Efficient Indexing, Color Descriptors and Browsing in Image Databases

Nikos Batalas, Christos Diou* and Anastasios Delopoulos
Multimedia Understanding Group,
Department of Electrical and Computer Engineering,
Aristotle University of Thessaloniki - Greece
E-mail: amv@ee.auth.gr, diou@mug.ee.auth.gr, adelo@eng.auth.gr

Abstract

This work provides an experimental evaluation of various existing approaches for some of the major problems content based image retrieval applications are faced with. More specifically, global color representation, indexing and navigation methods are analyzed and insight is provided regarding their efficiency and applicability. Furthermore this paper proposes and evaluates the combined use of FastMap and kd-trees to enable accurate and fast retrieval in image databases.

1. Introduction

The main goal of content based image retrieval research is to devise suitable representations of images in order to allow query and retrieval based on the visual properties of images and not user annotations. Often the queries themselves are images and the user expects similar images to be retrieved.

Significant research has been performed on image retrieval systems in the past few years and the promising results contributed to the development of the MPEG-7 standard ([7, 11]). The ultimate goal of automatic semantic characterization of images based on their visual content remains largely unsolved (even though there are partially successful approaches under controlled environments e.g., [2, 6]). Still, the simple descriptions of images based on color, textures, shapes etc. provide adequate results for a user to begin a search (the authors believe that navigation and inspection will always be a part of any retrieval system).

Apart from efficient image representation, however, a practical image retrieval application has additional requirements. Queries must be answered fast, hence appropriate indexing mechanisms must be employed as well. The problem is that often images are represented as points in a high dimensional (> 10 dimensions) space and searching for similar images becomes a problem of finding the nearest neighbors of the given query point in that space. This a problem of very high computational cost for a large number of database entries and the major obstacle that prevents the development of large scale image databases. Furthermore, often descriptions of images are not points in a space, preventing the use of indexing data structures provided by theoretical computer science research. As a final point, user interfaces should assist navigation through the database contents by grouping similar images, a problem that adds additional difficulties in the already complex image retrieval system.

This paper provides the experimental evaluation of known methods to the following problems: Global color representation (Section 2), indexing (Section 3) and browsing (Section 4). Additionally, it proposes a different approach to controlling the dimensionality of the image representation via the FastMap algorithm (Section 5). The experimental results show that a lower number of dimensions can be used (thus faster indexing) without significant loss in retrieval efficiency.

2. Global color representation

2.1. Histogram and dominant color

Global color descriptors are used to describe the color properties of an image independent of spatial color distribution. The most important descriptors of this form are the well known color histogram and the dominant color descriptors.

Both these image properties are described by

$$D = \{\{c_i, p_i\}, i = 1, \ldots, N\} \qquad (1)$$

where c_i is a color from a predefined colorspace and p_i is the percentage of image pixels having that color. Of course, having a 3-channel 8-bits per channel image described by

*Christos Diou is supported by the Greek State Scholarships Foundation

(1) is very impractical, since (i) $2N = 2^{25}$ values are used to describe a single image and (ii) this level of granularity is not informative for the purpose of image retrieval. Therefore, images are quantized prior to extraction of D.

In histograms, the given colorspace (e.g., RGB) is usually quantized to a predefined number of "bins" independent of the images. While this approach can reduce the number of values required for D (the colors c_i are essentially predefined for a given image format), the description does not adapt to each image. Consider, for example, a 24-bit RGB image with only 64 different colors all at the same color region. Then, if the colorspace is quantized at $N = 12$ levels (4 levels for each color channel), all these colors will be concentrated at a single bin; the rest of the 11 color-value pairs are left unused.

The dominant color descriptor, on the other hand, overcomes this issue by allowing the use of the more general form of (1) where the colors c_i and their number N can be different for each individual image. Naturally, a method for selecting the appropriate dominant colors c_i for each image must be defined. In [7] regarding the MPEG-7 standard color descriptor the use of the Generalized Lloyd Algorithm is proposed. In this work, a different approach utilizing octrees for color reduction [4] was used.

