Volume 6, Issue 2, February 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Study of Content Based Image Retrieval Techniques

[1]Graceline Abigail P., [2]Dr. Josephine Prem Kumar
[1] IV Year B.Tech. (IT), Thiagarajar College of Engineering, Madurai, India
[2] Professor – ISE, East Point College of Engineering and Technology, Bangalore, India

Abstract: Content Based Image Retrieval (CBIR) is one of the most popular and interesting research areas because of the proliferation of video and image data in digital form. Fast and accurate retrieval of image from large databases is an important problem that needs to be addressed. Basically, CBIR is on developing technologies to bridge the semantic gap that currently prevents wide-deployment of image content-based search engines. Image search engines currently in use such as Google Images and Yahoo! Image search, are based on textual annotation of images. Here, images are manually annotated with keywords, which depends totally on the person's perception, and then retrieved using text-based search methods. This method is both time-consuming and prone to errors. Hence, such search engines result in retrieving many non-relevant images. To overcome such drawbacks of text based image retrieval, CBIR is introduced where the visual content of an image, such as color, texture and shape, is extracted automatically; the retrieval of images is totally dependent on these features. In this paper, we will try to study the basic block diagram of CBIR and the different techniques currently being used, such as color histogram and edge histogram.

Keywords: Image Retrieval, CBIR, color feature, color histogram, edge histogram.

I. INTRODUCTION

Content Based Image Retrieval (CBIR) is one of the most popular and interesting research areas because of the proliferation of video and image data in digital form. Increased bandwidth availability to access the internet has led to a huge amount of image databases being added every minute. Hence, fast and accurate retrieval of images from large databases is an important problem that needs to be addressed.

II. IMAGE RETRIEVAL METHODS

Image retrieval consists of two different methods:

1. Text-Based image retrieval
2. Content-Based image retrieval

The Text-Based image retrieval method is based on keyword and it is easy to be implemented. The images are manually annotated with keywords, and then retrieved using text-based search methods. Textual annotation can be thought of as an instantiation of mental image; hence it depends totally on the person's perception. This method is both time-consuming and prone to errors. Hence, such search engines result in retrieving many non-relevant images. For example, in a search engine if the user enters text as Plane, it can be a plane surface or an aeroplane. Also, the fact that two visually different images can convey the same concept and different concepts may be present in an image, brings about a gap between image retrieval by concept and retrieval by content.

Content-Based image retrieval (CBIR) is also known as query by image content. Content-based means the search will analyze the actual contents of image. In CBIR, the visual content of an image is extracted automatically. There are many features that make an image; but four of them are considered to be main features i.e. colour, texture, shape and spatial properties. The retrieval of images is totally dependent on these features. However, spatial properties are implicitly taken into account. So, the main features to consider are colour, texture and shape.

III. CONTENT-BASED IMAGE RETRIEVAL SYSTEM

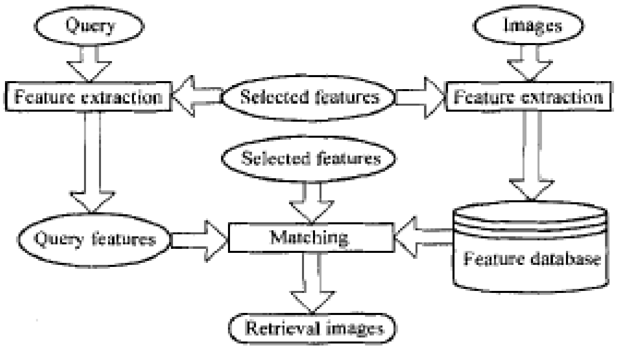


Fig. 1 Block diagram for CBIR system

In a typical CBIR process, the features from each image are extracted and stored in the database efficiently. To retrieve the images, extract the corresponding features from the query image and search the image database to identify the images that are similar to it; and returns the results[2] [4].

Thus, a typical CBIR system (Fig.1) consists of three major components and the variations of them depend on features used.

i. Feature extraction – Analyze raw image data to extract feature specific information; plays an important role to support for efficient and fast retrieval of similar images from image databases.
ii. Feature storage – Provide efficient storage for the extracted information, also help to improve searching speed.
iii. Similarity measure – Measure the difference between images for determining the relevance between images, yielding a result that is visually similar.

