Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Content Based Image Retrieval Using Color, Texture and Shape Features

Aruna Verma
Department of Computer Science and Application
DAV Institute of Management,
Faridabad, India

Deepti Sharma
Department of Computer Science and Engineering
Advanced Institute of Technology,
Palwal, India

Abstract---- Content-Based Image Retrieval (CBIR) allows to automatically extracting target images according to objective visual contents of the image itself. Representation of visual features and similarity match are important issues in CBIR. Color, texture and shape information have been the primitive image descriptors in content-based image retrieval systems. This paper presents a novel framework for combining all the three i.e. color, texture and shape information, and achieve higher retrieval efficiency. The combination of the color, texture and shape features provide a robust feature set for image retrieval. The experimental results demonstrate the efficacy of the method. In this paper an efficient CBIR method is proposed by exploit the wavelets, which represent the visual feature. We use Haar wavelet to decompose color images into multilevel scale and wavelet coefficients, with which we perform image feature extraction and similarity match. Two images are then considered similar if their feature vectors lay close in the feature space. The features that are extracted usually fall in three general Categories color, shape and texture.

Keywords: CBIR, Wavelet, Histogram, Gabor.

I. INTRODUCTION

Content-Based Image Retrieval (CBIR) is the process of retrieving desired images from huge databases based on extracted features from the image themselves. Features of an image are extracted and analyzed by means of computer processing [1]. CBIR provide access of multimedia databases that deal with text, audio, video and image data, which could provide us with enormous amount of information. Many commercial and research CBIR systems have been built and developed (e.g.: QBIC) [2]. CBIR is a technique, which uses visual contents (features), to search images from large-scale image databases according to users' requests in the form of a query image. The usual features like color, texture, and edge density (shape) extracted with the help of feature extraction algorithm. The CBIR is used to operate on the query image and then obtain the output relevant to that query Image.[11] In this paper the low level feature extraction is being done. In this paper we have discussed the texture features, the color and shape features. The [13] texture feature extraction is done using GLCM, histogram is used for color feature extraction and different shape features are extracted from the query image. From the output obtained, it is found that the combination of low-level features provides the better results in image retrieval. The CBIR is the technique to map each of the images in the database to a feature space and then retrieve based on the feature of the query image. The main challenge of CBIR is to bridge the semantic gap between low-level content descriptors with high-level concepts (like faces, flowers, architectures etc). CBIR, systems have gained popularity because of their objective means of assessing image content. Textual annotation, in contrast, is plagued by inconsistencies of the human annotator. Perhaps more important, in many domains, text cannot accurately capture the visual attributes of an image in the user's mind [5]. In CBIR, the query is an image that the user presents as an example of what he or she is looking for. The goal of CBIR is to retrieve images that are visually similar to the query image. To this end, a CBIR system applies image processing and computer vision algorithms to extract a vector of features from each of the database images. CBIR systems rest on the assumption that points in this space that are proximal to the Point represented by the query image's vector should correspond to the feature vectors of images that are visually similar to the query. There is a problem of impetus for what is known as relevance feedback .

2. Structure of Proposed CBIR system

Our proposed CBIR algorithm is based on decomposition of the database images using Haar wavelets in the offline as well as in online for query image. With resulting coefficients we extract the features and perform highly efficient image matching. In the database we used huge range of images with different colors. For the given a query image its feature vectors are computed. If the distance between features of the query image and images in the database is small, the corresponding image in the database is to be considered as a match to the query.

The search is usually based on similarity rather than on exact match and the retrieval results are then ranked accordingly to [6] a similarity index. Conventional databases allow for textual searches on Meta data only. Edges convey essential information to a picture and therefore can be applied to image retrieval. The edge histogram descriptor captures the spatial distribution of edges. This model expects the input as Query by Example (QBE) and any

© 2014, IJARCSSE All Rights Reserved

combination of features can be selected for retrieval. The focus is to build a universal CBIR system using low level features. These are mean, median, and standard deviation of Red, Green, and Blue channels of color histograms. Then the texture features such as contrast, energy, correlation, and homogeneity are retrieved.

