Fusion of Colour, Shape and Texture Features for Content Based Image Retrieval

Pratheep Anantharatnasamy, Kaavya Sriskandaraja, Vahissan Nandakumar and Sampath Deegalla
Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka
pratheepap007@gmail.com
srikaavya@gmail.com
email@vahissan.com
dsdeegalla@pdn.ac.lk

***Abstract*—Image retrieval in general and content based image retrieval in particular are well-known research fields in information management. A large number of methods have been proposed and investigated in both areas but satisfactory general solution have still not been developed. An image contains several types of visual information which are difficult to extract and combine manually by humans. In this paper, we propose a content based image retrieval system based on three major types of visual information: colour, texture and shape, and their distances to the origin in a three dimensional space for the retrieval. We experimentally investigated several feature extraction methods and learning algorithms for content based image retrieval. The results show that 5-Nearest Neighbour yield the highest accuracy for the chosen feature extraction methods.**

I. Introduction

In many areas such as medicine, military, crime prevention, architecture, art and academic, large collections of digital images are being created. Many of these collections are the product of digitizing existing collections of photographs, diagrams, drawings, paintings, and prints. To access appropriate information, we need to retrieve these images from large image databases. Thus, image retrieval becomes an important issue. Image retrieval could be based on textual metadata or image content information. Traditional methods of image retrieval are based on associated metadata such as keywords and text [1]. The traditional metadata based image retrieval may suffer from several critical problems, such as, the lack of appropriate metadata associated with images, incorrect metadata, and the limitation of characters in the keywords to express the visual content of the image. In addition, it may not be feasible to manually add metadata to a large collection of images [2]. The problems of metadata based image retrieval and rapid growth in the quantity and availability of digital images motivates research into automatic image retrieval. In contrast to traditional images retrieval, Content Based Image Retrieval (CBIR) uses the information that is already available in the image.

In the next section, we review similar research studies and compare them with our proposed methodology. In section three, we present our proposed CBIR system and examine different visual information representation methods used in the system. In the performance evaluation section, results of different information representation methods along with different learning algorithms are presented. Finally, we present conclusions and future directions of the study.

II. Related Work

When it comes to web based CBIR systems, TinEye [3] was the first image search engine (according to Idee Inc [4]) on the web to use content based image retrieval. When an image is to be searched, TinEye creates a unique and compact digital signature or fingerprint for the image, then compares this fingerprint to every other image in database. TinEye does not find similar images, but finds exact matches including those that have been cropped, edited or resized. On the other hand, our proposed system will find both exact matches and near-exact matches based on similarity on some of the visual features.

FIRE, the Flexible Image Retrieval Engine [5], is a content based image retrieval system which focuses on evaluating different image descriptors, for efficient search [6]. In FIRE, images are extracted using colour and texture. Colour histogram is used for extract the colour and Tamura [7] texture is used to extract the texture from images. In addition to colour and texture features, shape features are also considered in our system.

Another project, which is called GNU Image-Finding Tool (GIFT) [8], is also a CBIR system. GIFT is an open framework which enables users to perform Query by Example [9] on images, giving the opportunity to improve query results by getting relevance feedback from users. For processing queries the program relies entirely on the content of the images.

Google Images [10] is a search service developed by Google that allows users to search the web for image content. This feature was introduced in 2001. The keywords for the image search are based on the file name of the image, the link text pointing to the image, and text adjacent to the image. Google Image Search uses visual content as well as metadata to retrieve images [11].

III. Methodology

A. Overview

The architecture of our proposed CBIR system can be divided into several components as follows:

1) Colour feature extraction

978-1-4673-4463-0/13/$31.00 ©2013 IEEE

2) Texture feature extraction
3) Shape feature extraction
4) Image classification
5) Combining the three extracted features
6) Similarity measure

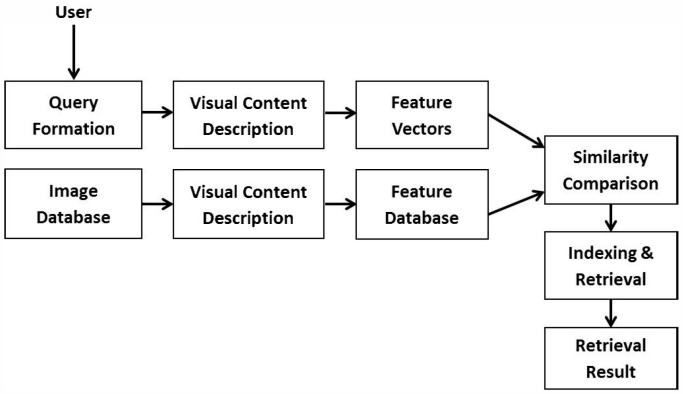

Fig. 1. Block diagram of the CBIR system

B. Colour Feature Extraction

The extraction of colour features from digital images depends on understanding the theory of colour and the representation of colour in digital images. Colour spaces are an important component of relating colour to its representation in digital form. The transformations between different colour spaces and the quantization of colour information are primary determinants of a given feature extraction method.

