

## A Scalable Algorithm for Image Retrieval by Color

Santhana Krishnamachari Mohamed Abdel-Mottaleb  
Philips Research  
345 Scarborough Road  
Briarcliff Manor, NY 10510  
{sgk,msa}@philabs.research.philips.com

### Abstract

*To deal with large databases, we present a clustering based indexing technique, where the images in the database are grouped into clusters of images with similar color content using a hierarchical clustering algorithm. At search time the query image is not compared with all the images in the database, but only with a small subset. Thus the retrieval is scalable to large databases. Experiments show that this clustering based approach offers a superior response time without sacrificing the retrieval accuracy, which is crucial for large databases.*

### 1. Introduction

Content-based image retrieval has become a prominent research topic because of the proliferation of video and image data in digital form. Increased bandwidth availability to access the internet in the near future will allow the users to search for and browse through video and image databases located at remote sites. Therefore fast retrieval of images from large databases is an important problems that needs to be addressed.

Image retrieval systems attempt to search through a database to find images that are perceptually similar to a query image. An ideal image retrieval engine is one that can completely understand a given image, *i.e.*, to identify the various objects present in the image and their properties. Given the state of the art of research in the image analysis community, this remains a utopian dream. Moreover retrieval based on human annotation is to no avail, because of the size of the video and image databases and the varying interpretations that different humans can attach to an image. Some examples of content based image retrieval algorithms can be found in [1-5].

Techniques presented in [1-5] extract specific features from a query image and compare these features with the corresponding pre-computed features of the database images. The search time, therefore, increases linearly with the size of the database. Efficient representations,

like the binary representation [10], have been used to speed-up the feature comparison. Still, the growing size of the database will result in long search delays which may be unacceptable in many practical situations. Even if the time required to compare two images is very short, the cumulative time needed to compare the query image with all database images is rather long and is probably longer than the time an average user wants to wait.

Our approach to solving this problem is to group or cluster the images beforehand, so that at the time of the query, it is not necessary to perform an exhaustive comparison with all the images in the database. Such schemes are being used by present day text retrieval engines. In the case of images, the clustering can be done based on visual features extracted from the images. Techniques for fast image retrieval from large databases presented in [7,8] require that the similarity measures used to compare images be a metric, *i.e.*, the similarity measure should satisfy the triangle inequality. However, many useful similarity measures do not satisfy the triangle inequality measure.

In this paper we present a technique for image retrieval based on color from large databases. The goal is to group similar images into clusters, so that during retrieval, the query image need not be compared with all the images in the database. A subset of clusters is chosen based on their similarity to the query image and only the images in these clusters are compared with the query image. Experiments show that this scheme, while being fast, performs as effective as comparing every image in the database to the query image.

### 2. Image Clustering

Searching large databases of images is a challenging task especially for retrieval by content. Most search engines calculate the similarity between the query image and all the images in the database and rank the images by sorting their similarities. One problem with this approach is that it does not scale up for large databases. The retrieval time is the sum of two times:  $T_{sim}$  and  $T_{sort}$ .

$T_{sim}$  is the time to calculate the similarity between the query and every image in the database, and  $T_{sort}$  is the time to rank all the images in the database according to their similarity to the query.

$$T_{total} = nT_{1sim} + O(n \log n)$$

where  $n$  is the number of images in the database,  $T_{1sim}$  is the time to calculate the similarity between two images, and  $O(n \log n)$  is the time to sort  $n$  elements.

When the images in the database are clustered, the retrieval time is the sum of three times, the time to calculate the similarity between the query and the cluster centers, the time to calculate the similarity between the query and the images in the nearest clusters and the time to rank these images. Therefore the total search time is:

$$T_{total} = kT_{1sim} + lT_{1sim} + O(l \log l)$$

Here  $k$  is the number of clusters,  $l$  is the number of images in the clusters nearest to the query,  $k \ll n$  and  $l \ll n$ .

### 2.1. Image representation

Images in the database are represented by multiple color histograms. Representation with multiple histograms capture the local color variations in the image. Some image retrieval systems attempt to segment the image into constituent objects and then extract the features for individual objects. However, general image segmentation is in itself a very difficult problem.

