step reduction from d dimensions 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. The resulting reduced Quadratic Form using A' is a lower bound of the original Quadratic Form using A. Moreover, it is also the greatest of all lower-bounding distance functions in the reduced space and is therefore optimal as no other complete filter distance function in the reduced space produces fewer candidates.

5.2 Geometric Approximations for Quadratic Forms

Another solution for Quadratic Forms stems from geometric approximations [Ankerst et al. 1998]. The ellipsoid iso-contours of Quadratic Forms can be circumscribed by Minimum Bounding Boxes (MBBs) or Minimum Bounding Spheres (MBSs) which can be used as complete filter functions in a candidate generation step.

The MBB of an object is the smallest rectilinear hyperrectangle totally enclosing the object (see fig. 8). The box representation is very compact using just $2d$ parameters in a d-dimensional data space via storing lower and upper bounds in each dimension. It is compatible to multidimensional index structures and can typically be efficiently computed. Thus, in terms of our ICES-filter criteria, determining these enclosing geometric approximations yields efficient and effective multistep algorithms.

Generalizing boxes to distance functions, which is required for kNN and incremental ranking queries, yields weighted maximum norms. The iso-contour of the maximum norm is a hyper-cuboid. By weighting individual dimensions, the shape of the cuboid is stretched to the MBB. It has been shown that the MBB distance function

$$LB_{MBB(A)}(x, y) = \max_{1 \le i \le d} \left\{ \frac{|x_i - y_i|}{\sqrt{A_{ii}^{-1}}} \right\}$$

is the minimum bounding box distance function of the QF using A.

A tangential point shared by the box distance and the Quadratic Form is constructed to show that this is the minimum bounding box. This means that filters using this MBB definition are complete and optimal.

In the same manner as for MBBs, we determine MBSs which, for a given object, are those smallest spheres which totally enclose the object (see fig. 9). Their storage requires only d+1 parameters in a d-dimensional space to store the radius and the d-dimensional center point.

$$LB_{MBS(A)}(x, y) = \sqrt{eigmin}\,|x - y|$$

has been shown to be the minimum bounding sphere distance function of the QF using A, where *eigmin* is the minimum eigenvalue of A.

It has been proven that there is no tighter hypersphere. This means that filters using this MBS definition are also complete and optimal.

Both the MBB and MBS approximation have specific characteristics with respect to their approximation quality and their potential of improving query processing efficiency. A combination of these filters exploits the advantages of both techniques. It can be shown that if two filter functions are both lower bounds of the actual distance function, their maximum is a lower bound as well. For two

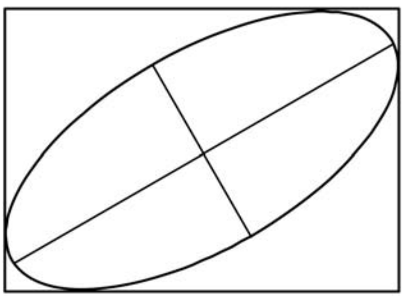
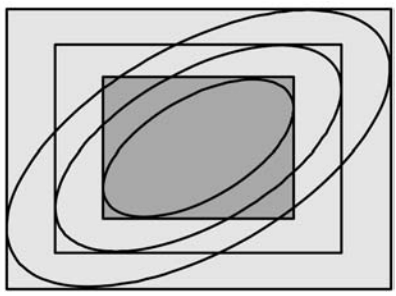


Fig. 8: Minimum bounding rectangle



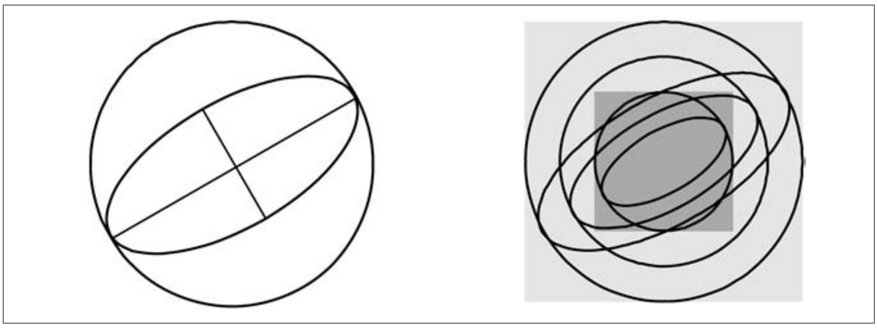


Fig. 9: Minimum bounding sphere

lower bounds, taking their maximum will improve their selectivity, i.e. the amount of objects which can be pruned from similarity search.