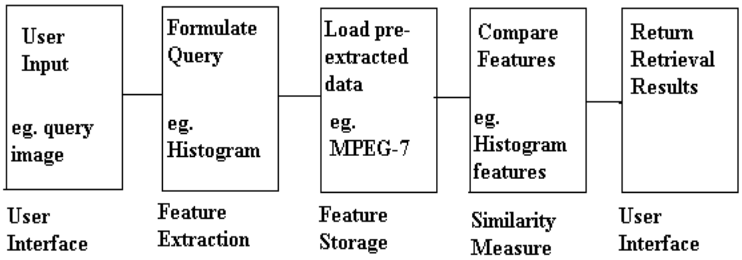


Fig. 2 Flow of a typical CBIR process

Fig. 2 describes the flow of a typical CBIR process.

In the whole CBIR process, feature extraction is crucial [3]. Feature extraction and similarity measure are very dependent on the features used. In each feature, there would be more than one representation. Among these representations, histogram is the most commonly used technique to describe features. In practice, a feature or a combination of features is often used to search for images. The relevance between a query image and any target image is ranked according to a similarity measure computed from the visual features.

IV. COLOR MODEL

One of the most important features that make possible the recognition of images by humans is color. Color is a property that depends on the reflection of light to the eye and the processing of that information in the brain. Color not only adds beauty to objects but also more information [6]. Hence, it s a powerful tool in CBIR.

There are many color models to express color such as the RGB color model, the YUV color model and the HSV color model. The HSV model is most consistent as it includes the human visual model [5]. H represents color hue and it is the wavelength of the light reflected from an object or throughout the object; S represents color saturation i.e. how much white is added to the color; V represents vale (brightness) i.e. the degree of color shading. But, the computer can identify only the RGB color components of an image, in which R represents the red component, G represents the green component and B the blue component. Therefore, the following formula is used for image conversion from RGB color space to HSV color space.

$$h' = \begin{cases} \dfrac{(g-b)}{\delta} & if \quad r = \max \\ \dfrac{2+(b-r)}{\delta} & if \quad g = \max \\ \dfrac{4+(r-g)}{\delta} & if \quad b = \max \end{cases} \qquad h = h' * 60 \qquad s = \frac{max - min}{max} \qquad v = \frac{r+g+b}{3}$$

$$\max = MAX(r,g,b),\ \min = MIN(r,g,b),$$
$$\delta = \max - \min,$$
$$h \in [0,360],\ (s,v) \in [0,1.0]$$

V. COLOR QUANTIZATION

For a true color image, the number of the kind of the colors up to 2^{24} = 16777216, so directly extract color feature from true color will lead to large computation. In order to reduce the computation, without a significant reduction in image quality, some representative color is extracted to represent image, thereby to reduce storage space and enhance the purpose of processing speed[7]. The color regions are perceptually distinguishable to some extent. The human eye cannot detect small color different and may perceive these very similar colors as the same color. This leads to the *quantization* of color, which means that some pre-specified colors will be present on the image and each color is mapped to some of these pre-specified colors. One obvious consequence of this is that each color space may require different levels of quantized colors, which is nothing but a different quantization scheme. There are 36, 72 and 256 quantiative color quantization and we can depend on different need to select different levels of quantitative methods.

In Fig.3, the effect of color quantization is illustrated. Fig. 3(a) is the original image with *RGB* color space and Fig.3(b) is the image produced after transformation into *HSV* color space and quantization.

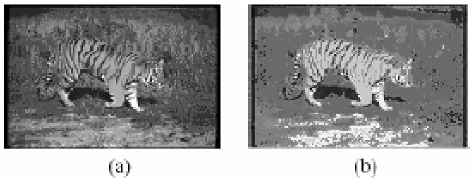


Fig.3 Transformation and Quantization of tiger image

(a) Original image., (b) Image produced by applying RGB to HSV color transformation and Quantization

VI. COLOR HISTOGRAM

A color histogram is a type of bar graph, where each bar represents a particular color of the color space being used. We can get a color histogram of an image in the RGB or HSV color space. The bars in a color histogram are referred to as bins and they represent the x-axis. The number of bins depends on the number of colors there are in an image. The y-axis denotes the number of pixels there are in eaczzh bin. In other words how many pixels in an image are of a particular color [8].