1) Colour Space Selection: A colour space is used to specify a three-dimensional colour coordinate system and a subspace of the system is in which colours are represented as three points [12]. The most common colour space for digital images and computer graphics is the RGB colour space [13] in which colours are represented as linear combinations of red (0 to 255), green (0 to 255), and blue (0 to 255) colour channels.

There are two main disadvantages with the RGB colour space:

- The RGB colour space is not perceptually uniform [14].
- All components (R, G, and B) have equal importance and, therefore, those values have to be quantized with the same precision.

In HSV colour space, colours are represented as combination of Hue (0 to 360), Saturation (0 to 1), and Value (0 to 1). The Value represents intensity of a colour, which is decoupled from the colour information in the represented image. The hue and saturation components are intimately related to the way human eye perceives colour resulting in image processing algorithms with physiological basis [15].

Since HSV colour space is close to human visual perception [16] and the hue component is more dominant than saturation and value components, we choose HSV colour space in this study.

2) Quantization of Colour Space: Quantization is the process of reducing the number of colours in a colour space by putting similar colours in the same bin. Quantization reduces computation time and comparison time of colour features. If we use a non-quantized colour space, we have to compare very large amount of colours (for example, in RGB 256 x 256 x 256 = 1677216 colours) between two images to find the similarity. Since we have already selected HSV as our colour space, we consider two types of quantization so that we can select the one which gives the best results:

- 14 bin quantization: 8 bins for Hue, 3 bins for Saturation, and 3 bins for Value.
- 24 bin quantization: 18 bins for Hue, 3 bins for Saturation, and 3 bins for Value.

Since the hue component dominantly decides the colour, we do not have to allocate more bins to saturation or value components. There is no need to quantize with too many bins either. Because that might lead to a case where the same colour might fall into two different bins and it is more costly by means of computational power.

3) Colour Descriptor Selection: Images can be represented by colour descriptors. In practice, colour descriptors such as colour histogram, colour moments, and colour coherent vector are used for content based image retrieval [17]. According to [17], performance of colour histogram which represents the number of pixels that have colours in each of the colour ranges, is superior compared to the other descriptors. The colour histogram is invariant to rotation of the image on the view axis, and changes in small steps when rotated otherwise or scaled [18]. Therefore, we choose colour histogram for our work.

C. Texture Feature Extraction

The ability to retrieve images solely based on the texture may not seem very useful. But it can often be useful to distinguish between areas of images with similar colour, such as sky and sea, or leaves and grass. There are several measures in texture feature comparison such as the degree of contrast, coarseness, directionality and regularity [7], or periodicity, directionality and randomness [19].

Many different methods for computing texture features have been proposed over the years. Unfortunately, there is still no single method that works best with all types of textures. So, we consider methods that work reasonably well on the most cases. The commonly used methods for texture feature description are statistical, structural and spectral methods [20], [21]. A statistical approach is the Gray Level Co-occurrence Matrix. This method characterizes texture by generating statistics of the distribution of intensity values as well as position and orientation of similar valued pixels. A structural approach to texture representation is characterized by generating complex texture patterns from lower level texture primitives, similar to how regular languages are generated by finite state automata.

Structural methods, including morphological operator and adjacency graph, describe texture by identifying structural primitives and their placement rules. They tend to be most

effective when applied to textures that are very regular. Spectral methods [21], such as Fourier and wavelet transforms, are used outside of texture analysis to extract features from images. Wavelets are used to search an image database from a low resolution example image or user-drawn sketch [22]. Their approach created image signatures of the query and stored images from the Haar wavelet decomposition method. Each signature is a truncated and quantized version of the coefficients computed from the images.

1) Gray Level Co-occurrence Matrix: Gray level co-occurrence matrix method uses grey-level co-occurrence matrix (GLCM) to sample statistically the way certain grey-levels occur in relation to other grey-levels. The elements of this matrix measure the relative frequencies of occurrence of grey level combinations among pairs of pixels with a specified spatial relationship.

