Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

(Special Issue for ITECE 2016)

Content Based Image Recognition by color and texture features of image

Asst. Prof. Nishant Sanghani[1], Asst. Prof. Pooja Vasani[2], Asst. Prof. Ravi Khimani[3]

[1]Computer Science & Engineering, SLTIET
[2]Computer Science & Engineering, AITS
[3]Computer Science & Engineering, SLTIET

Abstract —Nowadays digital database growing rapidly in size and so to find the image we need expediently in the spatial relationship feature and then classify the tremendous database into different sorts. There is also difficulty in searching images on web. Content based image retrieval system allows the user to present a query image in order to retrieve images stored in the database according to their similarity to the query image. Content based image retrieval system refers to the retrieval of content of images rather than the standard keywords. In this system composite feature measure, which combines the color and shape features of an image? Improved performance of retrieval by color features that generated from efficient color quantization and considered the spatial correlation with the Histogram matrix matching method. Here it also supplemented the color information using shape information with the improved moment invariants. Testing of the technique on image database consisting of different images. The method of system is the robust than the previous method under the various situations such as rotation images, translation images, noise added images, gamma corrected images and so on.

Keywords: Image retrieval, Object recognition, Image Classification, learning hierarchical multiple classifiers

I. INTRODUCTION

Advances in data storage and image acquisition technologies have enabled the creation of large image datasets. In order to deal with these data, it is necessary to develop appropriate information systems to efficiently manage these collections. Image searching is one of the most important services that need to be supported by such systems. In general, two different approaches have been applied to allow searching on image collections: one based on image textual metadata and another based on image content information.

The first retrieval approach is based on attaching textual metadata to each image and uses traditional database query techniques to retrieve them by keywords. However, these systems require a previous annotation of the database images, which is a very laborious and time-consuming task. In fact, different users tend to use different words to describe a same image characteristic. The lack of systematization in the annotation process decreases the performance of the keyword-based image search.

These shortcomings have been addressed by the so-called Content-Based Image Retrieval (CBIR) systems. In these systems, image processing algorithms (usually automatic) are used to extract feature vectors that represent image properties such as color, texture, and shape. In this approach, it is possible to retrieve images similar to one chosen by the user (query-by-example). One of the main advantages of this approach is the possibility of an automatic retrieval process, contrasting to the effort needed to annotate images.

II. BACKGROUND

The earliest use of the term content-based image retrieval in the literature seems to have been by Kato [1992], to describe his experiments into automatic retrieval of images from a database by colour and shape feature. The term has since been widely used to describe the process of retrieving desired images from a large collection on the basis of features (such as colour, texture and shape) that can be automatically extracted from the images themselves. The features used for retrieval can be either primitive or semantic, but the extraction process must be predominantly automatic. Retrieval of images by manually-assigned keywords is definitely not CBIR as the term is generally understood – even if the keywords describe image content.

CBIR draws many of its methods from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from these fields principally through its emphasis on the retrieval of images with desired characteristics from a collection of significant size. Image processing covers a much wider field, including image enhancement, compression, transmission, and interpretation. While there are grey areas (such as object recognition by feature analysis), the distinction between mainstream image analysis and CBIR is usually fairly clear-cut. An example may make this clear. Many police forces now use automatic face recognition systems. Such systems may be used in one of two ways. Firstly, the image in front of the camera may be compared with a single individual's database record to verify his or her identity. In this case, only two images are matched, a process few observers would call CBIR. Secondly, the entire database may be searched to find the most closely matching images. This is a genuine example of CBIR.

Text-based retrieval and Content-based retrieval

An image retrieval system is a computer system for browsing, searching and retrieving images in an image database. Text-based and content-based are the two techniques adopted for search and retrieval in image database.

In text-based retrieval, images are indexed using keywords, subject headings or classification codes, which in turn are used as retrieval keys during search and retrieval. Text-based retrieval is non-standardized because different users use different keywords for annotation. Text descriptions are sometimes subjective and incomplete because it cannot depict complicated image features very well. Examples are texture images that cannot be described by text. In text retrieval, humans are required to personally describe every image in the database, so for a large image database the technique is cumbersome, expensive and labour-intensive [7].