However, color histogram has its own drawbacks, such as the color histograms of different images may be the same: there are two unrelated images in Fig.4, but they are the same as the color histogram in Fig.5.

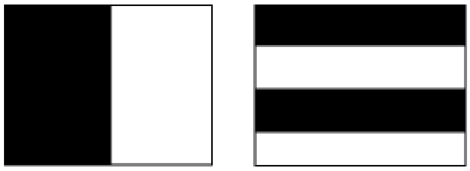


Fig.4 Two unrelated images

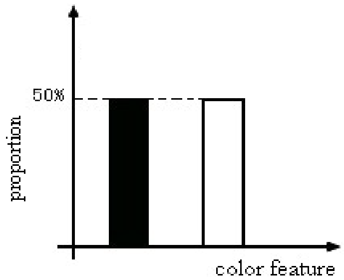
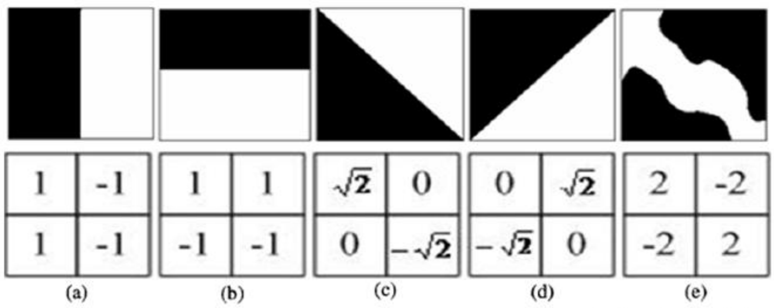


Fig.5 Color Histogram

VII. EDGE HISTOGRAM

The edge histogram descriptor captures the spatial distribution of edges. The distribution of edges is a good texture signature that is useful for image to image matching even when the underlying texture **is** not homogeneous. The computation of this descriptor is fairly straightforward [9]. A given image is first sub-divided into 4x4 sub-images, and local edge histograms for each of these sub-images are computed. Edges are broadly grouped into five categories: vertical, horizontal, 45° diagonal, **135°** diagonal, and isotropic (non orientation specific) (Fig.6**).** Thus, each local histogram has five bins corresponding to the above five categories. The image partitioned into 16 **sub**images results in 80 bins.

To compute the edge histograms, each of the 16 sub-images is further subdivided into image blocks. The size of these image blocks scale with the image size and is assumed to be a power of ***2.*** The number of image blocks per sub-image is kept constant, independent of the original image dimensions, by scaling their size appropriately.



(a) Vertical edge (b) Horizontal edge (c) 45 degree edge (d) 135 degree edge (e) Non-directional edge

Fig.6 Five Types of Edges

A simple edge detector is then applied to each of the macro-block, treating the macro-block as a 2x2 pixel image. The edge-detector operators include four directional selective detectors and one isotropic operator. Those image blocks whose edge strengths exceed a certain minimum threshold are used in computing the histogram. Thus, for an image block, we can compute five edge strengths, one for each of the five filters from Fig. 6**.** If the maximum of these edge strengths exceed a certain preset threshold, then the corresponding image block is considered to be an edge block. An edge block contributes to the edge histogram bins.

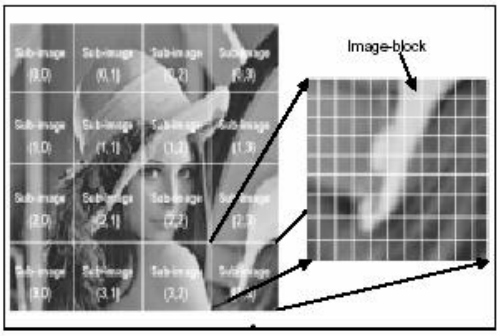


Fig.7 Sub image and sub block

As a result, as shown in Fig.7, each local histogram contains 5 bins. Each bin corresponds to one of 5 edge types. Since there are 16 sub-images in the image, a total of 5×16=80 histogram bins is required.