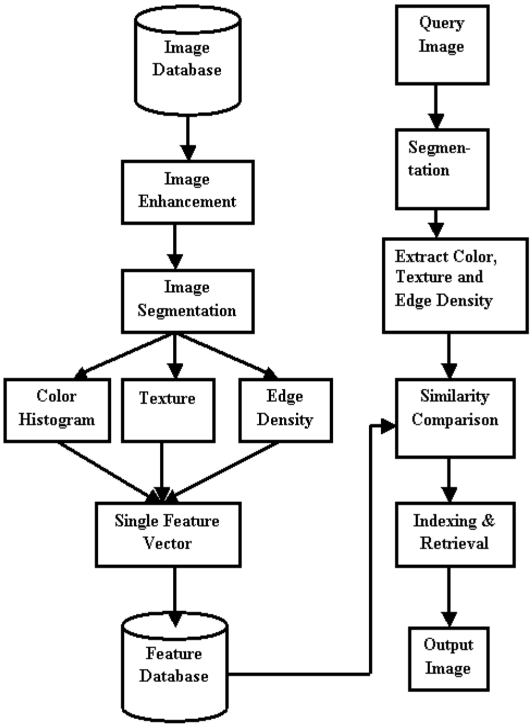


Fig1: Block Diagram of CBIR

3. Image Segmentation

We used wavelet approaches for color image segmentation, namely Haar wavelets. The resulting decomposition coefficients are employed to perform image feature extraction and similarity match by virtue of F-norm theory.

3.1 Haar Wavelet Transform

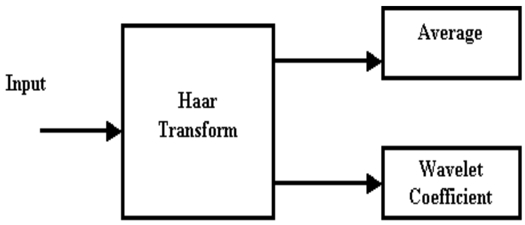


Fig2: Haar Wavelet Transform

If a data set S0, S1. …SN-1 contains N elements; there will be N/2 averages and N/2 wavelet coefficient values. The averages are stored in the upper half of the N element array and the coefficients are stored in the lower half as shown in the Fig.2. The averages become the input for the next step in the wavelet calculation, where for iteration i+1, Ni+1 = Ni/2. The recursive iterations continue until a single average and a single coefficient are calculated. This replaces the original data set of N elements with an average, followed by a set of coefficients whose size is an increasing power of two. The Haar equations to calculate an average a_i and a wavelet coefficient c_i from an odd and even element in the data set are:

$$\mathbf{a_i = s_i + s_{i+1} / \; 2}$$
$$\mathbf{c_i \; = s_i + s_{i-1} \; / \; 2}$$

Forward Haar transform for an eight element signal is shown in the Fig 3. Here signal is multiplied by the forward transform matrix.

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \Leftarrow \begin{bmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix}$$

Fig.3.Haar Forward transform for 8-element signal

© 2014, IJARCSSE All Rights Reserved

The arrow represents a split operation that reorders the result so that the average values are in the first half of the vector and the coefficients are in the second half. To complete the forward Haar transform there are two more steps. The next step would multiple the average values i a by a 4x4 transform matrix, generating two new averages and two new coefficients which would replace the averages in the first step. The last step would multiply these new averages by a 2x2 matrix generating the final average and the final coefficient.

4. Feature Extraction

The first step in the process is extracting image features to a distinguishable extent. The feature is defined as a function of one or more measurements, each of which specifies some quantifiable property of an object, and is computed such that it quantifies some significant characteristics of the object. Features can be classified as application independent features such as color, texture, and shape. On the other hand, all features can be coarsely classified into low-level features and high-level features. Low-level features can be extracted directed from the original images, whereas high-level feature extraction must be based on low-level features [8]. In this section we discuss various features that can be extracted from digital images for indexing and retrieval

4.1 Color Feature Extraction

The color feature is one of the most widely used visual features in image retrieval.

Typically, the color of an image is represented through some color model. There exist various color models to describe color information. A color model is specified in terms of 3-D coordinate system and a subspace within that system where each color is represented by a single point. The more commonly used color models are RGB (red, green, blue) and HSV (hue, saturation, value). Thus the color content is characterized by 3-channels from some color model. One representation of color content of the image is by using color histogram. Statistically, it denotes the joint probability of the intensities of the three-color channels. Color is perceived by humans as a combination of three-color stimuli: Red, Green, Blue, which forms a color space .RGB colors are called primary colors and are additive. By varying their combinations, other colors can be obtained. The representation of the HSV space is derived from the RGB space cube, with the main diagonal of the RGB model, as the vertical axis in HSV.