Content-based image retrieval (CBIR) technique use image content to search and retrieve digital images. Content-based image retrieval system was introduced to address the problems associated with text-based image retrieval. However, text-based and content-based image retrieval techniques complement each other. Text-based techniques can capture high-level feature representation and concepts. It is easy to issue text queries but text-based techniques cannot accept pictorial queries. On the other hand, content-based techniques can capture low-level image features and accept pictorial queries. But they cannot capture high-level concepts effectively. Retrieval systems exist which combine both techniques for more efficient retrieval.

Figure shows a typical architecture of content-based image retrieval system. Two main functionalities are supported: data insertion and query processing. The data insertion subsystem is responsible for extracting appropriate features from images and storing them into the image database. This process is usually performed off-line.

The query processing, in turn, is organized as follows: the interface allows a user to specify a query by means of a query pattern and to visualize the retrieved similar images. The query-processing module extracts a feature vector from a query pattern and applies a metric (such as the Euclidean distance) to evaluate the similarity between the query image and the database images. Next, it ranks the database images in a decreasing order of similarity to the query image and forwards the most similar images to the interface module [1-6].

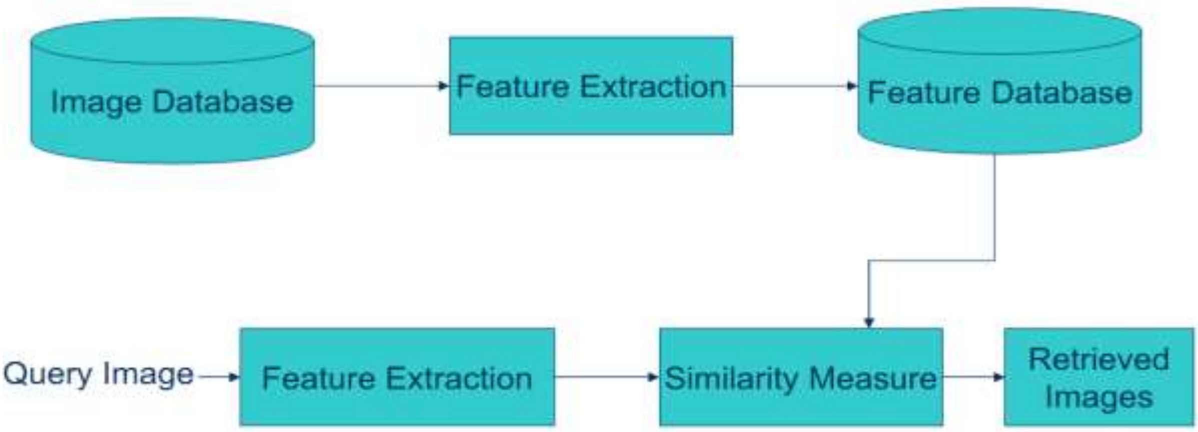

Figure 1. Basic CBIR Architecture

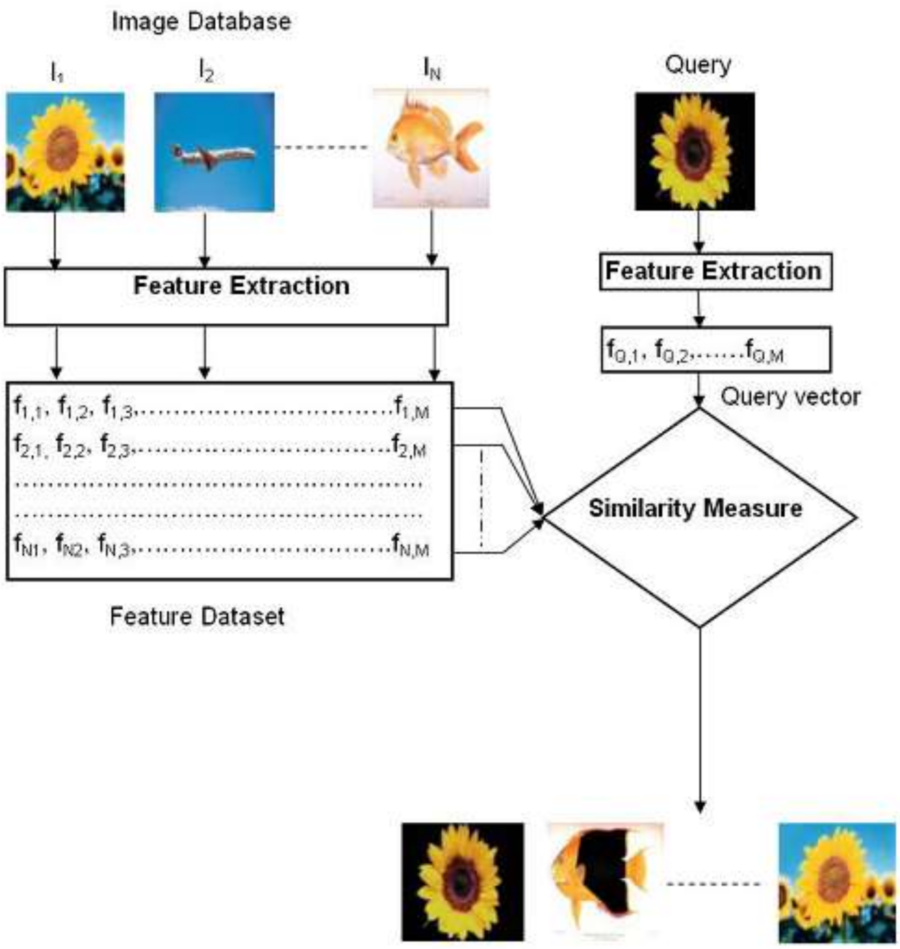

Figure 2. CBIR Processed Image Example

III. TECHNIQUES IN CBIR

3.1 Image Content Descriptors

Generally speaking, image content may include both visual and semantic content. Visual content can be very general or domain specific. General visual content include color, texture, shape, spatial relationship, etc. Domain specific visual content, like human faces, is application dependent and may involve domain knowledge. Semantic content is obtained either by textual annotation or by complex inference procedures based on visual content. This chapter concentrates on general visual contents descriptions. Later chapters discuss domain specific and semantic contents [9].