5.3 Experiments for the QF Filters

On an image database containing ~12,000 color pictures from commercially available CD-ROMs, the efficiency of dimensionality reduced multi-step query processing is evaluated. Depicted in figure 10 are *k*NN queries with $k = 12$ which represents 0.1% of the data. Acceptable runtimes are achieved for dimensions $r \geq 15$, and the overall minimum is reached at about 30. We observe that the overall runtime does not significantly vary for a wide range of index dimensionalities.

Other experiments run on a large image database, containing 8-D color histograms of 112,000 images as well as on a database of 1,000,000 objects that are uniformly distributed in the 8-D. Figure 11 demonstrates the improvement achieved for *k*-nearest neighbor queries for a varying value of *k*. For the image database, the performance gain was approximately 40% for the MBS approximation, and for the uniform distribution 35% to 40%.

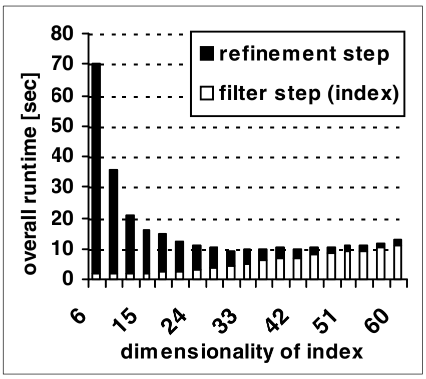


Fig. 10: Runtime of reduced QF [Seidl & Kriegel 1997]

6 Complete filters for the EMD

Multistep query processing has been successfully adapted to metric Earth Mover's Distances comparing equal mass histograms. Metric distances are very natural in that they require *symmetry*, i.e. the distance of an object *x* to another object *y* is identical to the distance of *y* to *x*; *definiteness*, i.e. only identical features have zero distance; and *triangle inequality*, i.e. the distance between two objects is never greater than the distance of a detour via some other object. For equal mass histograms and metric cost matrices, the Earth Mover's Distance is a metric as well.

A simple feature space average e.g. in the color space can serve as a complete filter [Rubner et al. 1997]. Let r_i denote the feature representatives (centroids) and *m* the total mass in each histogram, then the 3D-color averages, a weighted sum of the feature representatives form a simple lower bound for the EMD:

$$LB_{Avg}(x,y) = \left\| \sum_i \frac{x_i r_i}{m} - \sum_i \frac{y_i r_i}{m} \right\|$$

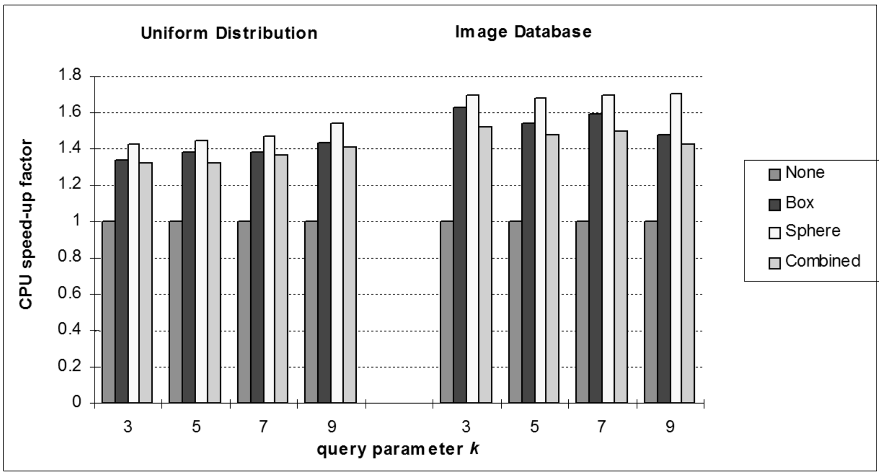


Fig. 11: Geometric approximations [Ankerst et al. 1998]

This filter improves computation times in smaller settings, but the efficiency gains are not sufficient for high-dimensional settings as the dimensionality of the filter is fixed to that of the color space.