4.1.1 Color histograms

Each image in the database is computed to obtain the color histogram, which shows the proportion of pixels of each color within the image. The color histogram of each image is then stored in the database. When the user does the search by specifying the query image, the system registers the proportion of each color of the query image and goes through all images in the database to find those whose color histograms match those of the query most closely. The color histograms are used to represent the color distribution in an image. Mainly, the color histogram approach counts the number of occurrences of each unique color on a sample image. Since an image is composed of pixels and each pixel has a color, the color histogram of an image can be computed easily by visiting every pixel once. By examining the color histogram of an image, the colors existing on the image can be identified with their corresponding areas as the number of pixels. Histogram search characterizes an image by its color distribution, or histogram. Euclidian histogram distances have been used to define the similarity of two color histogram representations.

4.2 Texture Feature Extraction

Texture is a feature that is quite difficult to describe, and subjected to the difference of human perception. Texture feature extraction is computationally intensive, and the operating speed is very critical in CBIR systems, as the response time needs to be short enough for good interactivity. Therefore, our motivation is to provide a fast and efficient texture feature extraction method for the CBIR systems in embedded systems.

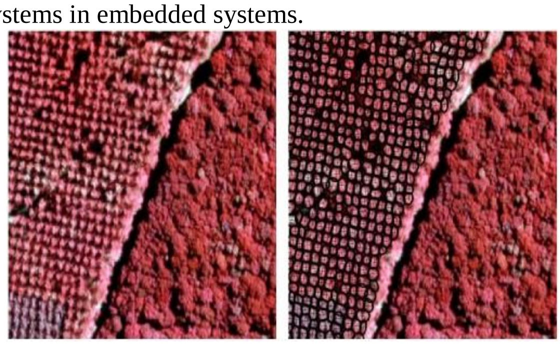


Fig 4: The original image and the result of extraction process

4.2.1 Gabor Filter

One of the most popular signal processing based approaches for texture feature extraction has been the use of Gabor filters. These enable filtering in the frequency and spatial domain. It has been proposed that Gabor filters can be used to model the responses of the human visual system. Turner [9] first implemented this by using a bank of Gabor filters to analyze texture. Gabor filters have been used in several image analysis applications including texture segmentation [4], defect detection [5], face recognition, motion tracking [9], and image retrieval. A bank of filters at different scales and orientations allows multichannel filtering of an image to extract frequency and orientation information. This can then be

© 2014, IJARCSSE All Rights Reserved

used to decompose the image into texture features. The feature is computed by filtering the image with a bank of orientation and scale sensitive filters and computing the mean and standard deviation of the output in the frequency domain. Filtering an image I(x, y) with Gabor filters gmn designed according to [10] results in its Gabor wavelet transform:

$$Wmn(x, y) = \int I(x, y)g^{*}mn(x - x1, y - y1)dx1dy1 \qquad \textbf{(1)}$$

The mean and standard deviation of the magnitude |Wmn| are used to for the feature vector. The outputs of filters at different scales will be over differing ranges. For this reason each element of the feature vector is normalized using the standard deviation of that element across the entire database.

4.3 Shape Feature Extraction
The shape of objects plays an essential role among the different aspects of visual information [10]. Therefore, it is a very powerful feature when used in similarity search and retrieval. If a shape is used as feature, edge detection might be the first step to extract that feature. In our work, the canny edge detector is used to determine the edge of the object in the scene. After the edge has been detected the important step is tracing the contour of the object in the scene. For this the edge image is scanned from four directions (right to left, left to right, top to bottom, bottom to top) and the first layer of the edge occurred is detected as image contour. To avoid discontinuities in object boundary the contour image is then re-sampled. After the object contour has been detected the first step in shape representation for an object is to locate the central point of the object. In our work, shape based image retrieval experiment is performed on a color image database.

4.3.1 Edge Detection
Edge detection is the process to detect the important features of image. Here, features mean the properties of image like discontinuities in physical and geometric characteristics of image or abrupt variation in the intensity of image. The quality of edges is affected by the presence of objects in similar illumination, noise and density of edges [12]. The variation in characteristics can leads to the variation in gray level of image. Edge detection represents an important step for facilitating higher-level image analysis and therefore remains an area of research with new approaches is continually being developed [9].