A good visual content descriptor should be invariant to the accidental variance introduced by the imaging process (e.g., the variation of the illuminant of the scene). However, there is a tradeoff between the invariance and the discriminative power of visual features, since a very wide class of invariance loses the ability to discriminate between essential differences. Invariant description has been largely investigated in computer vision (like object recognition), but is relatively new in image retrieval.

A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions or objects to describe the image content. To obtain the local visual descriptors, an image is often divided into parts first. The simplest way of dividing an image is to use a partition, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using region segmentation algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete object segmentation to obtain semantically meaningful objects (like ball, car, horse). Currently, automatic object segmentation for broad domains of general images is unlikely to succeed.

In this section, we will introduce some widely used techniques for extracting color, texture, shape and spatial relationship from images [7-10].

3.1.1. COLOR

Color Histogram

The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing both the global and local distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle [8].

Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in RGB space, or hue, saturation, and value in HSV space), a histogram, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases.

Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance in many applications. One way to reduce the number of bins is to use the opponent color space which enables the brightness of the histogram to be down sampled. Another way is to use clustering methods to determine the K best colors in a given space for a given set of images. Each of these best colors will be taken as a histogram bin. Since that clustering process takes the color distribution of images over the entire database into consideration, the likelihood of histogram bins in which no or very few pixels fall will be minimized. Another option is to use the bins that have the largest pixel numbers since a small number of histogram bins capture the majority of pixels of an image Such a reduction does not degrade the performance of histogram matching, but may even enhance it since small histogram bins are likely to be noisy [8].

