Efficient multi-step query processing for EMD-based similarity

Ira Assent, Thomas Seidl

Data Management and Exploration Group, RWTH Aachen
52056 Aachen, Germany
{assent,seidl}@cs.rwth-aachen.de

Abstract. Similarity search in large multimedia databases requires efficient query processing based on suitable similarity models. Similarity models consist of a feature extraction step as well as a distance defined for these features, and they demand an efficient algorithm for retrieving similar objects under this model. In this work, we focus on the Earth Movers Distance (EMD), a recently introduced similarity model which has been successfully employed in numerous applications and has been reported as well reflecting human perceptual similarity. As its computation is complex, the direct application of the EMD to large, high-dimensional databases is not feasible. To remedy this and allow users to benefit from the high quality of the model even in larger settings, we developed various lower bounds for the EMD to be used in index-supported multistep query processing algorithms. We prove that our algorithms are complete, thus producing no false drops. We also show that it is highly efficient as experiments on large image databases with high-dimensional features demonstrate.

1 Introduction

In content-based retrieval, multimedia objects are compared using similarity models. These include a feature extraction scheme, a (dis-)similarity model between features and efficient query processing algorithms. A common model for feature extraction is recording the feature distribution of objects in histograms, e.g. color or texture histograms for images or shape histograms for 3D objects [1,2].

For (dis-)similarity measures, a simple approach are L_p norms, such as Manhattan norm, Euclidean norm or Maxnorm, distances. The basic idea is to assess the distance between histograms by summarizing the differences in individual bins. L_p norms are defined for two histograms $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, where x_i and y_i denote the bin entries, as follows: $L_p(x,y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p}$.
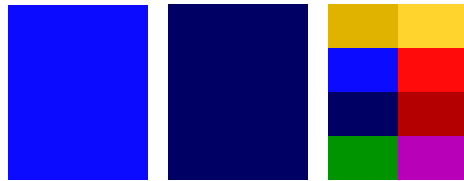
Fig. 1. Example color images

To understand the implications of this model, consider the three images in Figure 1. The two left ones are similar in terms of color distribution with only a minor shift in color tone, whereas the right one has a completely different color structure.
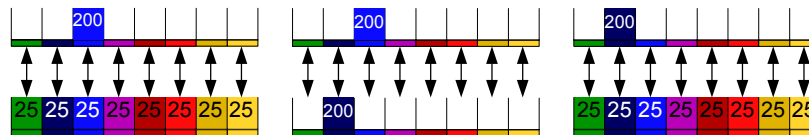

Fig. 2. Computation of L_p norms

The corresponding histograms are illustrated above: depending on the desired resolution, histogram bins representing parts of the color space are created. To obtain an image's histogram, the number of pixels belonging to a certain bin are added up. The L_p norm values between the respective images are then obtained by summing up the differences in each bin. Doing so, comparing e.g. via Manhattan norm (summing up the absolute differences) the leftmost one with the right image, we obtain a value of $25 + 25 + 175 + 25 + 25 + 25 + 25 + 25 = 350$, and the two images on the left yield $0 + 200 + 200 + 0 + 0 + 0 + 0 + 0 = 400$. As we can see, the result is counterintuitive since L_p norms ignore the relationship between neighboring histogram bins.

To remedy this, adaptable similarity models like Quadratic Forms have been introduced. Quadratic Forms use a similarity matrix to encode the neighborhood relationship between bins and correspondingly weights the bin differences. Given a similarity matrix $A = [a_{ij}]$ for distances between bin i and bin j the Quadratic Form is defined as $QF_A(x, y) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - y_i) a_{ij} (x_j - y_j)}$. Thus, Quadratic Forms weight the differences between histogram bin entries by the entries in the similarity matrix. Quadratic Forms have been studied in [3,1].

Another model, recently introduced in Computer Vision, the Earth Mover's Distance (EMD) uses a cost matrix to direct a similarity match between histograms [4].

2   Adaptable similarity: The Earth Mover's Distance

The Earth Mover's Distance takes a slightly different approach towards assessing the (dis-)similarity between features. The basic idea is to find the best match between e.g. two color histograms by measuring the minimal effort for transforming one into the other. The intuition, which explains the name, is that one histogram's entries are earth piles, whereas the other's are holes of earth. The cost matrix contains the cost for moving earth from piles to holes. By determining the "cheapest" way of moving all earth from the piles to the holes, the distance between the two histograms is measured.