VIII. SIMILARITY MEASURE

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. The histogram euclidean distance 'is generally used for similarity measure.

Let H_1, and H_2 represent two histograms. The Euclidean distance between the histograms H_1 and H_2 can be computed as:

$$d_{HE}(H_1,H_2)=\sqrt{\sum_{x\in X,\, y\in Y,\, z\in Z}\left(H_1(x,y,z)-H_2(x,y,z)\right)^2}$$

This distance is the L^2 -norm of the different vector.

IX. EXPERIMENTAL RESULTS

Performance evaluation experiment based on image features **is** used to evaluate the retrieval performance of different features on the same image database, which uses the same similarity measure. The experiment selects color histogram and edge histogram methods[10].


Fig. 8 Example Input Image

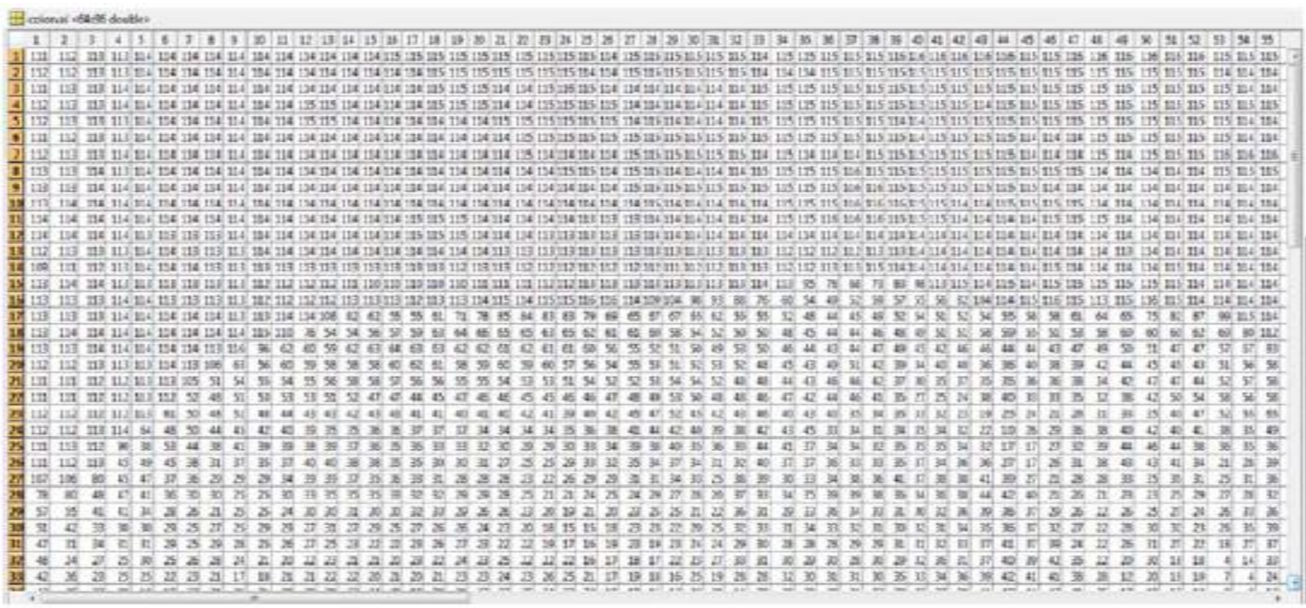


Fig. 9 Color Value for the image in Fig. 8

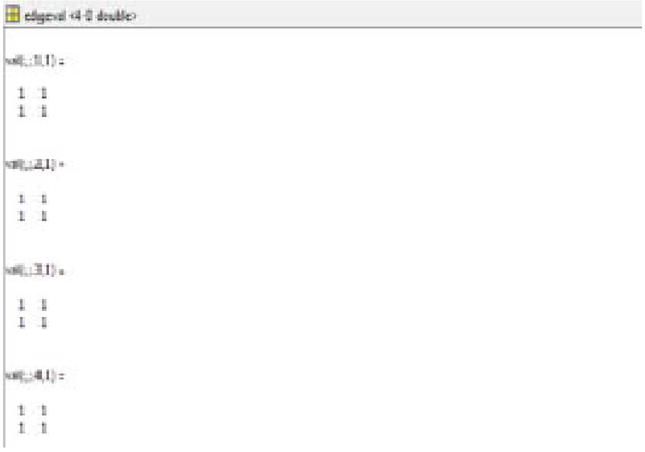


Fig. 10 Edge Value for the image in Fig. 8

Fig. 8, 9, 10 show sample experimental results that had been obtained.