When an image database contains a large number of images, histogram comparison will saturate the discrimination. To solve this problem, the joint histogram technique is introduced. In addition, color histogram does not take the spatial information of pixels into consideration, thus very different images can have similar color distributions. This problem becomes especially acute for large scale databases. To increase discrimination power, several improvements have been proposed to incorporate spatial information. A simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. As introduced above, the division can be as simple as a rectangular partition, or as complex as a region or even object segmentation. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

Color Correlogram

The color correlogram was proposed to characterize not only the color distributions of pixels, but also the spatial correlation of pairs of colors. The first and the second dimension of the three-dimensional histogram are the colors of any pixel pair and the third dimension is their spatial distance. A color correlogram is a table indexed by color pairs, where the k-th entry for (i, j) specifies the probability of finding a pixel of color j at a distance k from a pixel of color i in the image. Let I represent the entire set of image pixels and Ic(i) represent the set of pixels whose colors are c(i). Then, the color correlogram is defined as:

$$\gamma_{i,j}^{(k)} = \Pr_{p_1 \in I_{c(i)}, p_2 \in I} \left[p_2 \in I_{c(j)} \middle| \, |p_1 - p_2| = k \right]$$

where i, j Є {1, 2, …, N}, k Є {1, 2, …, d}, and | p1 – p2 | is the distance between pixels p1 and p2. If we consider all the possible combinations of color pairs the size of the color correlogram will be very large (O (N2d)), therefore a simplified version of the feature called the color autocorrelogram is often used instead. The color autocorrelogram only captures the spatial correlation between identical colors and thus reduces the dimension to O(Nd). Compared to the color histogram and CCV, the color autocorrelogram provides the best retrieval results, but is also the most computational expensive due to its high dimensionality.

3.1.2 TEXTURE

Gabor Filter Features

The Gabor filter has been widely used to extract image features, especially texture features. It is optimal in terms of minimizing the joint uncertainty in space and frequency, and is often used as an orientation and scale tunable edge and line (bar) detector. There have been many approaches proposed to characterize textures of images based on Gabor filters [7][9]. The basic idea of using Gabor filters to extract texture features is as follows.

A two dimensional Gabor function g(x, y) is defined as:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right]$$

Where, σx and σy are the standard deviations of the Guassian envelopes along the x and y direction.

Then a set of Gabor filters can be obtained by appropriate dilations and rotations of g(x, y):

$$g_{mn}(x, y)=a^{-m}\, g(x',y')$$
$$x' = a^{-m} (x \cos\theta + y \sin\theta)$$
$$y' = a^{-m} (-x \cos\theta + y \sin\theta)$$

where $a > 1$, $\theta = n\pi/K$, $n = 0, 1, \ldots, K-1$, and $m = 0, 1, \ldots, S-1$.K and S are the number of orientations and scales. The scale factor a-m is to ensure that energy is independent of m.

Given an image I(x, y), its Gabor transform is defined as:

$$W_{mn}(x,y) = \int I(x,y)\, g_{mn}^{*}(x-x_1, y-y_1)\,dx_1 dy_1$$

Where * indicates the complex conjugate. Then the mean μmn and the standard deviation σmn of the magnitude of Wmn (x, y), i.e., f [μ00,σ00,…,μmn, σ mn, Λ, μs-1k-1, σs-1k-1] can be used to represent the texture feature of a homogenous texture region.

Wavelet Transform Features

Similar to the Gabor filtering, the wavelet transform provides a multi-resolution approach to texture analysis and classification. Wavelet transforms decompose a signal with a family of basis functions ψmn(x) obtained through translation and dilation of a mother wavelet ψ(x), i.e.,

$$\psi_{mn}(x) = 2^{-m/2}\psi(2^{-m}x - n)$$

Where m and n are dilation and translation parameters. A signal f(x) can be represented as:

$$f(x) = \sum_{m,n} c_{mn}\psi_{mn}(x)$$

The computation of the wavelet transforms of a 2D signal involves recursive filtering and sub-sampling. At each level, the signal is decomposed into four frequency sub-bands, LL, LH, HL, and HH, where L denotes low frequency and H denotes high frequency. Two major types of wavelet transforms used for texture analysis are the pyramid-structured wavelet transform (PWT) and the tree-structured wavelets transform (TWT). The PWT recursively decomposes the LL band. However, for some textures the most important information often appears in the middle frequency channels. To overcome this drawback, the TWT decomposes other bands such as LH, HL or HH when needed.