Formally, the Earth Mover's Distance between histograms x and y with respect to a cost matrix $C = [c_{ij}]$ is defined as follows:

$$EMD_C(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{c_{ij}}{m} f_{ij}$$

where f_{ij} ($f_{ij} \geq 0$ for all $1 \leq i, j \leq n$) is the minimum flow subject to

$$\sum_{j=1}^{n} f_{ij} = x_i, \quad \sum_{i=1}^{n} f_{ij} = y_j$$

where $m := \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}$ normalizes the EMD by the mass of the histograms. When defined this way, the EMD is metric as long as the ground distance is metric.

Reconsidering the exemplary images in Figure 1, the corresponding EMD distances best distributes the upper histogram's mass into the lower one's bin entries. As we can see, earth may be moved to neighboring bins, resulting in small cost values (assuming e.g. Manhattan cost matrix); whereas moving the earth to bins which are further away results in larger values, thus resulting in large distance values for structurally different color distributions. Consider a Manhattan cost matrix which means zero cost for identical bins, a cost of "1" for direct neighbors, a cost of "2" for next-to-direct neighbors, and so on. For the shift in color tone on the left in Figure 2, we thus have an EMD distance of $200 * 1 = 200$. Comparing one of them to the structurally different image yields an EMD distance of $1*25+2*25+0*25+1*25+2*25+3*25+4*25+3*25 = 400$.
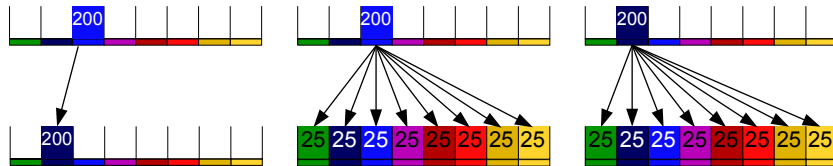

Fig. 3. EMD computation

The Earth Mover's Distance has been thoroughly evaluated and successfully employed in content-based image retrieval [4] and numerous other application areas. Examples include graph matching, where a low-distortion embedding is used to transform graph matching to geometric point matching in vector spaces and the EMD is used for dissimilarity assessment [5]. A low-distortion embedding of the Earth Mover's Distance is utilized for contour matching [6]. In physics, vector fields may be described using critical points and EMD measures their dissimilarity [7]. Another example from color-based image retrieval evaluates region-based similarity via EMD and employs a relevance feedback scheme to improve result quality [8]. Region-based similarity focusing on texture is studied in another EMD application approach [9]. Music can be described in terms of time and pitch as well as note durations which may be compared via EMD or a modified version thereof, the pseudometric PTD (proportional transportation distance) [10].

The Earth Mover's Distance can be computed using a Linear Programming scheme, following a Transportation Problem [11]. While this is a feasible method for small, low-dimensional applications, large multimedia databases cannot benefit. To overcome this limitation, we propose a multi-step query processing algorithm, using novel filter distances, which guarantees exactly the same result, yet at far faster response times. These perfect recall/precision results stem from a lower-bounding property in a GEMINI or KNOP multistep query processing algorithm (see Section 4), which is proven in [12]. The effectiveness and efficiency of the approach is validated in experiments on large color image databases.

3   Speeding up response times: indexing structures

Index structures are used to organize the data with the objective that only a small percentage needs to be accessed during query processing. In a multi-dimensional space, index structures such as R-trees or X-trees [13,14] can be used.
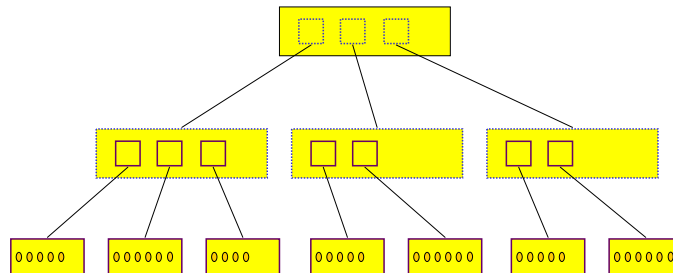

Fig. 4. R-Tree structure

The basic idea is illustrated in Figure 3: data is organized hierarchically, grouping objects into minimum bounding rectangles (Fig. 3, whose description is stored in the tree's nodes. During query processing, the query is compared with these minimum bounding rectangles in a top-down fashion. Whenever the distance between query and minimum bounding rectangle exceeds the given limit, the corresponding subtree can be safely pruned from search. Reaching the lowest level, the leafs contain the actual objects.