The similarity between two histograms is calculated using the histogram intersection measure [1]. Given two normalized histograms,  $P = \{p_1, p_2, \dots, p_N\}$ ,  $Q = \{q_1, q_2, \dots, q_N\}$ , the histogram intersection similarity measure is defined as  $\sum_k \min(p_k, q_k)$ . The similarity measure,  $s_{i,j}$ , between two images,  $i$  and  $j$ , is calculated by computing the cumulative value of the histogram intersection measure between the corresponding histograms.

The clustering algorithm described below does not directly depend on the histogram intersection similarity measure. Any desired or application-relevant similarity measure can be used with the following algorithm. We have reported the results of comparing various different similarity measures with the following clustering algorithm in [9].

### 2.2. Clustering

The images in the database are clustered into groups using the hierarchical clustering algorithm [6], as shown below.

1. Let  $n$  be the number of images in the database. These  $n$  images in the database

are placed in  $n$  distinct clusters indexed by  $\{C_1, C_2, \dots, C_n\}$ .

2. Two distinct unmerged clusters  $C_k$  and  $C_l$  are picked such that the similarity measure is the largest.

3. These two clusters are merged into a new cluster. At each step two clusters are merged to form a new cluster. Therefore, the number of clusters is reduced by one.

4. Steps 2 and 3 are repeated until the number of unmerged clusters has reduced to a required number  $n_c$  or the largest similarity measure between clusters has dropped to some lower threshold.

Figure 1 shows a simple example of hierarchical clustering with eight images. The clustering is stopped after reaching two clusters.

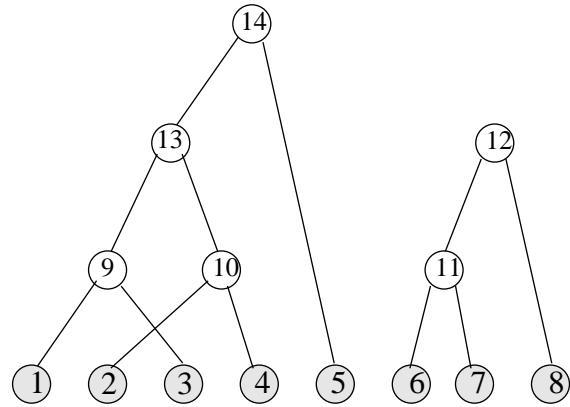


Figure 1: A sample of cluster merging process with hierarchical clustering.

The similarity measure between two images is defined in the previous section. The measure of similarity,  $S_{k,l}$ , between two clusters,  $C_k$  and  $C_l$ , is defined in terms of the similarity measures of the images that are contained in those clusters as follows:

$$S_{k,l} = \frac{\sum_{i,j \in \{E_k \cup E_l\}, i \neq j} s_{i,j}}{P_{(N_k + N_l)}} \quad (1)$$

where  $E_k$  is the set of images present in the cluster  $C_k$  and  $N_k$  is the number of images in cluster  $C_k$ .  $P_n$  is the number of pairs of images in a cluster with  $n$  images:

$$P_n = (n - 1) \frac{n}{2}$$

In other words,  $S_{k,l}$  is defined to be the average similarity between all pairs of images that will be present in the cluster obtained by merging  $C_k$  and  $C_l$ . This ensures that when two clusters are merged, the resulting cluster has the largest average similarity between all images in those two clusters. Since the similarity between clusters is defined in terms of the similarity measures between the images in the clusters, there is no need to compute the cluster centers every time two clusters are merged.

When two clusters,  $C_k$  and  $C_l$ , are merged to form a new cluster  $C_m$ , then it is necessary to compute the similarity of this cluster with all other unmerged clusters as given in Eq. (1). This computation is cumbersome as shown below. For any cluster  $C_p$ , the similarity measure between  $C_m$  and  $C_p$  is:

$$S_{m,t} = \left( \sum_{i,j \in \{E_l \cup E_k\}, i \neq j} s_{i,j} + \sum_{i,j \in E_p, i \neq j} s_{i,j} + \sum_{i \in E_l \cup E_k, j \in E_t} s_{i,j} \right) / P_{(N_l + N_k + N_t)} \quad (2)$$

and  $S_{m,m}$  is set equal to  $S_{k,l}$ .

