Indian Journal of Science and Technology http://www.indjst.org Vol.1 No 5 (Oct. 2008)

# Content based image retrieval using signature based similarity search

D. Rajya Lakshmi*, A.Damodaram[1], M.Sreenivasa Rao[2] and J.A.Chandu Lal[3]
*GITAM Engg. College, Visakhapatnam, India; [1]JNTU college of Engg, Hyderabad.*
*[2]JNT University, Hyderabad; [3]GITAM University, Visakhapatnam.*
dlakmi@rediffmail.com

**Abstract:** Two of the main components of the visual information are texture and color. In this paper, a content-based image retrieval system (CBIR), which computes texture and color similarity among images, is presented. CBIR is a set of techniques for retrieving semantically-relevant images from an image database based on automatically-derived image features. One of the main tasks for CBIR systems is similarity comparison, extracting feature signatures of every image based on its pixel values and defining rules for comparing images. These features become the image representation for measuring similarity with other images in the database. Images are compared by calculating the difference of its feature components to other image descriptors. Previously CBIR methods used global feature extraction to obtain the image descriptors. For example, several features like color, texture and shape extracted from each image. These descriptors are obtained globally by extracting information on the means of color histograms for color features; global texture information on coarseness, contrast, and direction; and shape features about the curvature, moments invariants, circularity, and eccentricity. These global approaches are not adequate to support queries looking for images where specific objects in an image having particular colors and/or texture are present, and shift/scale invariant queries, where the position and/or the dimension of the query objects may not relevant. For example, suppose in one image there are two flowers with different colors: red and yellow, the global features describe the image as the average of the global average color which is orange. This description is certainly not the representation of the semantic meaning of the image. Therefore, the weakness of global features is observable. Region-based retrieval systems attempt to overcome previous method limitations of global based retrieval systems by representing images as collections of regions that may correspond to objects such as flowers, trees, skies, and mountains.

*Keywords*: Content based, image retrieval, binary signature, region-based, Debouche compression, segmentation.

## Introduction

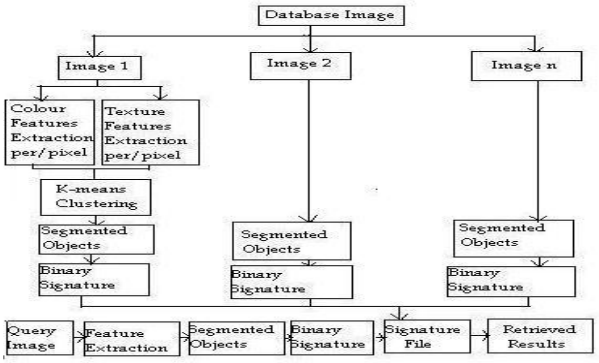
In recent years, very large collections of images and videos have grown rapidly. In parallel with this growth, content-based retrieval and querying the indexed collections are required to access visual information.

There are two general approaches for image retrieval. One is Text Based Image Retrieval and Content Based Image Retrieval. The text-based approaches apply traditional text retrieval techniques to image annotations or descriptions. One major problem with this is the task of describing image content is highly subjective. The perspective of textual descriptions given by an annotator could be different from the perspective of a user. A picture can mean different things to different people. It can also do different things to the same person at different time and with the same view, the words used to describe the content could vary from one person to another (Keister, 1994). In other words, there could be a variety of inconsistencies between user textual queries and image annotations or descriptions.

Content-based search will analyze the actual contents of the image. The term content refers to colors, shapes, textures, or any other information that can be derived from the image itself. The content-based approaches apply image processing techniques to extract image features and retrieve relevant images. We have developed a signature based similarity search and also introduced signature file for storing the feature descriptions. The approach is the color and texture features are extracted from the each image into color and texture Vector. For better performance these two vectors are combined and form a feature vector of an image. The feature vector is transformed into a binary signature and stored into a signature file. Thus image database is transformed into signature file storage.

The experimental evaluation is based on the 1000 COREL color image database with the Fuzzy Club, IRM and Geometric Histogram and the performance is compared between them. As compared with existing technique and systems, such as IRM, Fuzzy Club, and Geometric Histogram, the present study demonstrate the following unique advantages: (i) an improvement in feature extraction accuracy k-means algorithm (ii) an improvement in retrieval accuracy as a result of a better similarity distance and fast retrieval and iii) one to one image indexing.