An issue with the dominant color approach is that the definition of an effective distance metric comparing descriptors is not as straightforward. The next section deals with the evaluation of two distance metrics used to compare the descriptors D.

2.2. Evaluation of distance metrics

Given a predefined set of colors c_i, $i = 1, \dots, N$, a $N \times N$ matrix A where each element a_{jk} of A is the distance between c_j and c_k in their colorspace and two vectors h_1 and h_2 with the percentages of each color c_i, the quadratic histogram distance given by

$$d_h(h_1, h_2) = (h_1 - h_2)^T A (h_1 - h_2) \tag{2}$$

However, Equation (2) cannot be used if the colors c_i and their number N are different for each image.

Deng *et al.* proposed a similar quadratic metric in [1], for the dominant color descriptor. If $D_1 = \{\{c_i, p_i\}, i = 1, \dots, N_1\}$ and $D_2 = \{\{b_j, q_j\}, j = 1, \dots, N_2\}$ are two dominant color descriptors, then the distance between D_1 and D_2 is defined to be

$$d_q(D_1, D_2) = \sum_{i=1}^{N_1} p_1^2 + \sum_{j=1}^{N_2} q_j^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{ij} p_i q_j \tag{3}$$

where the similarity coefficient a_{ij} is

$$a_{ij} = \begin{cases} 1 - d_{ij}/d_{\max}, & d_{ij} \le T_d \\ 0, & d_{ij} > T_d \end{cases} \tag{4}$$

$d_{ij} = \|c_i - b_j\|$ is the euclidian distance between c_i and b_j, $d_{\max} = \alpha T_d$, α is an arbitrary value and T_d is the maximum distance for two colors to be considered similar.

Another metric that has been proposed for comparing two dominant color descriptors D_1 and D_2 is the Earth Mover's Distance (EMD). In simple terms, the EMD is a dissimilarity measure between two images indicating the amount of "work" required to "move" from the descriptor D_1 of the first image to D_2 of the second. Imagine the colors c_i in the first descriptor as locations in a field with piles of p_i mass of earth each. The colors b_j are also locations, but they consist of holes with capacity q_j earth each. EMD denotes the minimum work required to distribute the piles of earth at c_i to the holes in b_j. Computation of the EMD is based on a solution of the transportation problem and is covered in [10].

A simple experiment was setup in order to evaluate the performance of each distance metric in a realistic image database scenario. All 5122 images in the corel dataset were indexed based on dominant color descriptors with 16 color/percentage pairs. The performance of each distance metric was evaluated by the first N_C results for each image, where N_C is the number of images in its category C. More specifically, for each image in every category of the dataset a query was performed and the top N_C results were retrieved. Then, the precision was measured based on the results retrieved from the same category as well as their position i.e.,

$$precision_C = \frac{\sum_{n=1}^{N} A_n}{\sum_{n=1}^{N} 1/n} \tag{5}$$

where $A_n = 1/n$ if the n'th result belongs in C and zero otherwise. For the interpretation of precision values, note that expecting to retrieve the best results at a semantic level (same category) using only a global color descriptor is way too optimistic. It can be used, however, to compare the two distance measures.

A graph of the results is given in Figure 1, where the horizontal axis corresponds to categories in the corel image dataset and the vertical axis is the precision (average value for all images in a category). The results are clearly in favour of the EMD, that consistently achieved higher precision compared to the quadratic distance.

3. Indexing

3.1. Image databases and dimensionality

By far the most important problem of image databases is the high dimensionality of the descriptors used, effectively leading to a prohibitive computational cost of indexing and

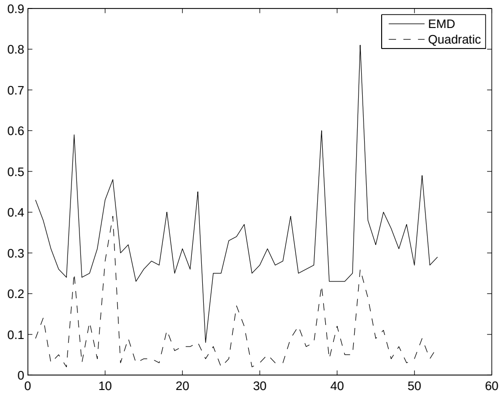


Figure 1. Results for retrieval based on dominant color descriptors with quadratic distance and EMD. Precision (Eq. (5)) vs. category.