Fig. 11 Sample Query for Color Histogram method

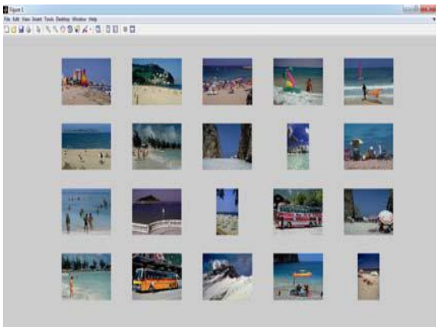


Fig. 12 Sample output for Fig. 11

Fig. 13 Sample Query for Edge Histogram method

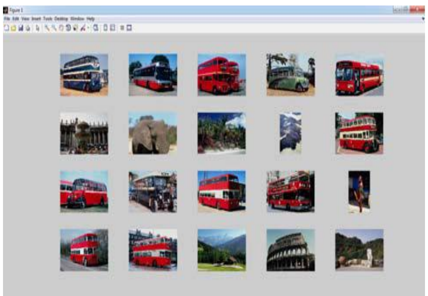


Fig. 14 Sample output for Fig. 13

X. CONCLUSIONS

The traditional image retrieval mainly depends on color, texture and shape. For these basic visual features are just parts of image information, the retrieval results are not so perfect. After the absolute analysis of the results obtained by each method following conclusions can be drawn. When only colour is considered as retrieval parameter in CBIR gives only 62.5% of average retrieval efficiency. Similarly when only edge features are considered as retrieval parameter the average retrieval efficiency obtained is only 68.75%.

This shows that only edge features or only colour features are not sufficient to describe an image. But there should be considerable increase in retrieval efficiency when both colour and texture features are combined for CBIR.

In addition, the similarity measures between visual features do not necessarily match human perception. Hence different similarity measures are also to be studied.

XI. APPLICATIONS

The CBIR technology has been used in several applications such as fingerprint identification, biodiversity information systems, digital libraries, crime prevention, medicine, historical research, among others.

REFERENCES

[1] Zhang Xinjun, “Research of image retrieval based on color features” [D] Liaoning Technical University. 2006, 9(2), pp. 42-50.
[2] Rao Juan, “Intelligent image retrieval technique based on content” [D] Lanzhou University of Technology. 2006, pp 32-64.
[3] Ye Yu-guang, “Research of image retrievalbased on fusing with multi-character”, Hua Qiao University, 2007, pp. 14-36.
[4] A.K. Jain, J.E. Lee, R. Jin and N. Gregg, “Content based image retrieval: an Application to tattoo images”, IEEE International Conference on Image Processing, pp. 2745-2748, November 2009.
[5] Zeng Tai-sheng, “Study on the image retrieval based on color and texture and feedback”, Hua Qiao University, 2006, pp.25-34.
[6] Manesh Kokare, B.N. Chatterji and P.K. Biswas, “A survey on current content based image retrieval methods”, IETE Journal of Research, Vol. 48, No. 3 and 4, May-Aug 2002.
[7] Hafner J, Sawhney H S , “Efficient color histogram indexing for quadratic form distance functions”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Intelligence,1995,17(7): pp.729-736.
[8] Swain MJ and Balland DH. “Color indexing of Computer Vision” [J], 1991-7(1):11-32.
[9] Manjunath B S, Jens-Rainer Ohm, Vasudevan V, et al. Color and texture descriptors, IEEE Transactions on circuits and systems for video technology, 2001, 1 1(6), 703-715
[10] SHI Dongcheng, XU Lan, HAN Ungyan, “Image retrieval using both color and texture features”, The Journal of China Universities of Posts and Telecommunications, Vol.14, Suppl, Oct. 2007.