*Fig. 1. Diagram for Content-Based Image Retrieval System*




## Content Based Image Retrieval

Content Based Image Retrieval is a set of techniques for retrieving semantically-relevant images from an image database based on derived image features. The main goal of CBIR is efficiency during image indexing and retrieval, thereby reducing the need for human intervention in the indexing process. The computer must be able to retrieve images from a database without any human assumption on specific domain. If searches depend on metadata such as captions or keywords, it becomes laborious or expensive unless CBIR system available. One of the main tasks for CBIR systems is similarity comparison, extracting feature signatures of every image based on its pixel values and defining rules for comparing images. These features become the image representation for measuring similarity images in the database. Images are matched by calculating the difference of its feature components to other image descriptors.

*CBIR Methodology*

The main processing steps in the region based CBIR are: 1) Pre-processing of the Image to reduce the total number of pixels per Image (without compromising the image quality) in order to reduce the total number of pixels per image and therefore decreasing the computation time, 2) From each image the features (color + texture) of each pixel are extracted, 3) Images are segmented into objects of similar descriptions, 4) Feature vector is converted to binary signature vector, and 5) Similarity distance is computed between the image query and database.

Fig. 1. provides the complete overview of the system architecture developed for region based CBIR. Feature selection during image segmentation is a crucial step. Feature vector is generated by combining the color space and texture of the image. Every pixel on the image is clustered using a k-means algorithm to group similar pixel together to form objects. The new similarity distance algorithm is introduced to minimize error obtained during image segmentation. Finally, the accuracy during retrieval is computed and compared against IRM, Geometric Histogram, and Fuzzy Club system.

*Similarity Search Methods and Techniques*

Similarity searching has become a fundamental computational task in a variety of application areas, including multimedia information retrieval, data mining, pattern recognition, machine learning, computer vision, biomedical databases, data compression and statistical data analysis. In such environments, an exact match has little meaning, and proximity/distance (similarity/ dissimilarity) concepts are typically much more fruitful for searching.

The search in traditional DBMS provides precise results, that is, an object either belongs to the result set or it does not. However, modern information resources, including the data found on the Web, do not always require such search precision, or precise answers are not possible at all. Since the traditional exact, partial, and range retrieval paradigms fail to satisfy the content-based retrieval needs of many emerging data processing applications, the concepts of similarity/proximity are becoming more and more relevant.

Many technical articles on similarity search from different application environments- including multimedia, scientific, and genomic databases - have already been published. However, these results are often isolated and are based on pragmatic peculiarities of specific data types (e.g., images, text), thus difficult to apply in other environments. Prototype systems have also demonstrated poor and often unpredictable performance.

*Signature Files*

A signature file (Faloutsos & Christodoulakis, 1987; Lin & Faloutsos,1988; Rijsbergen,1979; Roberts,1979) is a file that stores a signature record for each Image in the database. Each signature has a fixed size of b bits representing terms. A simple encoding scheme goes as follows. Each bit of a document signature is initialized to 0. A bit is set to 1 if the term it represents appears in the document. A signature s1 matches another signature s2 if each bit that is set in signature s2 is also set in s1. Since there are usually more terms than available bits, there may be multiple terms mapped into the same bit. Such multiple-to-one mappings make the search expensive since a image that matches the signature of a query does not necessarily contain the set of keywords of the

query. Improvements can be made by first performed frequency analysis, stemmed, and by filtering stop words, and then use a hashing technique and superimposed coding technique to encode the list of terms into bit representation. Nevertheless, the problem of multiple-to-one mappings still exists, which is the major disadvantage of this approach.

*Performance Evaluation of CBIR Systems*

Our content-based image retrieval system is first evaluated in terms of retrieval effectiveness. In order to evaluate effectiveness of retrieval systems, two well known metrics, precision and recall are used:

Precision =(the number of retrieved images that are relevant) / (The number of retrieved images)

Recall =(the number of retrieved images that are relevant) / (The total number of relevant images).