A simple recursive method to achieve the same can be obtained using the fact that the first term in Eq. (2) is equal to  $P_{(N_l + N_k)} S_{l,k}$ , the second term is equal to  $P_{N_t} S_{t,t}$ , and the third term is equal to  $P_{(N_l + N_t)} S_{l,t} + P_{(N_k + N_t)} S_{k,t} - P_{N_l} S_{l,l} - P_{N_k} S_{k,k} - 2P_{N_t} S_{t,t}$ . Thus the similarity  $S_{m,t}$  can be obtained from,  $S_{l,k}$ ,  $S_{l,t}$ ,  $S_{k,t}$ ,  $S_{t,t}$ ,  $S_{l,l}$ , and  $S_{k,k}$ . The following equation requires far less computation compared to the one above.

$$S_{m,t} = (P_{(N_l + N_k)} S_{l,k} + P_{(N_l + N_t)} S_{l,t} + P_{(N_k + N_t)} S_{k,t} - P_{N_l} S_{l,l} - P_{N_k} S_{k,k} - P_{N_t} S_{t,t}) / P_{(N_l + N_k + N_t)} \quad (3)$$

In Eq. (2),  $S_{m,t}$  is computed by summing up the similarity measures of all pairs of images in  $C_m$  and  $C_t$ , and hence the computation grows as the square of number of images present in the two clusters. The computation in Eq. (3) is independent of the number of images in the clusters. At the beginning of clustering, for all the clusters  $S_{i,j}$  is set equal to  $s_{i,j}$  and  $S_{i,i}$  is set equal to zero.

After the clustering process is completed, each cluster is represented by a cluster center. Similar to image representation, cluster centers are also represented by multiple histograms that are obtained averaging the histograms of the representative images in the cluster.

These representative images are chosen using the tree structure of each cluster obtained from the hierarchical clustering.

### 3. Experimental Setup

The results presented here are obtained with a database of 2000 images, 200 of these images are taken from two collections of COREL Professional Photo CD-ROMs, the Samper II - Series 400000 and the Sampler - Series 200000. The rest of the images are obtained from the Department of Water Resources, California. The images are of widely varying colors and scene content. The hierarchical clustering algorithm was applied to the 2000 images in the database resulting in 133 clusters with the largest cluster having 39 images and the smallest cluster having 2 images. The number of clusters,  $n_c$ , is chosen such that the average number of images per cluster is 15, i.e.,  $n_c = 2000/15 = 133$ .

#### 3.1. Retrieval Accuracy with Clustering

Performance evaluation has long been a difficult problem in image processing and computer vision, and content-based retrieval is no exception. This is primarily because of the difficulty associated with relevant quantitative measures for evaluation. In content-based retrieval, precision and recall measures have been frequently used by many researchers [5] to evaluate the performance of retrieval algorithms.

We have used a quantitative measure to compare the retrieval results *with* clustering against the retrieval results *without* clustering. A user searching through a large database, is interested in only the top few best matches (say 10 or 20). Hence, if the retrieval with clustering returns the same few best matches as the ones returned by retrieval without clustering, then the retrieval with clustering is very accurate. Assuming that the user is interested in only the top  $K$  best matches and that  $M$  is the number of images that are present both in the top  $K$  results returned by retrieval with and without clustering, the retrieval accuracy with clustering  $\psi_i$ , when  $i$  th image is used as a query is defined as:

$$\psi_i = \frac{M}{K} 100$$

The average retrieval accuracy with clustering,  $A_K$ , is obtained by taking the average  $\psi_i$  over all query images..

$$A_K = \frac{1}{n} \sum_{i=1}^n \psi_i$$

Figure 2 shows the average retrieval accuracy values obtained for  $K=10$  and  $20$ . The average values are obtained from using each of the 2000 images in the database as a query image. Each plot contains eight points, obtained by examining the top 3,4,7,10,13,19,25, and 31 clusters. The leftmost point corresponds to the result obtained by examining 3 clusters and the rightmost point corresponds to the result obtained by examining 31 clusters. Figure 1 shows that the retrieval accuracy with clustering is around 90% when only thirteen clusters (out of 133) that are most similar to the query are examined. On average, examining the top thirteen clusters required only about 300 image similarity comparisons. Note that 2000 comparisons would be required to compare the query with all the database images. Hence our clustering scheme has achieved the computational gain without losing the retrieval performance. We expect the computational gain to be much better as the size of the database grows larger. In a second experiment, we used a set of 300 images that are *not* a part of the database, as queries. Figure 3 shows the average retrieval accuracy values obtained for  $K=10$  and  $20$ . The performance here is comparable to that in Figure 2.