After the decomposition, feature vectors can be constructed using the mean and standard deviation of the energy distribution of each sub-band at each level. For three-level decomposition, PWT results in a feature vector of 3x4x2 components. For TWT, the feature will depend on how sub-bands at each level are decomposed. A fixed decomposition tree can be obtained by sequentially decomposing the LL, LH, and HL bands, and thus results in a feature vector of 52x2 components. Note that in this example, the feature obtained by PWT can be considered as a subset of the feature obtained by TWT. Furthermore, according to the comparison of different wavelet transform features, the particular choice of wavelet filter is not critical for texture analysis.

Pyramid-structure wavelets transform

The pyramid-structure wavelet transform indicate that it recursively decomposes sub signals in the low frequency channels. This method is significant for textures with dominant frequency channels. For this reason, it is mostly suitable for signals consisting of components with information concentrated in lower frequency channels. The pyramid-structured wavelet transform is highly sufficient for the images in which most of its information is exist in lower sub-bands [7][9].

Using the pyramid-structure wavelet transform, the texture image is decomposed into four sub images, as low-low, low-high, high-low and high-high sub-bands. The energy level of each sub-band is calculated. This is first level decomposition. Using the low-low sub-band for further decomposition is done. Decomposition is done up to third level in this project. The reason for this type of decomposition is the assumption that the energy of an image is concentrated in the low-low band.

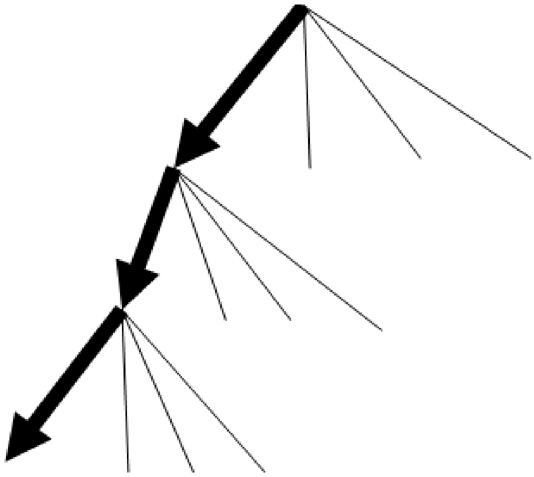

Figure 3. Pyramid structure wavelet transform

IV. IMPLEMETATON & RESULTS

Content-based image retrieval is the task of searching images in databases by analyzing the image contents. Here in the implementation, a simple image retrieval method is presented, based on the color distribution of the images. The user simply provides an "example" image and the search is based upon that example (query by image example).

4.1 Implementation

Method Description

Almost 1000 images have been used for populating the database. For each image a 3-D histogram of its HSV values is computed. At the end of the training stage, all 3D HSV histograms are stored in the same .mat file.

Query

In order to retrieve M (user-defined) query results, the following steps are executed:

The 3D (HSV) histogram of the query image is computed. Then, the number of bins in each direction (i.e., HSV space) is duplicated by means of interpolation.

1. For each image i in the database:
    - Load its histogram Hist(i).
    - Use interpolation for duplicating the number of bins in each direction.
    - For each 3-D hist bin, compute the distance (D) between the hist of the query image and the i-th database image.
    - Keep only distances (D2) for which, the respective hist bins of the query image are larger than a predefined threshold T (let L2 the number of these distances).
    - Use a 2nd threshold: find the distance (D3) values which are smaller than T2, and let L3 be the number of such values.
    - The similarity measure is defined as: S(i) = L2 * average(D3) / (L3^2).
2. Sort the similarity vector and prompt the user with the images that have the M smaller S values.