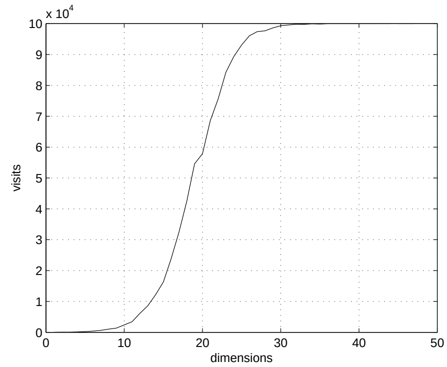
retrieval for a large number of entries. This, actually, is the reason why a content-based image retrieval system cannot be implemented for large scale datasets, such as the Internet.

In theoretical computer science the problem of "Nearest Neighbor" search in high dimensional spaces has been a very active research topic in the past few years. One of the most popular indexing data structures proposed is the kd-tree [8]. The idea is to construct a binary tree by successively using elements of the dataset as pivot points to partition the k-dimensional space into hyperrectangles, each containing at most one point. When searching, an initial estimate of the nearest neighbor is provided and then only hyperrectangles and pivot points that are possible to contain a point closer to the query than the initial estimate are visited. Thus, with kd-trees only a subset of the indexed points (i.e., database images) are visited, compared to the exhaustive search where the query is compared against all points in the dataset. The drawback is that the complexity for each point visit is increased (since branching conditions etc. have to be evaluated).

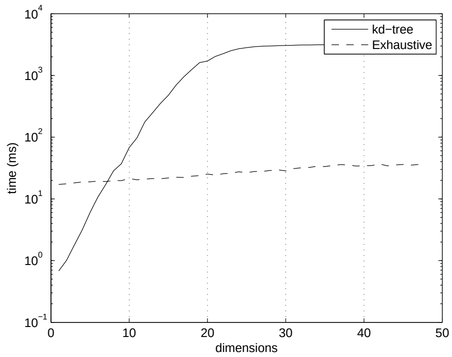
3.2. Limits of kd-tree effectiveness

In order to examine the behaviour of kd-trees w.r.t. the dimensionality of the space considered, the kd-tree data structure and associated algorithms were implemented and a dataset of 10^5 uniformly distributed random points was created for various dimensions. The number of nodes visited per dimension was measured and the results are given in Figure 2 (a).

The number of points visited for a number of dimensions close to 30 is practically the complete dataset and the use



(a)



(b)

Figure 2. (a) Number of visits for nearest neighbor search in the kd-tree for a dataset of 10^5 uniformly distributed random points. (b) Time for nearest neighbor search in ms for the kd-tree and exhaustive search for various dimensions on an average personal computer.

of kd-tree has no advantage over the exhaustive search. In fact, the upper limit of dimensions that the kd-tree is useful is lower, since each visit has additional costs in terms of CPU time. Figure 2 (b) provides the time (in ms) required per dimension for indexing performed on the same dataset using kd-trees and exhaustive search on an average personal computer.

These results indicate that for the test computer the kd-tree keeps an advantage in terms of computational time for 8 dimensions or less. Furthermore, through the experiments conducted, it was observed that the efficiency of the kd-tree search is largely dependent on the size of the dataset to be searched. Larger datasets allow for the kd-tree to be more efficient in even higher dimensions, compared to exhaustive searching. Also, note that a uniform dataset is the worst case scenario; kd-tree searches are significately faster within distinctively clustered datasets.