#### 4. Conclusions

In this paper we have presented an algorithm for scalable image retrieval by color, where images are represented by local color histograms. The similarity between images is calculated based on local color properties. Images in the database are clustered into groups with similar color contents. This grouping enables searching only images that are relevant to the query image. The clustering technique can also be used for browsing a large database of images and video clips.

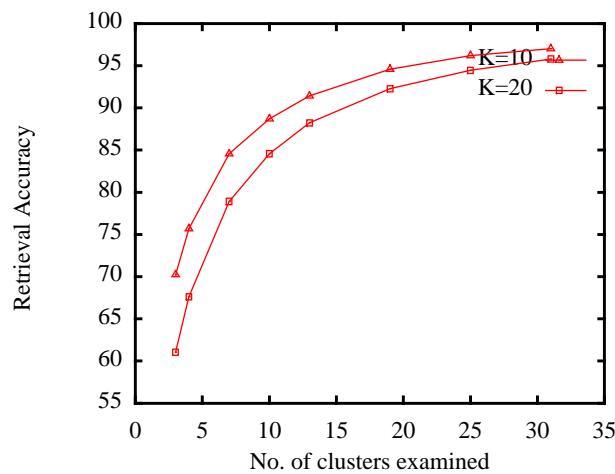


Figure 2: Average retrieval accuracy with database images used as queries

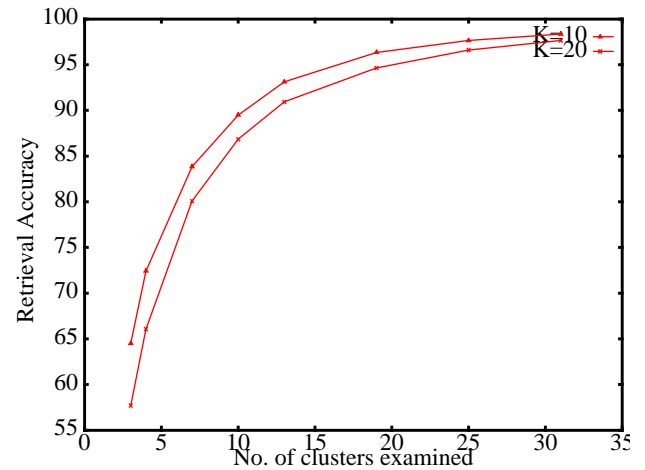


Figure 3: Average retrieval accuracy with external images used as queries

#### References

- [1] M. J. Swain and D. H. Ballard, Color Indexing, Intl. J. of Computer Vision, 7(1), pp. 11-32, 1991.
- [2] M. Abdel-Mottaleb, N. Dimitrova, R. Desai, and J. Martino, "CONIVAS: CONtent-based image and video access stem, Proc. of ACM Intl. Multimedia Conference, Nov. 1996.
- [3] W. Niblack, R. Barber, et. al., The QBIC Project: Querying images by content using color, texture and shape. In Storage and Retrieval for Image and Video Databases I, Vol. 1908, SPIE Proceedings, Feb. 1993.
- [4] J. Smith and S.-F. Chang, A fully automated content-based image query system, Proc. of ACM Intl. Multimedia Conference, Nov. 1996.
- [5] H. Zhang, Y. Gong, C. Y. Low and S. W. Smoliar, Image retrieval based on color features: an evaluation study, Proc of SPIE, Vol 2606, pp. 212-220, 1995.
- [6] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice Hall, New Jersey, 1988.
- [7] J.-Y. Chen, C.A. Bouman, and J.P. Allebach, "Fast Image Database Search using Tree-Structure VQ", Proc. of ICIP, Vol 1, 1997.
- [8] J. Barros, et.al., "Using the triangle inequality to reduce the number of computations required for similarity-based retrieval", Proc. of SPIE/IS&T Conf. on Storage and Retrieval of Image and Video Databases IV, Vol. 2670, 1996.
- [9] M. Abdel-Mottaleb, S. Krishnamachari, and N.J. Mankovich, "Performance Evaluation of Clustering Algorithms for Scalable Image Retrieval", In IEEE Computer Society Workshop on Empirical Evaluation of Computer Vision Algorithms", Santa Barbara, 1998.
- [10] W.Y. Ma and B.S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases", pp 568-572, Proc. of ICIP, Vol 1, 1997.