Typical query types on these index structures are range queries, where a maximum distance threshold is given by the user along with the query object. All those objects are returned which are within at most this maximum distance from the query. Another query type is the k nearest neighbor (kNN) query, where the k most similar objects to the query are returned. While this is convenient for users who do not need to know typical distance values in the data base and can simply adjust the size of the result set, query processing is slightly more complex in this case for both indexing structures as well as multistep query processing algorithms. For range queries, all those regions indexed which are farther away than the maximum distance can be safely pruned. For kNN queries, this maximum distance is not known in advance and needs to be constantly updated as more similar objects are found during search.
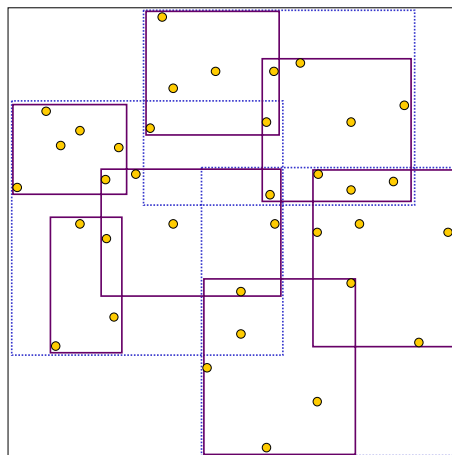

Fig. 5. Geometric visualization of R-Tree index

4   Fast query processing: multi-step algorithms

Index usage can be even more accelerated by multistep retrieval algorithms as illustrated in Figure 6. A query is first executed as an approximative query using a suitable index structure. This filter step, based on an approximate filter

distance function, generates a set of candidates which is then evaluated using the real distance function to retrieve the desired result from the database.
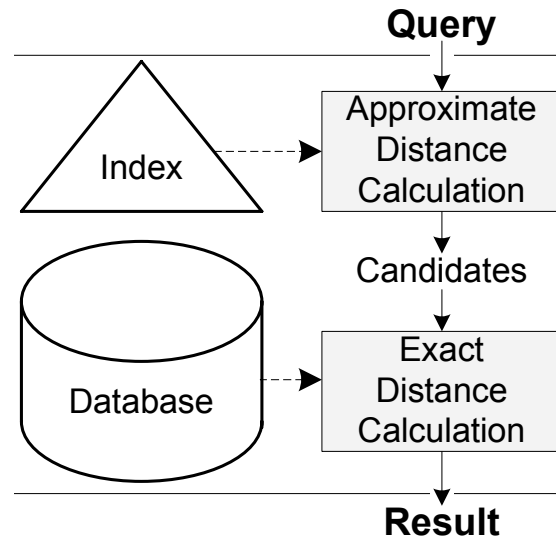

Fig. 6. Multistep Retrieval

A first multistep algorithm, GEMINI was presented in [15]. It was further optimized in terms of number of pages which have to be accessed in [16]. Crucial for the efficiency of these multistep algorithms is the quality of the filter distance measure. This measure should meet a number of criteria (ICES):

Fig. 7. Filter criteria

Index:	as mentioned above, index structures shorten response times.
Completeness	means no false drops.
	For GEMINI, and KNOP, completeness can be ensured by proving
	that the filter distance lower bounds the exact object distance
	for any two objects in the data base.
Efficiency	of single computations of the filter is crucial for the overall
	response time of the multistep algorithm.
Selectivity	means that the filter should discard as many objects as possible
	without producing false drops (completeness).