4. Browsing

Users often wish to browse the images in a database or the results retrieved from a query. An effective browsing interface is one that groups similar images together making the task of browsing easier for the user. This requirement can be translated to the following problem statement: “Given the observed distances d_{ij} between any two objects (images in a database), produce a configuration of points in the n-dimensional space, such that the new distances d'_{ij} are as close as possible to the original d_{ij} for all the points”. An popular measure of effectiveness for a solution is Kruskal’s *stress* function (6) [5].

$$\text{stress} = \left[\frac{\sum_{i,j} (d'_{ij} - d_{ij})^2}{\sum_{i,j} d_{ij}^2} \right]^{1/2} \qquad (6)$$

4.1. MDS and FastMap

Two approaches were evaluated for the mapping problem, namely metric Multidimensional Scaling (MDS) [9] and the FastMap algorithm [3]. Metric MDS is a technique that receives as input the observed dissimilarities d_{ij} between points and produces a configuration P' of points in the n-dimensional space through an iterative optimization process. For visualization and navigation purposes, $n = 2$ was used. It is interesting to note that contrary to embedding approaches such as PCA, the original points are not known or may not exist. Only the distances between objects must be observed. This is convenient for use with the dominant color descriptor, since the latter does not define points in a vector space. The complexity of MDS is $\mathcal{O}(N^2)$, where N is the number of objects.

FastMap is an approach that solves the same problem as MDS, but computationally it is much more effective, since its complexity is $\mathcal{O}(nN)$, where n is the number of dimensions of the target configuration. In the conducted experiments using the corel dataset MDS provided better results in terms of the stress function (see Figure 3). Its high computational cost, however was prohibitive for relatively large image collections. An example of a browsing interface that uses FastMap for $n = 2$ is shown in Figure 4 (a). The visualization result for a larger dataset using the experimentation environment that was developed is shown in Figure 4 (b).

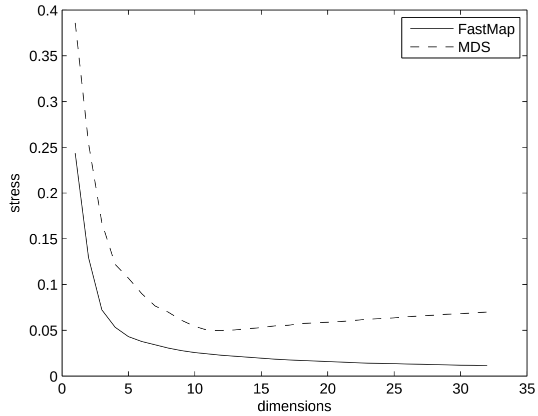


Figure 3. MDS vs. FastMap stress performance for various dimensions. Experiment for 500 images.

5. Efficient indexing with FastMap and kd-trees

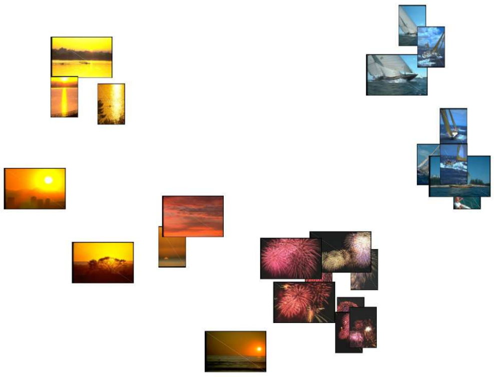
Color / percentage pairs in dominant color descriptors do not define points in a k-dimensional space. Therefore, kd-trees cannot be directly applied for indexing in this case. Even if the descriptors D were to be interpreted as feature vectors with each element (color value or percentage) corresponding to a separate dimension, using 16 colors leads to 32-dimensional feature vectors that kd-trees cannot handle efficiently. This section proposes the combined use of FastMap and kd-trees to address these issues.

