***I.J. Image, Graphics and Signal Processing*, 2012, 4, 42-48**
Published Online May 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2012.04.06

An Efficient and Generalized approach for Content Based Image Retrieval in MatLab.

Shriram K V [1]
[1] Research Scholar, PRIST University, Tanjore, India.
E-mail: shriramkv@gmail.com

Dr.P.L.K Priyadarsini [2]
[2] Associate Professor, CSE Department, PRIST University, Tanjore, India.
E-mail: priya.ayyagari@yahoo.co.in

Subashri V [3]
[3] M.Tech CSE-II Year, BS Abdur Rahman University, Chennai, India.
E-mail: vrsuba@gmail.com

Abstract— There is a serious flaw in existing image search engines, since they basically work under the influence of keywords. Retrieving images based on the keywords is not only inappropriate, but also time consuming. Content Based Image Retrieval (CBIR) is still a research area, which aims to retrieve images based on the content of the query image. In this paper we have proposed a CBIR based image retrieval system, which analyses innate properties of an image such as, the color, texture and the entropy factor, for efficient and meaningful image retrieval. The initial step is to retrieve images based on the color combination of the query image, which is followed by the texture based retrieval and finally, based on the entropy of the images, the results are filtered. The proposed system results in retrieving the images from the database which are similar to the query image. Entropy based image retrieval proved to be quite useful in filtering the irrelevant images thereby improving the efficiency of the system.

Index Terms—Image Processing, CBIR, Histogram, Wavelets, Quadratic distance, Euclidean distance, Entropy

I. Introduction

Image processing is a field which faces drastic changes and increased users day by day. One of the widely used applications of image processing is the content based image retrieval, which aims to retrieve similar kinds of images from the database with respect to the query image. Most of the existing CBIR based search engines are keyword-dependant. If the keywords are not relevant to the images, then the very purpose of retrieving similar kinds of images is lost! In order to overcome this problem, CBIR based on the semantics of the images came into existence.

Color based image retrieval was the basic strategy in retrieving images in the early days [4]. Histograms of the images were the tool to analyze the color composition in any image. But, image retrieval based on only the color combo is not sufficient since this approach won't consider the content of the image at all thereby not achieving the basic objective. The next generation CBIR systems considered color and texture of the images before retrieving [5]. This approach too proved to be insufficient, since the results had irrelevant images. To improve the efficiency of CBIR systems, relevance feedback based image retrieval was experimented [1]. But, relevance feedback based approach consumed much of the users' time making them train the system with positive and negative images, (those images in the initial results that are relevant to query images are positive images, those do not have any relevancy are termed as negative images). Since it is too complex to follow the steps in training the system, this approach too did not win the contest of CBIR systems. In this paper, in addition to the above mentioned color and texture properties of an image, we have considered a phenomenal property of an image, the entropy, which can analyze the randomness factor of an image. By combining these factors, we could achieve the retrieval efficiency up to 96%. Figure 1 shows the basic design of our CBIR system. We retrieve images in three stages, in the initial stage, we retrieved images with respect to the color, if the user is satisfied with the retrieval results, we stopped there itself, thereby not giving much work to the system. If not, the retrieval was carried out based on the texture of the images (we calculated the overall pixel intensity of the images.). If the user is still not satisfied with the results, we calculated the entropy of images before comparing them.