4.2 Results

Searching the image with the same color along with comparing histograms of relevant images stored in database with the "Example" image given by user as shown below:

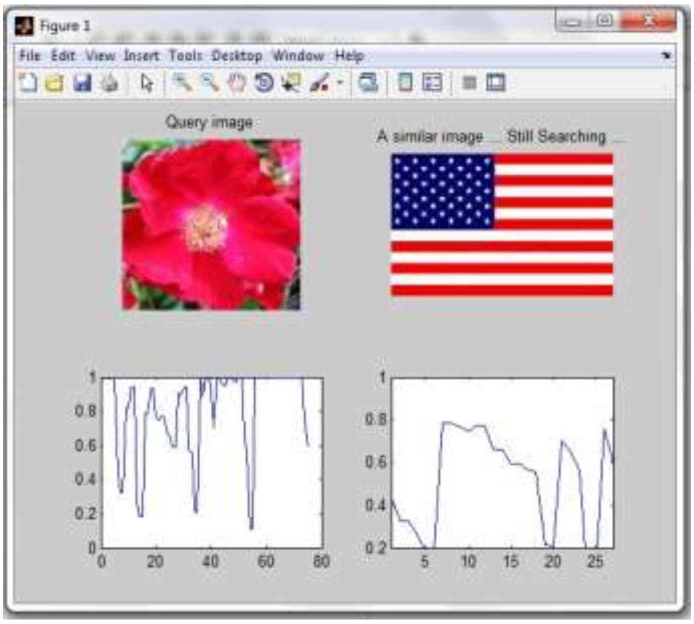

Figure 4. Searching some similar images (based on a pre-defined threshold).

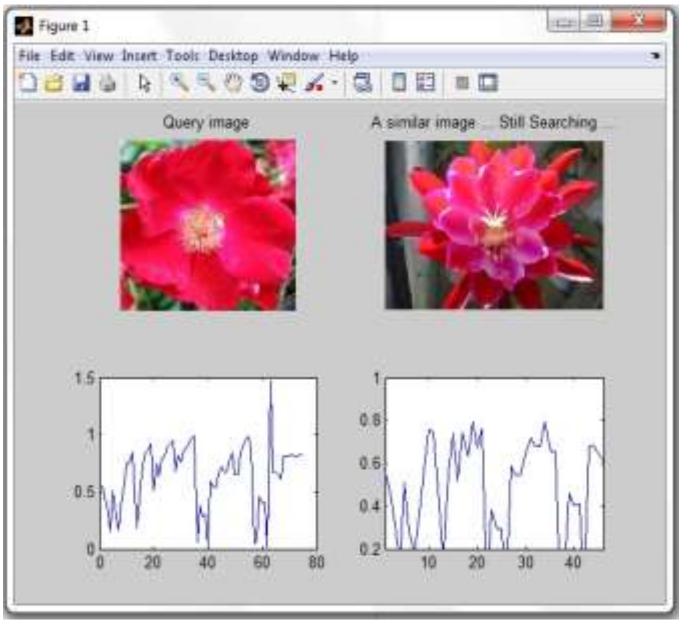

Figure 5. Searching some similar images (based on a pre-defined threshold).

Final output of execution gives the list of the different images which are matched with the given input query image as shown below:

Figure 6. Query image, along with list of closest image.

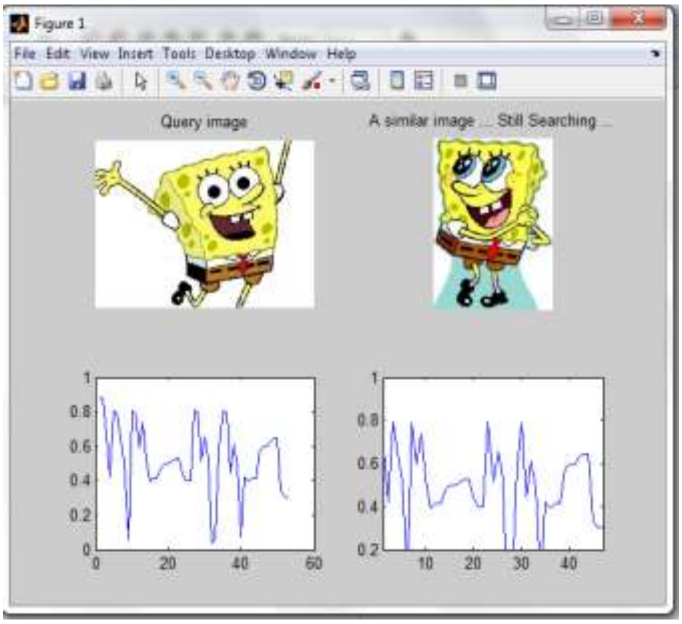

Figure 7. Searching some similar images (based on a pre-defined threshold).