For a query q, the data set of images in the database that are relevant to the query q is denoted as $R(q)$, and the retrieval result of the query q is denoted as $Q(q)$. The precision of the retrieval is defined as the fraction of the retrieved images that are indeed relevant for the query as shown in equation (1):

$$\text{Precision} = \frac{|Q(q) \cap R(q)|}{|Q(q)|} \quad \text{.....(1)}$$

The recall is the fraction of relevant images that is returned by the query as shown in equation (2):

$$\text{Recall} = \frac{|Q(q) \cap R(q)|}{|R(q)|} \quad \text{.....(2)}$$

Usually, a tradeoff must be made between these two measures since improving one will sacrifice the other. In typical retrieval systems, recall tends to increase as the number of retrieved items increases; while at the same time the precision is likely to decrease. In addition, selecting a relevant data set $R(q)$ is much less stable due to various interpretations of the images. Further, when the number of relevant images is greater than the number of the retrieved images, recall is meaningless. As a result, precision and recall are only rough descriptions of the performance of the retrieval system.

*K- means Clustering Algorithm*

K-means (MacQueen,1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids as binary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

.....(3), Where

$\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre $c_j$,, is an indicator of the distance of the *n* data points from their respective cluster centers.

The algorithm is composed of the following steps: 1) Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids, 2) Assign each object to the group that has the closest centroid, 3) When all objects have been assigned, recalculate the positions of the K centroids, and 4) Repeat Steps 2 and 3 until the centroids no longer move. It results in separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect.

K-means is a simple algorithm that has been adapted to many problem domains and a good candidate for extension to work with fuzzy feature vectors.

*An Example*

Suppose we have n sample feature vectors $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$ all from the same class, and we know that they fall into k compact clusters, k < n. Let $\mathbf{m}_i$ be

the mean of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. That is, we can say that **x** is in cluster i if || **x** - $\mathbf{m}_i$ || is the minimum of all the k distances. This suggests the following procedure for finding the k means:

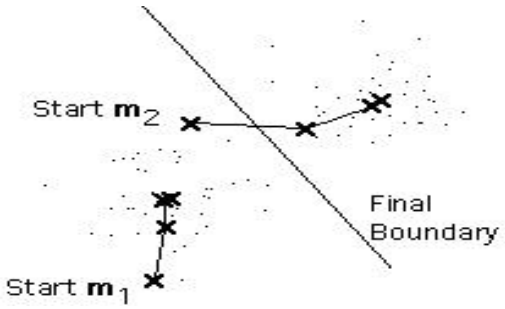
- Make initial guesses for the

Means **m1, m2…., mk**

- Until there are no changes in any Mean
  - o Use the estimated means to classify the samples into clusters
  - o For i from 1 to k
    - Replace $\mathbf{m}_i$ with the mean of all of
    - the samples for cluster i
  - o end_for
- End_until

The Fig. 2. shows an example, how the means $\mathbf{m}_1$ and $\mathbf{m}_2$ move into the centers of two clusters.

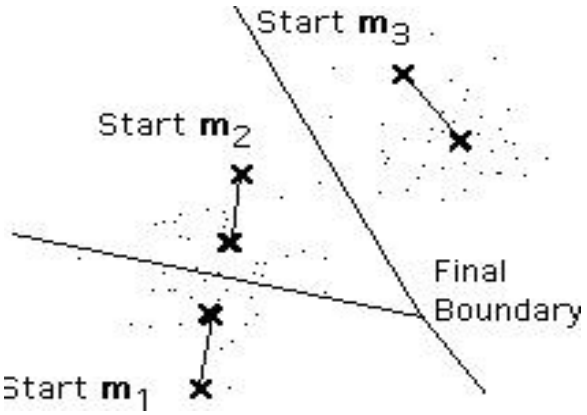
*Fig. 2. Grouping of data sets into clusters*.




*Remarks*

This is a simple version of the k-means procedure. It can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. It does have some weakness: i) the way to initialize the means was not specified. One popular way to start is to randomly choose k of the samples, ii) the results produced depend on the initial values for the means, and it frequently happens that suboptimal partitions are found. The standard solution is to try a number of different starting points, iii) It can happen that the set of samples closest to $\mathbf{m}_i$ is empty, so that $\mathbf{m}_i$ cannot be updated. This is an annoyance that must be handled in an implementation, but that we shall ignore, iv) the results depend on the metric used to measure || **x** - $\mathbf{m}_i$ ||. A popular solution is to normalize each variable by its standard deviation, though this is not always desirable, and v) the results depend on the value of k.