5 Reducing costly computations: lower bounds for the EMD

A simple lower bound for smaller, low-dimensional feature spaces can be obtained by averaging histograms in the features space, i.e. for color histograms by computing a 3D color average [17]: $EMD(x,y) \geq \| \sum_{i=0}^{n} \frac{x_i r_i}{m} - \sum_{i=0}^{n} \frac{y_i r_i}{m} \|$, where the r_i denote the bin representatives. As the efficiency gains are not sufficient for high-dimensional and large multimedia databases, a geometrical approach for developing novel filters is followed.
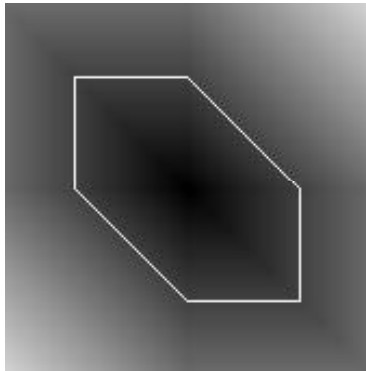

Fig. 8. Iso-lines of the EMD

Figure 8 highlights an exemplary 2D projection of an EMD iso-contour, i.e. points which have the same EMD-distance value from the center point. A good lower bounding filter approximation should describe a closely surrounding geometry.

The distance functions for diamonds, rectangles and spheres are represented by weighted Manhattan (L_1), weighted maximum (L_∞) and weighted Euclidean norms (L_2), respectively (Fig. 5). The weights stretch and compress these geometries. They have to be determined as parameters to optimally enclose the EMD
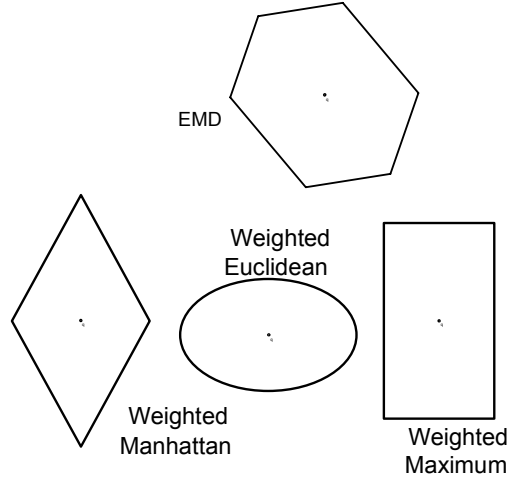
Fig. 9. Iso-contours of weighted L_p norms and EMD

as a lower bound in multi step query processing (Fig. 5). L_p filter distances are computed in linear time and can be supported by any available multi-dimensional indexing structure.
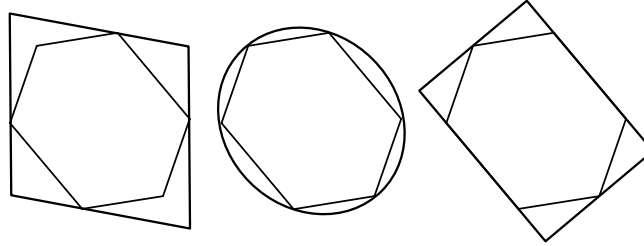

Fig. 10. EMD and surrounding geometries (3D projections)

We define these L_p filters as follows (for details refer to [12]):

$$LB_{Manh}(x,y) := \sum_{i=1}^{n} w_i |x_i - y_i|, \ w_i = min_{j,i\neq j}\{\tfrac{c_{ij}}{2\cdot m}\} \ \leq \ EMD(x,y)$$

$$LB_{Max}(x,y) := \max_{i}\{\min_{\substack{j \\ i\neq j}}\{\tfrac{c_{ij}}{m}\}|x_i - y_i|\} \ \leq \ EMD(x,y)$$

$$LB_{Eucl}(x,y) := \sqrt{\sum_{i=1}^{n}(\min_{\substack{j \\ i\neq j}}\{\tfrac{c_{ij}}{2\cdot m}\})^2(x_i - y_i)^2} \ \leq \ EMD(x,y)$$