A geometrically motivated approach to develop complete filter distances is to use diamonds, rectangles and spheres represented by Manhattan distance (L_1), weighted maximum norms (L_∞), and weighted Euclidean norms (L_2), respectively. As mentioned, L_p norms are well-suited in terms of ICES-multistep query processing, since they can be processed in linear time w.r.t. dimensionality per object (»efficiency«). They are immediately index enabled (»index«). Completeness has to be guaranteed by design and selectivity is shown empirically in our experiments.

6.1 L_p-based complete EMD filters

As discussed, (weighted) L_p norms base their distance calculation on the difference between individual feature values. The weighted L_1 (Manhattan) distance is defined by $\sum w_i |x_i - y_i|$ where the w_i represent weights for the individual dimensions. These weights are determined such that a complete filter is obtained:

$$LB_{Man}(x,y) = \sum_i \min_{j \neq i} \left\{ \frac{c_{ij}}{2m} \right\} |x_i - y_i|$$

For the minimum bounding rectangle approximation, the corresponding filter distance function is a weighted maximum norm (L_∞):

$$LB_{Max}(x,y) = \max_i \left\{ \min_{j \neq i} \left\{ \frac{c_{ij}}{m} \right\} |x_i - y_i| \right\}$$

The weighted Euclidean Distance corresponds to stretched (hyper-)spheres. A lower bound is defined as:

$$LB_{Euc}(x,y) = \sqrt{\sum_i \min_{j \neq i}\left\{\frac{c_{ij}}{2m}\right\}(x_i - y_i)^2}$$

For all these filters, we prove the lower-bounding property in [Assent et al. 2006], thus guaranteeing completeness in KNOP multistep query processing. Moreover, we show that $LB_{Euc} \leq LB_{Man}$ which means the Euclidean-based filter is of worse selectivity than the Manhattan-based one.

While lower bounds based on L_p norms are well-suited for index support and can be efficiently computed due to their dimension-wise computability, their selectivity decreases for higher dimensions due to the »curse of dimensionality«. When the number of features (and therefore the number of cost matrix entries) increases, the minimum of those features decreases as well. Therefore, in high-dimensional settings, the selectivity of bounds based on these minima drops.

Moreover, index structures fall prey to the curse of dimensionality as well. Refining the idea derived from the geometrically inspired weighted L_p lower bounds, we have developed an even tighter lower bound for higher-dimensional settings.

6.2 Independent minimization lower bound

Approximating the actual cost by its row- or column-wise minimum can be further refined. Instead of picking one minimum for each histogram feature, re-approach the EMD by taking into consideration that histogram features can only absorb the mass corresponding to their respective entries. That is, for each dimension, we reinstall the constraint that each histogram feature should not receive any excess mass. The difference to the EMD is that the sum of flows for different dimensions might well be above this limit.

The Independent Minimization lower bound (LBIM) is defined as:

$$LB_{IM}(x,y) = \min_{F \in AF}\{\sum_i \sum_j f_{ij} c_{ij} / \sum_{i,j} f_{ij}\},$$

with AF the set of all admissible flows $F=[f_{ij}]$ w.r.t.

a. no more earth is moved from any one feature component of x than available in that feature component,
b. no more earth is moved from any one feature component to any other feature component of y than allowed by that feature component,
c. only non-negative amounts of earth are moved and
d. the entire earth is moved from x to y (as the histograms have equal total mass).

The difference between the new LB_{IM} and the EMD is the constraint on how much mass one feature may receive (b). While the EMD requires that the sum of flows to a certain feature equals its mass over all dimensions, the LB_{IM} only ensures that for any single dimension the incoming flows do not exceed its mass.

It is shown in [Assent et al. 2006] that LB_{IM} lower bounds the EMD, making it a complete filter for KNOP query processing. The new constraints for the LB_{IM} are a weaker version of those set for the EMD. Thus, these constraints describe a superset of the set described by the EMD constraints. As the minimum of the set is contained in its superset, the minimum of the superset is always smaller than or at most equal to that of the set itself. Thus LB_{IM} is indeed a complete filter for the EMD and can be used in ICES multistep.

This lower bound can be efficiently computed, as the constraints can now be checked individually for each dimension and linear programming complexity is avoided.

LB_{IM} is further improved by adopting two properties seen for L_p lower bounds:

- Remove the mass of corresponding features: metric cost matrices have zero diagonal entries (definiteness). As they do not contribute to the EMD value, removing the corresponding mass does not violate the lower bounding property. Since the resulting reduced histograms have fewer ›options‹ for moving mass, the filter is more selective.
- Instead of considering row-wise independent minimization, the same can be done column-wise. As both are valid lower bounds, take the combination (maximum) of LB_{IM}(x,y) and LB_{IM}(y,x) to increase the selectivity.