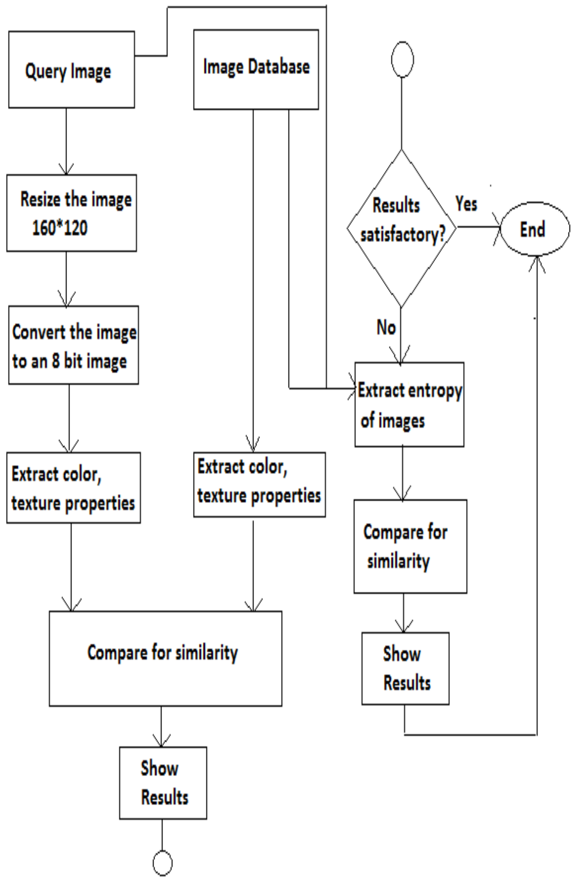


Figure 1 Design Diagram

II. Detailed analysis of the design proposed.

The search begins with providing an image as the 'query image'. The objective is to retrieve images which are similar to the query image. The features that were considered initially include, color, texture and entropy.

A. Color

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. We use color every day to tell the difference between objects, places, and the time of day. Usually colors are defined in three dimensional color spaces. These could either be RGB (Red, Green, and Blue), HSV (Hue, Saturation, and Value) or HSB (Hue, Saturation, and Brightness). Color combination can be easily visualized using histograms. A color histogram is a type of bar graph, where each bar represents a particular color of the color space being used.

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 that are in an image. The y-axis denotes the number of pixels there are in each bin. In other words how many pixels in an image are of a particular color. A sample histogram for an image has been shown in the Figure 2.

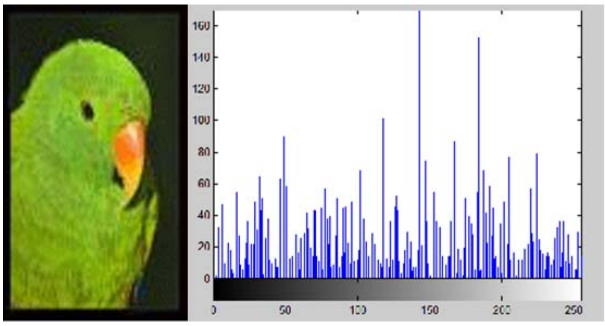


Figure 2 Sample image and its histogram

Histogram is generated for the query image and for all the images in the image database. Then quadratic distance between them is found out using the formula[13],

$$\mathbf{d^2(Q,I) = (H_Q - H_I)^t A(H_Q - H_I)} \text{———} \mathbf{(1)}$$

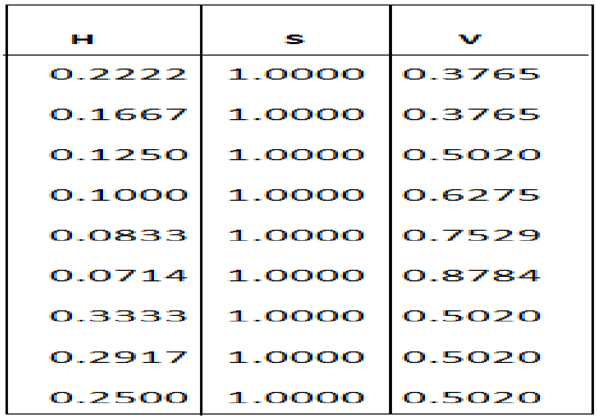
Where, d is the measured quadratic distance between the image histograms, Q and I denote the Queried image and the Image in the database respectively. H_Q and H_I are the Histograms of the query image and the database image respectively. A denotes the 'matrix of similarity' which is calculated using the formula [13],

$$\mathbf{A_{(i,j)}=1-1/\sqrt{5}[(v_q-v_i)+(s_q\cos(h_q)-s_i\cos(h_i))^2+ \quad s_q\sin(h_q)-s_i\sin(h_i))^2]^{1/2}} \qquad \text{------------(2)}$$

Where, A is the matrix of similarity which has 3 columns each representing Hue (h), Saturation(s) and the Value (v) of the image under consideration. Using (1) and (2), similarity between the images with respect to the histograms are found out and, lesser the quadratic distance value, more will be the similarity.

Figure 3 shows a sample query image, and its corresponding HSV matrix (a portion) is tabulated in Table 1.

Figure 3 : Sample Query Image



H	S	V
0.2222	1.0000	0.3765
0.1667	1.0000	0.3765
0.1250	1.0000	0.5020
0.1000	1.0000	0.6275
0.0833	1.0000	0.7529
0.0714	1.0000	0.8784
0.3333	1.0000	0.5020
0.2917	1.0000	0.5020
0.2500	1.0000	0.5020

Table 1: HSV values of Figure 3

Every image' HSV matrix as shown in Table 1 is used in (2) to calculate the quadratic distance value.