4.3.2 Canny Algorithm
Canny Edge Detection is one of the commonly used edge detection algorithm. It was developed in 1985 then it became popular because of its good localization and better response in noisy conditions. Canny Edge Detection algorithm is a multistage process used to detect the edges of the image.
1. Canny edge detector uses the first derivative of Gaussian to reduce the noise in image and produces a blur image.
2. The edge can be in any direction horizontally, vertically or diagonally, so the edge detector operator returns the first derivative in horizontal direction (Gx) and vertical direction (Gy). Edge direction is identified by Q=arctan (Gy/Gx) G=ÖGx2+Gy2 [1]
3. From the given values of image gradient, the direction of edge is calculated by comparing the gradient value with its local maxima. This step is also called as non-maximum suppression because it gives a wide range of edges including thin edges.
4. Once the gradient values have been computed, thresholding is performed. The total number of edge points depends on the value of threshold. Large the value of threshold produce small number of edges. Small the value of threshold produce large number of edges.
5. After applying the threshold, edge thinning is performed to remove the false edges that are shown in image. It removes all the unwanted edge pixels.

5. Feature Vector
Suppose X is a square matrix and $X_{f,t}$ is its f,tth order sub matrix where

$$X_{f,t} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,t} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{f,1} & x_{f,2} & \cdots & x_{f,t} \end{bmatrix} \qquad \textbf{(2)}$$

6. Similarity Criteria

The different feature extraction methods are explained above separately. The similarity feature which is used for comparing the various features is the Euclidean Distance. To retrieve the similarity images from the large image dataset, three types of Distance Metric Measures like Euclidean Distance, Chi-Square Distance and Weighted Euclidean but in the proposed method Euclidean distance is used.

Euclidean Distance:

The Euclidean distance between the color histograms h and g colors can be computed as:

$$d^2(h,g) = \sum_A \sum_B \sum_C (h(a,b,c) - g(a,b,c))^2 \quad \textbf{(3)}$$

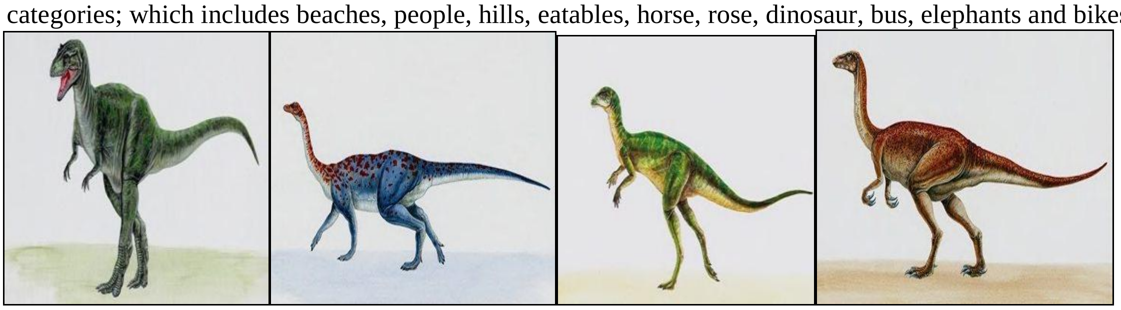
In this distance formula, there is only comparison between the identical bins in the respective histograms, where h(a,b,c) and g(a,b,c) represents the frequency values in bin. These features are compared with database images stored features. The features values which are less than defined threshold are sorted based on increasing difference between query and database images then stored separately.