6.3 Multistep filter concept

L_p-norm-based filters are well suited for index support and have low one-to-one computation times. As they do not provide optimal selectivity for increasing dimensionality, they are best used for low-dimensional problems. As dimensionality increases, LB_{IM} lower bound is favorable as it yields a better selectivity which eventually means that overall computation times are lower in high dimensionality where many more candidates might otherwise be generated.

Since a combination of filters further reduces the computation time, we propose to integrate a reduction of dimensionality into a two-phase multistep algorithm which combines the advantages of low-dimensional index-support with the good selectivity of our multi-dimensional LB_{IM} filter. As depicted in figure 12, we construct three-dimensional indexes based on the averaging lower bound or on the weighted Manhattan lower bound, which, as experiments confirm, is best among the L_p-based filters.

6.4 Experiments for the EMD Filters

The proposed approach was evaluated on a large color image database of 200,000 images for kNN queries with 64-dimensional histograms in HLS color space. Using KNOP query processing, we queried for 10 nearest neighbors, and varied histogram sizes from 16 to 64 dimensions.

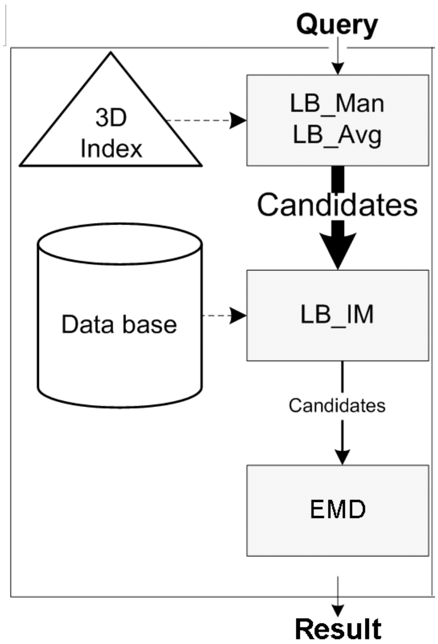


Fig. 12: Multistep concept

We evaluated the performance by measuring the selectivity for database sizes from 25,000 images to 200,000 images. The selectivity is the average percentage of database images for which an EMD computation is necessary.

In the left diagram of figure 13 we can see that LB_{Max} produces many more candidates than LB_{Man} or LB_{Avg}; its selectivity is inferior by an order of magnitude. This means that the EMD is more closely bound by the hyperdiamond which corresponds to LB_{Man} than by the hyperrectangle belonging to LB_{Max}. LB_{IM} is of noticeably higher selectivity. It outputs far less than 0.1% of the data as candidates for all database sizes. This is an improvement by more than two orders of magnitude to the second-best. In the right part of figure 13 we see that for the same setup, the response times of LB_{Man} and LB_{Avg} are closely related to their selectivity as their computational overhead is low. LB_{IM} requires more computational effort, thus despite its far superior selectivity, it shows only similar response times. However, the combination with the former two lower bounds yields the lowest response times.

We varied histogram sizes from dimension 16 to 64. We can see in the left diagram of figure 14 that LB_{IM} yields clearly best selectivity results in all cases. The right diagram (fig. 14) shows that with increasing dimensionality, the computation of LB_{Avg} increases in complexity. The response times of LB_{Man} are more closely related to its selectivity ratios. The combination as presented in the multistep filter concept yields the best performance improvements. We include the sequential scan EMD computation times as a baseline comparison. Note that the improvement for 64 dimensions comparing EMD and the best multistep concept is from 1000 seconds to less than one second, i.e. more than three orders of magnitude.

7 Conclusion

Adaptable similarity models perform well in terms of assessing human perceived similarity tailored to specific applications. Their high complexity hinders their application in high-dimensional or large databases. ICES-filters used in multistep query processing have been shown to speed up these adaptable similarity models for those settings.

Acknowledgment

This research is partially funded by project SE 1039/1-2 of the German Research Foundation (DFG).

References

[Ankerst et al. 1998] *Ankerst, M.; Braunmüller, B.; Kriegel, H.-P.; Seidl, T.:* Improving adaptable similarity query processing by using approximations. In: Proc. VLDB 1998, S. 206-217.