B. Texture

Comparing images based on color alone will not be sufficient for efficient retrieval. Hence, another property called, 'texture' is taken into account. Texture describes the physical composition of the picture. We have made use of the wavelet transform technique to analyze the texture of the image. Wavelet is a small wave and wavelet transformation is the process of converting a signal into a series of wavelets. This technique was very helpful to obtain coarser information from the image that is not readily available in the raw image and helped in analyzing the image and identifying patterns on it. The main objective of the wavelet transformation was to calculate the pixel intensity of the images, based on which, the further comparisons can be made. Since MatLab has built-in functions for wavelet transformation; it was very easier for us to carry out our work.

We first decomposed the image into 4 sub-bands (low-low, low-high, high-low, high-high bands) each of different frequency. We observed that, the frequency was concentrated much in the low-low sub band and used the same for further decompositions. We considered almost all kinds of wavelet methods available for decomposition. Of those, we found that, Daubechies wavelet proved to provide us good results. In particular, we incorporated 'db10' for our testing purposes. The following Figure 4 shows the wavelet function as a graph.

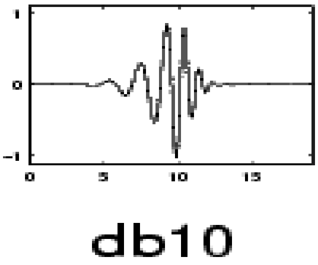


Figure 4 *Daubechies* wavelet

The next step is to obtain energies of each sub band using the formula [5],

$$E = \frac{1}{MN}\sum_{i=1}^{m}\sum_{j=1}^{n}|X(i,j)| \text{ ———— } (3)$$

Where, m and n indicate the row and column of the matrix of the decomposed image. Step 3: repeat Step 1 and 2 for every image in the database and find the Euclidean distance between the query image and every image in the database using the formula [5],

$$D_i = \sum_{k=1}^{k}(x_k - y_{i,k})^2 \text{ ———— } (4)$$

Where Di denotes the Euclidean distance between the query image and ith image in the database. [Note: For every image i in the database, k set of energies have to be calculated. (yi, k)] Xk is the energy level of the query image. Finally we sorted the distance values in the increasing order and displayed the top most images.

C. *Entropy*

Entropy is a factor that is used to characterize the randomness of an image. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy can be defined as follows (13),

$$-\text{sum}(p.*\text{log2}(p)) \text{ ——— } (5)$$

The 'p' in the above term defines the histogram count of the image being processed. We have subtracted the entropy values of query image and all the images in the database. Our foremost job was to choose the entropy difference threshold for choosing the most relevant images. We kept the threshold as 0.59. Figure 5 shows a pair of images and their corresponding entropy values.

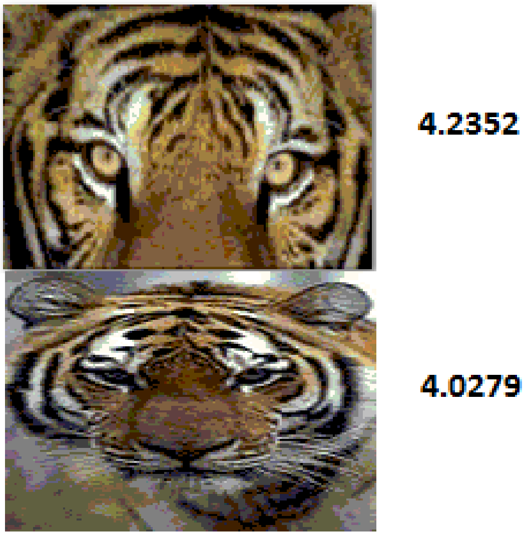


Figure 5 Images and their entropy

III. Relevance feedback based image retrieval

Most of the currently available CBIR systems require the end user to provide feedback to the application, in the sense, the end user is expected to train the CBIR system with some positive images which are visually similar to the query image and, some negative images which are in no way related to the query image. This approach, though seems to be beneficial for the application to study the query image in detail, it consumes much of the user's time in making them interact with the application in a long duration. This will ultimately make the end user tired of following the process, which is not the expected response. Hence, we have excluded the relevance feedback option, and introduced the entropy based retrieval.