5.1. Efficiency of FastMap configurations

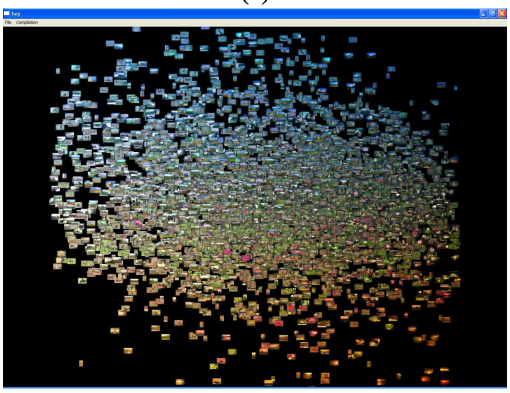
Given a dominant color descriptor D_i for each image I_i in the database, the EMD measures d_{ij} between I_i and I_j are computed for all i, j. Subsequently, FastMap is applied to create a configuration P_n of n-dimensional points, one for each image. This allows the use of kd-trees for indexing. The questions that naturally arise have to do (i) with the quality of the retrieval results and (ii) how these results are affected by the choice of n.

Using the ranking results obtained from EMD based queries (Section 2) as ground truth, the performance of the FastMap configurations for image retrieval was evaluated as follows. All 5122 images of the corel dataset were considered as queries successively and for each index in the EMD results, the corresponding index in the FastMap processed query were noted. For example, a query image from the “action sailing” category gives 2 4 which reads “the 2nd result of EMD was ranked 4th using FastMap”. Figure 5 shows an example of the rank results for a random image and the average for all images for $n = 6$.

Figure 5 shows that even though the results of EMD ranking and those of FastMap configurations are not identical, they are averagely very close to each other at 6 dimensions. The similarity is practically not improved if more

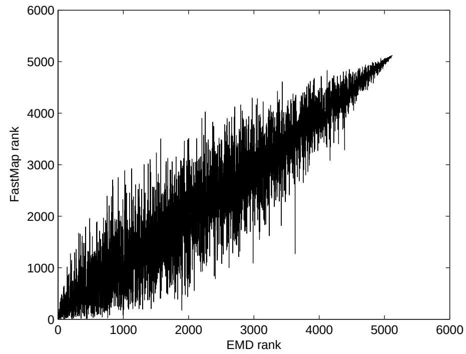


(a)

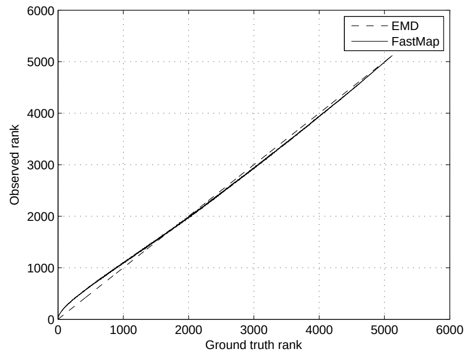


(b)

Figure 4. (a) Result of applying FastMap for two dimensions on a set of images from three categories of the corel dataset. (b) Navigation on a larger dataset. Users can zoom in specific areas, modify all distances by a factor and select specific images.



(a)



(b)

Figure 5. Example rank results for $n = 6$ dimensions. The EMD rank (ground truth) is the $y = x$ line. (a) Results for a random image. (b) The average for all images.

than 10 dimensions are used. Hence one may conclude from this experiment that *by using FastMap configuration for indexing, the retrieval does not deviate significantly from the EMD results*. It is therefore highly unlikely to find images ranked in the first results by EMD at the lower ranks (i.e., least similar results) of FastMap based retrieval.

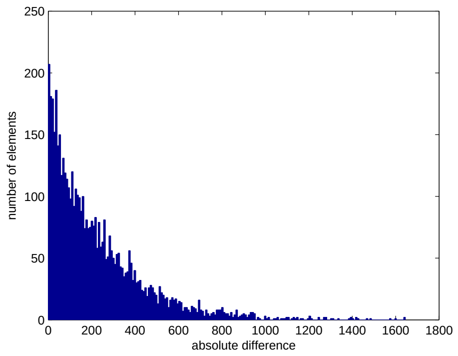
In order to quantify this result, one more experiment was conducted. Two images were selected at random from each category (resulting in 100 images in total) and the retrieval results for these images were observed for EMD and FastMap, as in the previous. Then, the distribution of the variable $\mathbf{d}_r(I, dim) = |r_{FastMap}(I, dim) - r_{EMD}(I)|$ was calculated for 1 to 32 dimensions, where I is an image, $dim = 1, \ldots, 32$ a dimension and $r_{FastMap}(I, dim)$, $r_{EMD}(I)$ are the ranks of FastMap and EMD based retrieval correspondingly for I and dim (Figure 6 (a) shows the result for $dim = 6$ dimensions for a random image). The probabilities $P_I(\mathbf{d}_r \leq k)$ that the difference in rank will be less than k positions was also calculated for all dimensions and all images. The expected values $E\{P_I(\mathbf{d}_r \leq k)\}$ are shown in Figure 6 (b) for various k.