Figure 8. Query image, along with list of closest image.

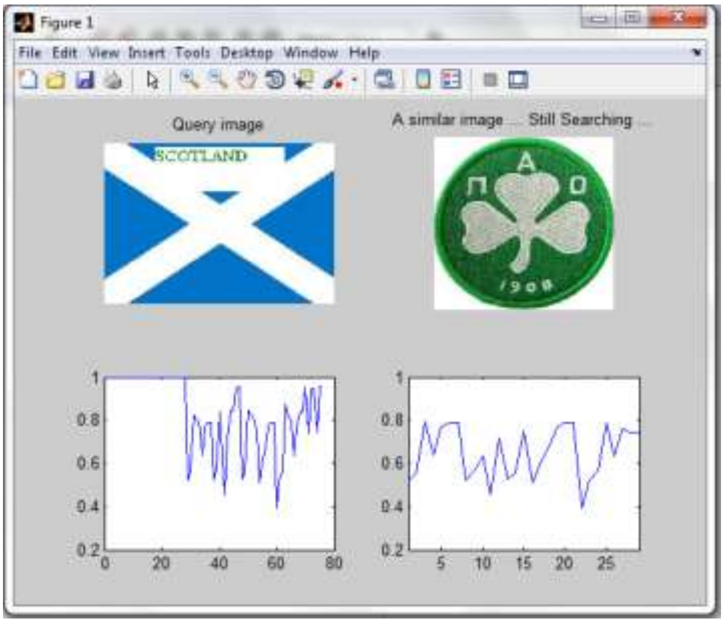

Figure 9. Searching some similar images (based on a pre-defined threshold)

Figure 10. Query image, along with list of closest image.

From above figures of different results we see that result of figure 4 is better than other two examples of figure 7 and figure 9 which gives better result using color histogram matching.

V. CONCLUSION

The most important contribution of this research is the proposed hybrid method combining the advantages of low-level image characteristics extraction with textual description of image semantics. Most systems use color and texture features, few systems use shape feature, and still less use layout features. The retrieval on color usually yields images with similar colors. Retrieval on texture does not always yield images that have clearly the same texture, unless the database contains many images with a dominant texture. Searching on shape gives often surprising results. Apparently the shape features used for matching are not the most effective ones. It is difficult to evaluate how successful content-based image retrieval systems are, in terms of effectiveness, efficiency, and flexibility. Of course there are the notions of precision (the ratio of relevant images to the total number of images retrieved) and recall (the percentage of relevant images among all possible relevant images). From these statements we can come to a conclusion that CBIR systems currently existing are primitive and need some more advanced techniques.

REFERENCES

[1] http://en.wikipedia.org/wiki/Content_based_image_retrieval.
[2] http://www.eecs.berkeley.edu/~cchang/docs/Spie01.pdf.
[3] http://www.jtapac.org/
[4] http://imageprocessing.wordpress.com/2007/12/09/content- Based-image-retrieval-cbir/
[5] http://eprints.utm.my/9466/1/TeeChengSiewFSKSM2008.pdf
[6] Xuejing Dai "Content-Based Image Retrieval Method and its Application to Shoeprint identification", IEEE-2010.
[7] Lidiya Xavier, Thusnavis Bella Mary.I, Newton David Raj.W, "Content Based Image Retrieval Using Textural Features based on Pyramid-Structure Wavelet Transform", IEEE-2011.
[8] Ig-Jae Kim, Je-Ho Lee, Yong-Moo Kwon, Sang-Hui Park, "Content-Based Image Retrieval Method using Color and Shape Features" , Information, Communications and Signal Processing ,ICICS '97 , September 1997.
[9] B.Jyothi, Y.Madhavee Latha, V.S.K.Reddy, "Relevance Feed Back Content Based Image Retrieval Using Multiple Features", IEEE-2010.
[10] An-zen shih, "The Application of Fractal Compression To Content-Based", Machine Learning and Cybernetics, Baoding, 12-15 July 2009.