IV. Image conversion

Before uploading an image into the image database, we had to check its specifications. If the image doesn’t correspond to our requirement of 256bit, we had to convert it into an image of bit depth equal to 8. We used rgb2ind() method for achieving this. One more important prerequisite to deal with images in MatLab is the image size. We chose the dimension 160*120. We used the built-in function imresize() to automatically convert the images before uploading into the database.

V. Execution and Results

The database had nearly 500 images all of type 256 bit ‘.bmp’s, and size 160*120; Figure 4 shows a portion of database.

Figure 6 Images in the database

Figure 5 shows the query image as follows,

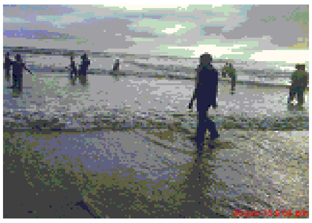


Figure 7 Query image

The above image is processed with respect to color. Following steps hold good.

1) Histogram will be generated first using imhist () method. Figure 8 shows the snapshot of the same.

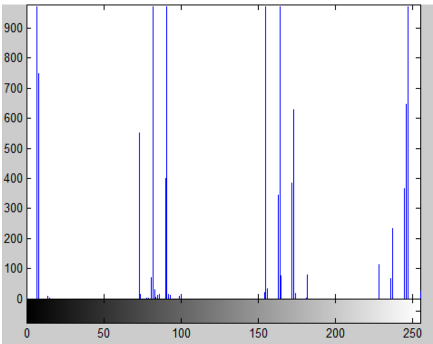


Figure 8 Histogram generated for Figure 7.

2) Step 1 is repeated for all the images in the database images.

3) Quadratic distance is found between each pair of images.

4) Distance value is sorted. Top most results are displayed.

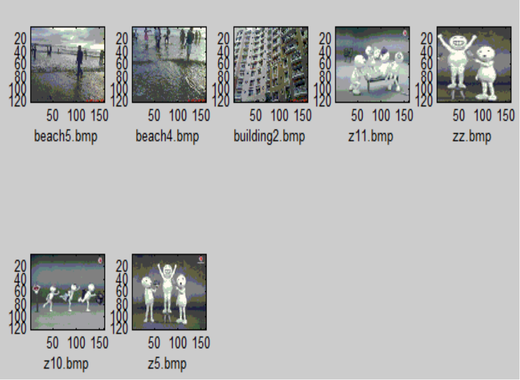


Figure 9 Sample Output for color search

5) Following graph shows the quadratic distance between images.

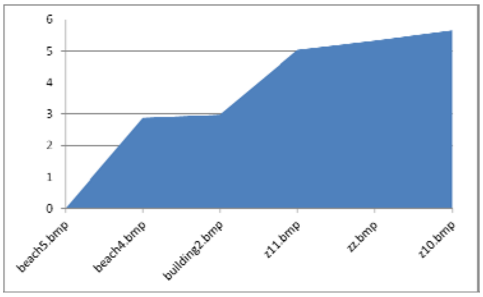


Figure 10 Graph of quadratic distance between images.

6) In the texture portion of the work, the query image is decomposed into 4 sub bands for 6

times (6 levels of decomposition for getting coarser information). So also the images in the database are decomposed. Figure 11 shows a sample decomposed image.

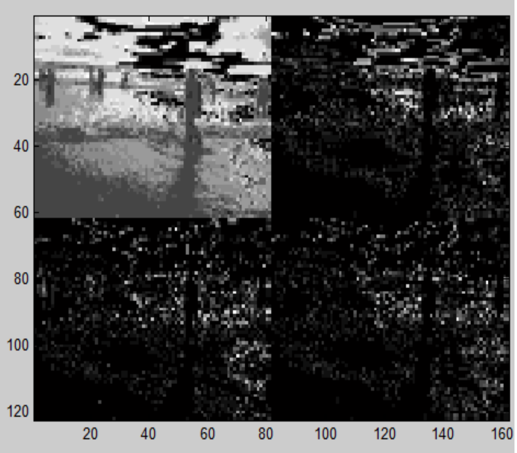


Figure 11 Decomposed image.

7) Finally, the energy (the pixel intensity) of each decomposed image is calculated using the formula (3). A graph showing the Euclidean distance between the images is shown in Figure 12.