*Fig. 3. Grouping of modified data sets In to clusters*.




This last problem is particularly troublesome, since we often have no way of knowing how many clusters exist. As shown in Fig. 3. below, the same algorithm applied to the same data produces the following K-means clustering.

Unfortunately there is no general theoretical solution to find the optimal number of clusters for any given data set. A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion, but we need to be careful because increasing k results in smaller error function values by definition, but also an increasing risk of over fitting.

Each object described by 6 features which are the average features of all the member pixels. This information is stored in an array database (Table 1).

As part of preprocessing, each 4x4 block is replaced by a single block containing the average value. This way, we still have a good texture granularity while reducing the number of total pixels per image, therefore decreasing the computation time.

To segment an image into objects, six features are extracted from each pixel. Three features are

*Table 1. Feature information for each object in an image.*

| Name | F1 | F2 | F3 | F4 | F5 | F6 |
|---|---|---|---|---|---|---|
| Image1Object1 | H11 | S11 | V11 | Ht11 | D11 | Vt11 |
| Image1Object2 | H12 | S12 | V12 | Ht12 | D12 | Vt12 |
| Image2Object1 | H21 | S21 | V21 | Ht21 | D21 | Vt21 |
| Image2Object2 | H22 | S22 | V22 | Ht22 | D22 | Vt22 |
| …… | … | .. | … | … | … | … |
| ImagenObjecti | Hni | Sni | Vni | Htni | Dni | Vtni |

color features, and the other three are texture features. The HSV color space is selected during color feature extraction due to its ability for easy transformation from RGB to HSV and vice versa. Since HSV color space is natural and approximately perceptually uniform, the quantization of HSV can produce a collection of colors that is also compact and complete. These features are denoted as (F1, F2, and F3).

To obtain the other three texture features, we apply the Haar wavelet transform to the image. The Haar wavelet is discontinuous and resembles a step function.

After a one-level wavelet transform, a 4 by 4 block is decomposed into four frequency bands, each band containing a 2 by 2 matrix of coefficients. Then, the feature of the block in the HL band is computed. The other two features are computed similarly in the LH and HH bands. These three features of the block are denoted as (F4, F5 & F6).

**Results**

We developed a signature based similarity search and also introduced signature file for storing the feature descriptions. The approach is the color and texture features are extracted from the each image into color and texture Vector. For better performance these two vectors are combined and form a feature vector of an image. The feature vector is transformed into a binary signature and stored into a signature file. Thus image database is transformed into signature file storage.

The experimental evaluation is based on the 1000 COREL color image database with the Fuzzy Club, IRM and Geometric Histogram and the performance (Gong *et al*., 2000) is compared between them. As compared with existing technique and systems, such as IRM, Fuzzy Club, and Geometric Histogram, our study demonstrate the following unique advantages: (i) an improvement in feature extraction accuracy k-means algorithm, and (ii) an improvement in retrieval accuracy as a result of a better similarity distance and fast.

*Processing tools and statistical tools*

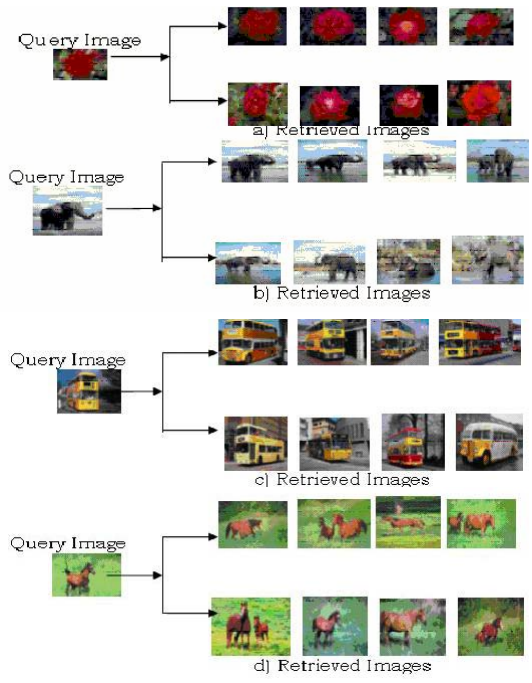
The image retrieval system is implemented using MATLAB image retrieval and one to one image indexing. We use a general-purpose image database containing 1000 images from COREL. These images are pre-categorized into 10 groups (Fig 4): African people, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountains and glaciers, and food. All images have the size of 384x256 and 256x386. These images are manually divided into 10 classes such as African people, busses, building, and flowers. Feature extraction time for the whole database takes 15-20 minutes using MATLAB software corresponding to about one second for each image.