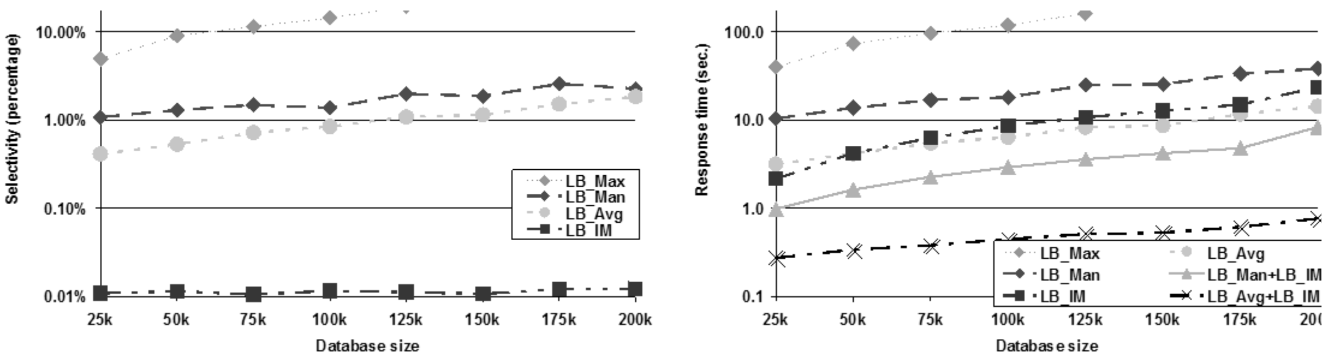
[Ankerst et al. 1999] *Ankerst M.; Kastenmüller G.; Kriegel H.-P.; Seidl T.:* Nearest Neighbor Classification in 3D Protein Databases. In: ISMB 1999.

[Assent et al. 2006] *Assent, I.; Wenning, A.; Seidl,T.:* Approximation techniques for indexing the Earth Mover´s Distance in multimedia databases. In: Proc. ICDE 2006.

[Berchtold et al. 1996] *Berchtold, S.; Keim, D.; Kriegel, H.-P.:* The X-tree: An index structure for high-dimensional data. In: Proc. VLDB 1996, S. 28-39.

[Böhm et al. 2001] *Böhm, C.; Berchtold, S.; Keim, D.:* Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. In: ACM Computing Surveys 33(3), 2001, S. 322-373.

[Demirci et al. 2004] *Demirci, M.; Shokoufandeh, A.; Dickinson, S.; Keselman, Y.; Bretzner, L.:* Many-to-many feature matching using spherical coding of directed graphs. In: Proc. ECCV 2004.



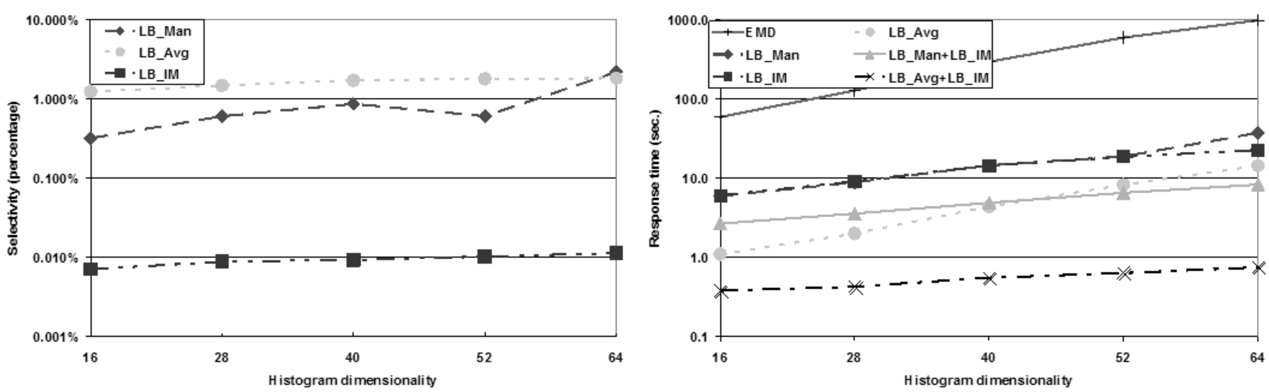


Fig. 14: EMD multistep: histogram sizes

[Faloutsos 1996] *Faloutsos, C.:* Searching Multimedia Databases by Content. Kluwer Academic Publishers, 1996.