In high-dimensional settings, selectivity of these filters is bound to drop as the minimum over more and more values decreases. Moreover, indexing structures also lose their pruning power. Thus, more selective complex filters are needed.
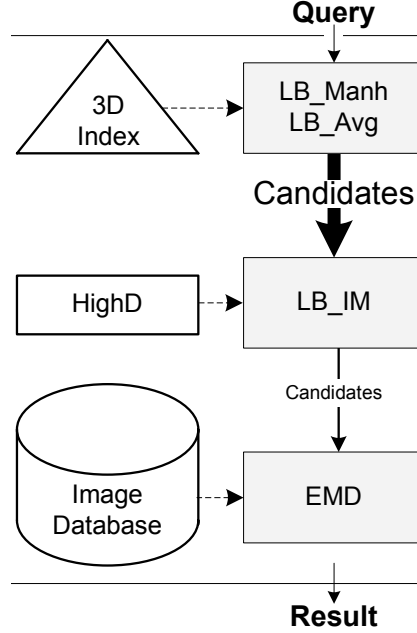

Fig. 11. EMD Multistep Concept

By refining the techniques used for the L_p norm based lower bounds, a repeated computing of minimal cost entries leads to the Independent Minimization lower bound:

$$LB_{IM}(x,y) := \min_{i,j} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{c_{ij}}{m} f_{ij}, f_{ij} \geq 0, \sum_{j=1}^{n} f_{ij} = x_i, f_{ij} \leq y_j \right\} \leq EMD(x,y)$$

Thus, the difference between the LB_{IM} and the EMD is the constraint on how much mass one histogram bin may receive. While the EMD requires that the sum of flows to a certain bin equals its mass over all dimensions, the LB_{IM} only ensures that for any single dimension the incoming flows do not exceed its mass. Further improvements of the Independent Minimization lower bound as well as completeness proofs for all of the above filters can be found in [12].

5.1 Adapted multistep concept for combined filters

The combination of the above filters further reduces the computation time as they dismiss different sets of objects as non-candidates. A sequential scan on

a good-selectivity-filter in high dimensions on the output of an efficient low-dimensional, index-supported filter makes best use of database technology available. We therefore construct three-dimensional indexes based on the averaging lower bound or on dimensionality reduced 3D weighted Manhattan lower bound (determining those three dimensions with highest variability and discarding the remainder).

6   Experiments

Experiments on 200,000 color image histograms using KNOP query processing were evaluated in terms of selectivity ratios as well as total response times for query processing. The selectivity is the average percentage of database images for which EMD computation is necessary. Note that this approximately reflects the number of false positives. As we have proven completeness, there are no false negatives.
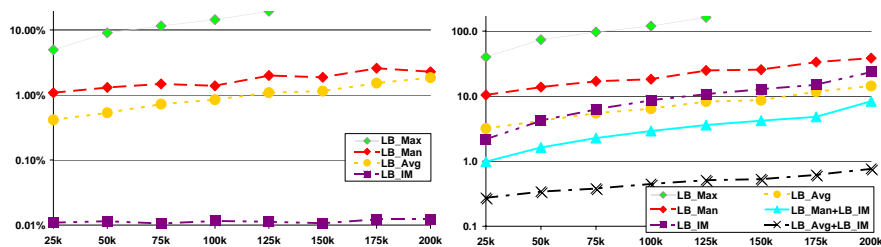

Fig. 12. Scalability: selectivity perc. (left), response time in sec. (right)

We measured the selectivity for database sizes ranging from 25.000 to 200,000 images for 10-nearest-neighbor queries on 64d histograms. In the left diagram in Figure 12 we can see that LB_{Max} produces many more candidates than LB_{Avg}; its selectivity is inferior by an order of magnitude. LB_{IM} is noticeably more selective. 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 lower bound. In the right part of this figure, we see that the response times of LB_{Man} and LB_{Avg} are closely related to their selectivity. LB_{IM} requires more computational effort, thus it shows only similar response times. Their combination, as presented in Section 5.1, yields fastest results.