*Fig. 4. Image categorization*

To see the performance of our algorithm we randomly selected four images from different class, namely flower, horse, bus, and elephant. Each query returns the top 8 images from database. The four query retrievals are shown in Fig. 5a-d. The best result from the object cluster in Experiment are then compared (Rao & Pujari, 1999) to three existing algorithm: IRM, FuzzyClub, and Geometric Histogram. Gong *et al*. (2000) already have the performance comparison between FuzzyClub, IRM and Geometric Histogram.

We used the performance comparison against our algorithm. In order to calculate the performance, we used the same approach as that of Rao and Pujari (1999). For each category in the 1000 database images, we randomly select 20 images as queries. Since we have 10 categories in the database, we have 200 query images. For

*Fig. 5.a-d. Query retrieval.*



each query, we examined the precision of the retrieval based on the relevance of the semantic meaning between the query and the retrieved images. Each of the 10 categories in the database portrays a distinct semantic topic, therefore this assumption is reasonable to calculate the precision. The average precisions for each groups based on the returned top 8 images were recorded. Since the number of relevant images in the database for each query image is the same, we do not calculate the recall explicitly since it's proportional to the precision in this case. The experiment ran once for each algorithm.

The comparison Table 2 shows that our algorithm (Hierarchical) performs significantly better than Geometric Histogram and furthermore, compared to FuzzyClub and IRM.

*Table 2. Comparison with previous systems.*

| Class | Fuzzy Club | IRM | Geometric Histogram | Signature based CBIR |
|---|---|---|---|---|
| 1. African People | 65% | 47% | 12.5% | 42% |
| 2. Beach | 45% | 32% | 13% | 46% |
| 3. Buildings | 55% | 31% | 19% | 25% |
| 4. Buses | 70% | 61% | 11% | 83% |
| 5. Dinosaurs | 95% | 94% | 16% | 92% |
| 6. Elephants | 30% | 26% | 19% | 95% |
| 7. Flowers | 30% | 62% | 15% | 96% |
| 8. Horses | 85% | 61% | 11% | 89% |
| 9.Mountains | 35% | 23% | 22% | 32% |
| 10.Food | 49% | 49% | 15% | 28% |

## Conclusion

i) We have developed improved algorithm for image segmentation into objects using k-means clustering, ii) Developed an improved object clustering algorithm and an improved similarity distance computation. To get faster retrieval speeds, we implemented signature file retrieval methodology, iii) Analyzed query performance on a 1000 image COREL database, and iv) Compared query performance with the well-known IRM and Fuzzy Club region based image retrieval systems.

## References

1. Faloutsos C and Christodoulakis S (1987) Description and performance analysis of signature file methods. ACM TOOLS, 5 (3), p:23.
2. Gong CY, Zhang HJ and Chua TC (2000) An image database system with content capturing and fast image indexing abilities. Proc. IEEE *Intnl. Conf. Multimedia Computing and Systems*, Boston, 14-19 May 2000, pp:121-130.
3. Keister LH (1994) User types and queries: impact on image access systems. In: Challenges in indexing electronic text and images, Fidel R *et al.* (Eds)., ASIS, pp: 7-22.
4. Lin Z and Faloutsos C (1988) Frame sliced signature files. CS-TR-2146 and CMI-ACS-TR-88 -88, Deptt. Computer Science, Univ. Maryland.
5. MacQueen JB (1967) Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability*, Berkeley, Univ. California Press, 1, 281-297.
6. Rao MS and Pujari AK (1999) A New Neural Networks architecture with associative memory, pruning and order sensitive learning, *Intnl. J. Neural Systems,* 9(4), 351-370.
7. Rijsbergen CJ (1979) Information Retrieval. 2nd Ed. Buttersworths, London.
8. Roberts CS (1979) Partial match retrieval via the method of superimposed codes. *Proc. IEEE*, 67 (12), 1624-1642.