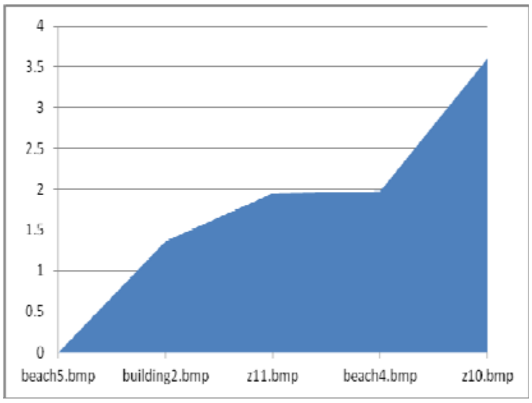


Figure12 Euclidean distance between image.

8) Final step is to find the Euclidean distance between query image and every image in the database using (4). Figure 13 shows the sample output.

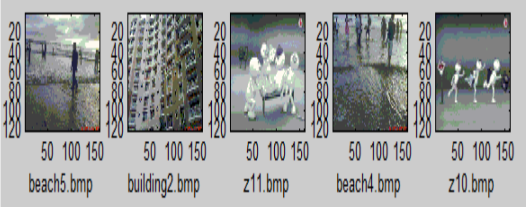


Figure13 Sample output for texture search.

9) If the user is not satisfied with the results, he/she can opt for entropy based image retrieval. There, only exactly matching images will be displayed. Figure 14 shows the entropy based image retrieval.

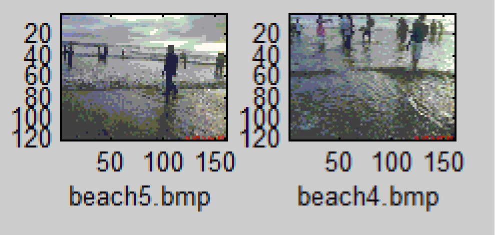


Figure 14 Sample output for entropy based search.

As it can be seen from the execution results, color based retrieval focuses only on color factor. The efficiency is less than 50%, whereas in texture based image retrieval, it was improved to 75 %. Finally, in entropy based image retrieval, the efficiency is nearly 95%.

VI. Future Work

We have planned to incorporate the text based image retrieval into the current work for even more improving the retrieval efficiency.

Acknowledgment

I wish to thank my guide and university for providing the consistent support for getting this project going on.

References

[1] Ja-Hwung Su, Wei-Jyun Huang, Philip S. Yu, and Vincent S. Tseng, "Efficient Relevance Feedback for Content-Based Image Retrieval by Mining User Navigation Patterns" IEEE transactions on knowledge and data engineering, vol. 23, no. 3, march 2011.

[2] Liana Stanescu, IADIS International Conference on Applied Computing, 2005 "on-line software system for content-based visual query of a color medical imagery."

[3] Neetesh Gupta, R.K.Singh, “A New Approach for CBIR Feedback based Image Classifier”, International Journal of Computer Applications (0975 – 8887) Volume 14– No.4, January 2011.

[4] P.S.Suhasini, K.Sri Rama Krishna, and I. V. Murali Krishna, “CBIR using color histogram processing” Journal of Theoretical and Applied Information Technology, 2005-2009.

[5] Rahul Metha, Nishkhol Mishra, Sanjeev Sharm, “Color – Texture based image retrieval system”, International Journal of Computer Applications (0975 – 8887), Volume 24 – No. 5 June 2011.

[6] Satya Sai Prakash, RMD. Sundaram. Combining Novel features for Content Based Image Retrieval. *IEEE XPLORE DIGITAL LIBRARY pages: 373 – 376, 12 November 2007.*

[7] Wasim Khan, Shiv Kumar. Neetesh Gupta, Nilofar Khan, “A Proposed Method for Image Retrieval using Histogram values and Texture Descriptor Analysis” , International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-I Issue-II, May 2011.

[8] Gonzalez, R.C., Woods, R.E, “Digital Image Processing” 2nd Ed., Prentice Hall.

[9] MatLab Manual.

[10]http://pages.cs.wisc.edu/~beechung/docs/papers.html

[11]http://www.ee.columbia.edu/~jrsmith/html/pubs/tatfcir/node22.html

[12] A.M.W. Smeulders, M. Worring, S. Santini, A.Gupta, and R. Jain, “Content-based image retrieval at the end of early years, “IEEE Trans. On Pattern Analysis and machine intelligence, vol. 22, no. 12, december 2000.

[13]Kondekar V. H., Kolkure V. S., Kore S.N. “Image Retrieval Techniques based on Image Features: A State of Art approach for **CBIR**”, International Journal of Computer Science and Information Security,Vol. 7, No. 1, 2010.