We also varied histogram sizes from 16 to 64. We can see in the left diagram in Figure 13, where the largest database with 200,000 was queried for $k = 10$ nearest neighbors, that LB_{IM} has the best selectivity. For finer histogram resolutions, by more than two orders of magnitude. In the right part of the figure, we can see 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. As in the previous experiment, the overhead of LB_{IM} is greater

than that of the other two lower bounds. Once again, the proposed combination 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.
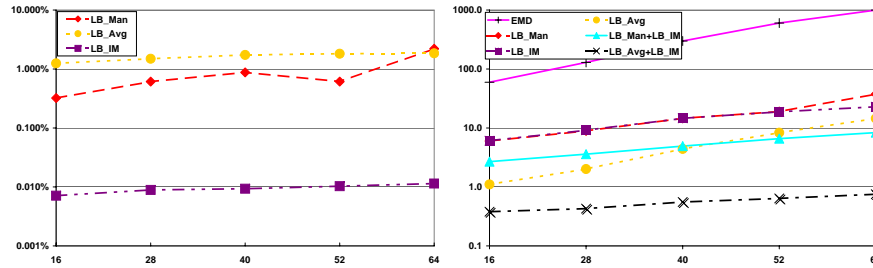

Fig. 13. Dimensionality: selectivity perc. (left), response time in sec. (right)

7 Conclusion

In summary, our experiments demonstrate that the best strategy for query processing is a combination of assets. By building a small three-dimensional index based on simpler filter functions and combining it with a highly selective LB_{IM} filter, index support can be profited from while expensive EMD computations are minimized. Users benefit from noticeably smaller response times while losing no actual result.

References

1. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D. and Equitz, W.: Efficient and effective querying by image content. Journal of Intelligent Information Systems **3** (1994) 231–262
2. Ankerst, M., Kastenmüller, G., Kriegel, H.-P. and Seidl, T.: Nearest neighbor classification in 3d protein databases. In: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB), Lecture Notes in Computer Science, Springer (1999) 34–43
3. Ankerst, M., Braunmüller, B., Kriegel, H.-P. and Seidl, T.: Improving adaptable similarity query processing by using approximations. In: Proceedings of the 24th International Conference on Very Large Data Bases, New York, USA. (1998) 206–217
4. Rubner, Y. and Tomasi, C.: Perceptual Metrics for Image Database Navigation. Kluwer Academic Publishers (2001)

5. Demirci, M., Shokoufandeh, A., Dickinson, S., Keselman, Y. and Bretzner, L.: Many-to-many feature matching using spherical coding of directed graphs. In: Computer Vision - ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, Lecture Notes in Computer Science, Springer (2004)

6. Grauman, K. and Darrell, T.: Fast contour matching using approximate earth movers distance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2004)

7. Lavin, Y., Batra, R. and Hesselink, L.: Feature comparisons of vector fields using earth mover's distance. In: Proceedings of the conference on Visualization. (1998) 103–109

8. Jing, F., Li, M., Zhang, H., Zhang, B.: An Efficient and Effective Region-Based Image Retrieval Framework. IEEE Transactions on Image Processing **13** (2004) 699–709

9. Lazebnik, S., Schmid, C. and Ponce, J.: Sparse texture representation using affine-invariant neighborhoods. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2003)

10. Typke, R., Giannopoulos, P., Veltkamp, R., Wiering, F. and Oostrum, R. van: Using transportation distances for measuring melodic similarity. In: Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR). (2003)

11. Hillier, F. and Lieberman, G.: Introduction to Linear Programming. McGraw-Hill (1990)

12. Assent, I., Wenning, A. and Seidl, T.: Approximation techniques for indexing the earth mover's distance in multimedia databases. In: Proceedings of the International Conference on Data Engineering (ICDE). (2006)

13. Berchtold, S., Keim, D. and Kriegel, H.-P.: The x-tree : An index structure for high-dimensional data. In: Proceedings of 22th International Conference on Very Large Data Bases, Mumbai (Bombay), India. (1996) 28–39

14. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proceedings of the 1994 ACM SIGMOD international conference on Management of data, Boston, Massachusetts. (1984) 47–57

15. Faloutsos, C.: Searching Multimedia Databases by Content. Kluwer Academic Publishers (1996)

16. Seidl, T. and Kriegel, H.-P.: Optimal multi-step k-nearest neighbor search. In: Proceedings of the 1998 ACM SIGMOD international conference on Management of data. (1998) 154–165

17. Rubner, Y., Tomasi, C., and Guibas, L.: A metric for distributions with applications to image databases. In: Proceedings of the IEEE International Conference on Computer Vision. (1998) 59–66