7. Indexing and Retrieval

The objective of image indexing is to retrieve similar images from an image database for a given query image (i.e., a pattern image). Each image has its unique feature. Hence image indexing can be implemented by comparing their features, which are extracted from the images. The criterion of similarity among images may be based on the features such as color, intensity, shape, location and texture, and above mentioned other image attributes. Another important issue in content-based image retrieval is effective indexing and fast searching of images based on visual features. Because the feature vectors of images tend to have high dimensionality and therefore are not well suited to traditional indexing structures, dimension reduction is usually used before setting up an efficient indexing scheme. One of the techniques commonly used for dimension reduction is principal component analysis (PCA). It is an optimal technique that linearly maps input data to a coordinate space such that the axes are aligned to reflect the maximum variations in the data. In addition to PCA, many researchers have used Karhunen-Loeve (KL) transform to reduce the dimensions of the feature space. Although the KL transform has some useful properties such as the ability to locate the most important sub-space, the feature properties that are important for identifying the pattern similarity may be destroyed during blind dimensionality reduction. Apart from PCA and KL transformation, neural network has also been demonstrated to be a useful tool for dimension reduction of features [10]. After dimension reduction, the multi-dimensional data are indexed. A number of approaches have been proposed for this purpose, including R-tree (particularly, R*-tree [16]), linear quad-trees, K-d-B tree and grid files. Most of these multi-dimensional indexing methods have reasonable performance for a small number of dimensions (up to 20), but explore exponentially with the increasing of the dimensionality and eventually reduce to sequential searching. Furthermore, these indexing schemes assume that the underlying feature comparison is based on the Euclidean distance, which is not necessarily true for many image retrieval applications.

8. Experimental Results

In this paper, experimental data set contains number of images from database of images. We retrieved the image based on color, texture and shape feature parameter. We have shown some sample query image and it's retrieval image. Our methods has been implemented with a general-purpose image database including about 2000 pictures, which are stored in JPEG format with size 8x8. To make more efficient color histogram based CBIR method, firstly texture feature attribute i.e. cross correlation & color feature attributes are calculated for all 2000 database images against query image. There are different categories; which includes beaches, people, hills, eatables, horse, rose, dinosaur, bus, elephants and bikes etc.



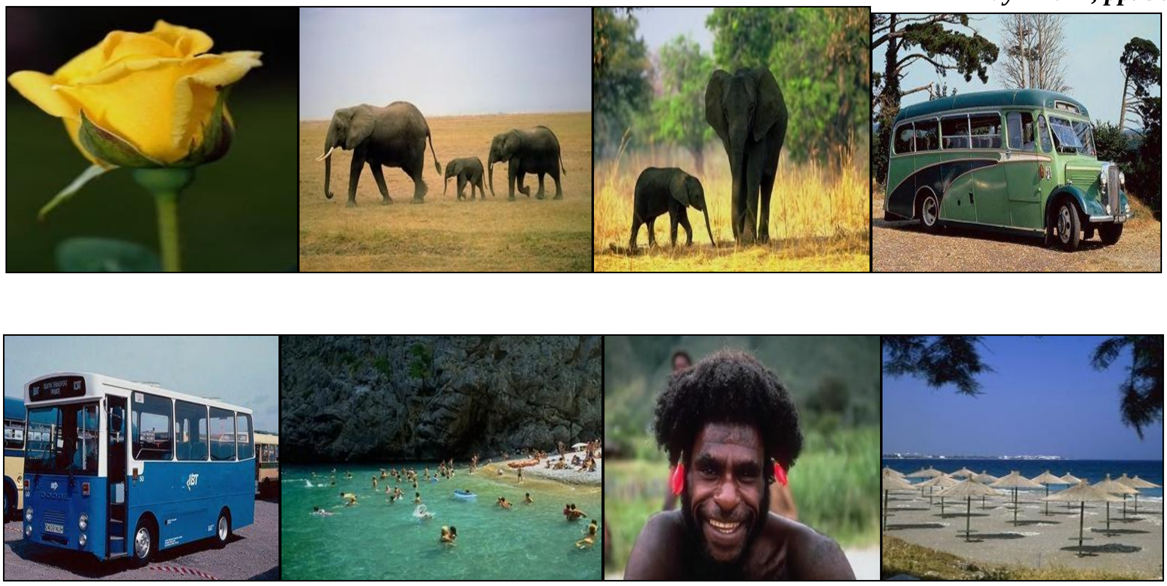


Figure 5: Sample Image database

The test image database contains over 2000 images of 24 bits true color. They are divided into 4 groups each containing 600 images. The four groups are Horses, Fishes, Sunflowers, and Roses. For simplicity, all images are pre processed to be 256x256 sizes before decomposition. Sample data base images for each category are shown in the Fig.6. The Recall rate is defined as the ratio of the number of relevant (same category) retrieved images to the number of relevant items in collection [8].