[Grauman & Darrell 2004] *Grauman, K.; Darrell, T.:* Fast contour matching using approximate earth movers distance. In: Proc. IEEE CVPR 2004.

[Guttman 1984] *Guttman, A.:* R-trees: A dynamic index structure for spatial searching. In: Proc. ACM SIGMOD 1984, S. 47-57.

[Hafner et al. 1995] *Hafner, J.; Sawhney, H.; Equitz, W.; Flickner, M.; Niblack, W.:* Efficient color histogram indexing for quadratic form distance functions. In: IEEE TPAMI 17(7), 1995, S. 729-736.

[Hillier & Lieberman 1990] *Hillier, F.; Lieberman, G.:* Introduction to Linear Programming. McGraw-Hill, 1990.

[Ishikawa et al. 1998] *Ishikawa Y.; Subramanya R.; Faloutsos C.:* MindReader: Querying Databases Through Multiple Examples. In: Proc. VLDB 1998, S. 218 - 227.

[Jing et al. 2004] *Jing, F.; Li, M.; Zhang, H.; Zhang, B.:* An Efficient and Effective Region-Based Image Retrieval Framework. In: IEEE TPAMI 13(5), 2004, S. 699-709.

[Lavin et al. 1998] *Lavin, Y.; Batra, R.; Hesselink, L.:* Feature comparisons of vector fields using earth mover's distance. In: Proc. VIS 1998, S. 103-109.

[Lazebnik et al. 2003] *Lazebnik, S.; Schmid, C.; Ponce, J.:* Sparse texture representation using affine-invariant neighborhoods. In: Proc. IEEE CVPR 2003.

[Rubner & Tomasi 2001] *Rubner, Y.; Tomasi, C.:* Perceptual Metrics for Image Database Navigation. Kluwer Academic Publishers, 2001.

[Rubner et al. 1997] *Rubner, Y.; Guibas, L.; Tomasi, C.:* The Earth Mover's Distance, Multi-Dimensional Scaling, and color based image retrieval. In: Proc. ARPA IUW 1997, S. 661-668.

[Schmitt 2005] *Schmitt, I.:* Ähnlichkeitssuche in Multimedia-Datenbanken – Retrieval, Suchalgorithmen und Anfragebehandlung. Oldenbourg, 2005.

[Seidl & Kriegel 1997] *Seidl, T.; Kriegel, H.-P.:* Efficient user-adaptable similarity search in large multimedia databases. In: Proc. VLDB 1997, S. 506-515.

[Seidl & Kriegel 1998] *Seidl, T.; Kriegel, H.-P.:* Optimal multi-step k-nearest neighbor search. In: Proc. ACM SIGMOD 1998, S. 154-165.

[Smeulders et al. 2000] *Smeulders, A.; Worring, M.; Santini, S.; Gupta, A.; Jain R.:* Content-Based Image Retrieval at the End of the Early Years. In: IEEE TPAMI 22(12), 2000, S. 1349-1380.

[Smith 1997] *Smith, J.:* Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression, PhD thesis, Columbia University, NY, 1997.

[Typke et al. 2004] *Typke, R.; Veltkamp, R.; Wiering, F.:* Searching notated polyphonic music using transportation distances. In: Proc. ACM MM 2004, S. 128-135.

Ira Assent is a phd student and scientific employee at the Chair of Computer Science 9 at RWTH Aachen University since 2002, where she works on speeding up the EMD, finding subspace clusters and analysing time series.

Marc Wichterich is a phd student and DFG-funded scientific employee at the Chair of Computer Science 9 at RWTH Aachen University. He joined the data management and exploration group in October 2005 to work on speeding up the EMD.

Thomas Seidl is a full professor at the RWTH Aachen University where he heads the Chair of Computer Science 9 focused on data management and exploration. His research interests include data mining and data management in multimedia databases, spatial databases, computational biology, medical imaging and others.

Dipl.-Inform. Ira Assent
Dipl.-Inform. Marc Wichterich
Univ.-Prof. Dr. Thomas Seidl
RWTH Aachen
Lehrstuhl für Informatik 9
Ahornstr. 55
52054 Aachen
{assent, wichterich, seidl}
@informatik.rwth-aachen.de
http://www-i9.informatik.rwth-aachen.de