Notice that the expected probability of FastMap and EMD results having a difference lower than 100 is about 0.5 at 6 dimensions. However, the probability that the difference will be lower than 800 is close to 1 (approximately 0.95). Hence the observations made previously are confirmed quantitatively.

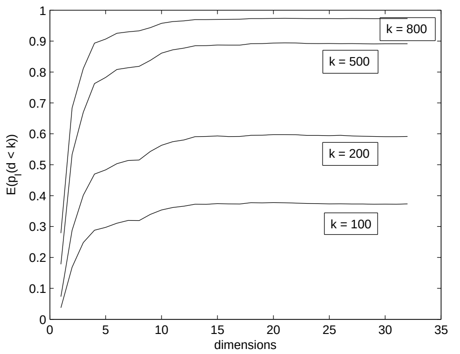
6. Conclusions

From the results of the previous sections several useful conclusions can be drawn.

For the global color representation and specifically the dominant color descriptor, the Earth Mover's Distance appeared to be the most accurate distance metric in the conducted experiments.



(a)



(b)

Figure 6. (a) The distribution (histogram) of $\mathbf{d}_r$ of a randomly selected image for 6 dimensions. (b) The expected value of $P_I(\mathbf{d}_r \leq k)$ for $k = 100, 200, 500, 800$.

For the indexing problem, kd-trees were more effective than exhaustive search, but their practical application is limited to a relatively low number of dimensions. Moreover, kd-tree structures index points in a k-dimensional space that dominant color descriptors do not provide.

For visualization and navigation purposes, the MDS approach proved more precise than FastMap in terms of the stress measure. However its practical use is limited due its high computational cost. FastMap is better suited to large-scale image databases.

Finally, a series of experiments showed that apart from navigation purposes, FastMap can be efficiently used for indexing as well. Constructing 6 to 10 dimensional configurations in FastMap allows the exploitation of kd-trees for faster indexing while keeping the ranking results at acceptable levels. It is important to note that these experiments were performed using EMD results as the ground truth, in lack of another absolute ranking measure. Hence, for the end user the retrieval might as well be better than the one indicated by the values of Figure 6 (b) that only provide the probabilities that the difference between an EMD and a FastMap based ranking will be kept below a certain limit.

The approaches evaluated in this work are by no means exhaustive and the authors believe that a future work with more extensive experimental analysis including other approaches to these problems will prove useful to those interested in developing practical and possibly large scale image retrieval applications.

References

[1] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, 10(1):140–147, January 2001.

[2] A. Dorado and E. Izquierdo. Semantic labeling of images combining color, texture and keywords. In *IEEE International Conference on Image Processing, ICIP*, volume 3, pages 9–12, Barcelona, Spain, September 2003.

[3] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In M. J. Carey and D. A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174, San Jose, California, 22–25 1995.

[4] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. In *New Trends in Computer Graphics*. Springer Verlag, Berlin, 1988.

[5] J. B. Kruskal. Multi-dimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrica*, 29:1–27, 1964.

[6] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[7] B. Manjunath, P. Salembier, and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. Wiley, 2002.

[8] A. W. Moore. An introductory tutorial on kd-trees. Technical Report 209, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Computer Laboratory, University of Cambridge, 1991.

[9] Y. Rubner, C. Tomasi, and L. J. Guibas. Adaptive color-image embeddings for database navigation. In *Proceedings of the 1998 IEEE Asian Conference on Computer Vision*, Hong Kong, 1998.

[10] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, November 2000.

[11] T. Sikora. The mpeg-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–702, June 2001.