8.1 Retrieved Images

Figure 6: Query Image

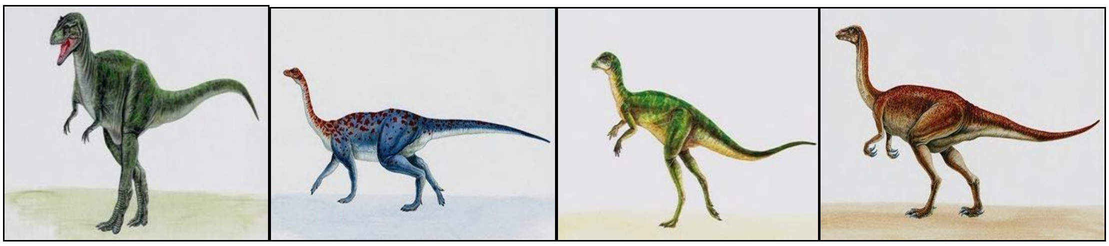


Figure 7: Retrieved Images

References

[1] Christopher C. Yang, "Content-based image retrieval: a comparison between query by example and image browsing map approaches ",Journal of Information Science, pp. 254-267, 2004.
[2] Rui Y. & Huang T. S., Chang S. F. "Image retrieval: current techniques, promising directions, and open issues". Journal of Visual Communication and Image Representation, 10, 39-62, 1999.
[3] Vishal Chitkara, "Color-Based image Retrieval Using Binary Signatures". Technical Report TR 01-08, University of Ulberta, Canada, May 2001.
[4] Qasim Iqbal and J. K. Aggarwal, "Combining Structure, Color, and Texture for Image Retrieval: A performance Evaluation". 16th International Conference on Pattern Recognition (ICPR), Quebec City, Canada, August 11-15. 2002, vol. 2, pp. 438-443

© 2014, IJARCSSE All Rights Reserved

[5] R. Calderbank, I. Daubechies, W. Sweldens and Y.Boon-Lock, "Wavelet transform that map integers to integers",Appl. Comut. Harmon. Anal., 5(3), 332-369, 1998

[6] M.G. Mostafa, M.F. Tolba, T.F. Gharib, M.A. Megeed, Medical Image Segmentation Using Wavelet based Multiresolution EM Algorithm". IEEE International Conference on Industrial Electronics, Technology, &Automation., Cairo IETA'2001.

[6] Zoran Pe¡cenovi´c, Minh Do, Serge Ayer, Martin Vetterli, New Methods for Image Retrieval, Laboratory for Audio-Visual Communications, Swiss Federal Institute of Technology (EPFL) CH-1015 Lausanne, Switzerland

[7] Meng, Xiannong. "A Comparative Study of Performance Measures for Information Retrieval Systems." Third International Conference on Information Technology: New Generations (ITNG'06). 2006. 578-579.

[8] QBIC - IBM's Query By Image Content. http://wwwqbic.almaden.ibm.com/ (accessed November 2008).

[9] Turner, M.: Texture discrimination by Gabor functions. Biological Cybernetics 55 (1986) 71–82

[10] O.D. Trier, A.K. Jain and T. Taxt, "Feature extraction methods for character recognition - a survey," Pattern Recognition, 29(4), pp. 641– 662, 1996.

[11] Manjunath, B., Wu, P., Newsam, S., Shin, H.: A texture descriptor for browsing and similarity retrieval. Journal of Signal Processing: Image Communication 16 (2000) 33–43

[12] Dr.H.B.Kekre, Ms. Swapna Borde," Content Based Image Retrieval," National Conference on Applications of Digital Signal Processing, NCDSP-2007, January 19-20, 2007

[13] Lei Zhu, Chun Tang, Aibing Rao and Aidong Zhang,"Using Thesaurus To Model Keyblock- Based Image Retrieval ," Technical Report, Department of Computer Science and Engineering, State University of New York At Buffalo,Buffalo,NY 14260,USA.

[14] K Bhandari, N Dugar, N Jain, N Shetty, "A novel Hiogh Performance Multi-Modal approach for Content Based Image Retrieval", ICWET 2010

[15] M.K. Hu, "Visual pattern recognition by moment invariants," IRE Trans. on Information Theory, 8, pp. 179–187, 1962.

[16] M.R. Teague, "Image analysis via the general theory of moments," Journal of Optical Society of America, 70(8), pp. 920–930, 1980.

[17] A. Khotanzad, "Rotation invariant pattern recognition using Zernike moments," in Proceedings of the International Conference on Pattern Recognition, pp. 326–328, 1988.

© 2014, IJARCSSE All Rights Reserved