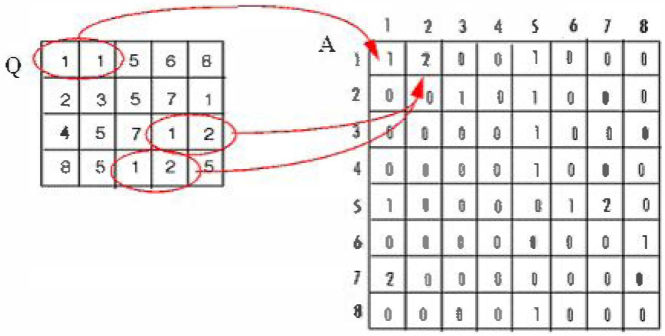

Fig. 2. Gray Level Co-occurrence Matrix

In order to estimate the similarity between different Gray level co-occurrence matrices, Haralick [23] proposed 14 statistical features extracted from them. To reduce the computational complexity, only some of these features were selected.

Energy, also called Angular Second Moment [23] and Uniformity in [24], is a measure of textural uniformity of an image. Energy reaches its highest value when Gray level distribution has either a constant or a periodic form. A homogeneous image contains very few dominant grey tone transitions, and therefore the P matrix (GCLM matrix - Fig. 2) [23] for this image will have fewer entries of larger magnitude resulting in large value for energy feature. In contrast, if the P matrix contains a large number of small entries, the energy feature will have smaller value. Entropy measures the disorder of an image and it achieves its largest value when all elements in P matrix are equal [24]. When the image is not texturally uniform many Gray level co-occurrence matrix elements have very small values, which imply that entropy is very large. Therefore, entropy is inversely proportional to Gray level co-occurrence matrix energy. Contrast is a difference moment of the P and it measures the amount of local variations in an image [23]. Inverse difference moment measures image homogeneity. This parameter achieves its largest value when most of the occurrences in Gray level co-occurrence matrix are concentrated near the main diagonal. Inverse Difference Moment (IDM) is inversely proportional to Gray level co-occurrence matrix contrast [25], [26].

2) Law's Texture Features: The "texture energy measures" developed by Laws have been widely used in texture analysis. Law's properties, which he called "texture energy measures", are derived from three simple vectors of length 3, L3 = (1,2,1), E3 = (-1,0,1) and S3 = (-1,2,-1), which represent the one-dimensional operations of center-weighted local averaging, symmetric first differencing (edge detection), and second differencing (spot detection). If we now multiply the column vectors of length 3 by row vectors of the same length, we obtain Law's 3x3 masks. The eight zero sum 3x3 masks (i.e. all but L3L3) are shown in Fig. 3.

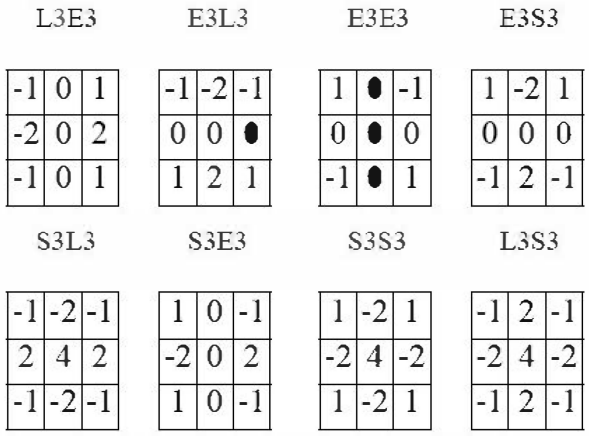

Fig. 3. Law's 3X3 masks

To use these masks for describing the texture in a (sub) image, we convolve them with the image and use statistics of the convolution result as textural features. Laws concluded that the most useful statistics are the variances of the convolution results. This was the basis for a feature extraction scheme based a series of pixel impulse response arrays obtained from combinations of 1-D vectors shown below. Each 1-D array is associated with an underlying micro structure and labelled using an acronym accordingly.

Level L5 = [1 4 6 4 1]
Edge E5 = [-1 -2 0 2 1]
Spot S5 = [-1 0 2 0 1]
Width W5 = [-1 2 0 -2 1]
Ripple R5 = [1 -4 6 -4 1]

Multiply these 1-D vectors with the transpose of each other we compute 25 different 55 matrices which are called convolution kernels (masks), typically labelled as L5L5 for the mask resulting from the convolution of the two L5 arrays. Then these kernels are applied to images and get the result of 25 different images in order to combined similar features such as horizontal edges and vertical edges.

D. Shape Feature Extraction

Compared to the other features like texture and colour, shape feature is more effective in characterising the content of an image. However, it is a challenging task to accurately extract the shape information from an image. The construction of shape descriptors is even more complicated when invariance with respect to a number of possible transformations, such as scaling, shifting and rotation is required [27].

1) Chain Codes Method: Chain codes are used to represent the boundary of a binary image by a connected sequence of straight-line segments of specified length and direction. Starting at a random pixel, chain code walks along all the pixels on an object's boundary. Typically, we use the angle based connectivity of segments to identify corners. When the direction of the boundary is changed by an angle bigger than a threshold value, it is identified as a corner, and then we count the number of corners in the image.

2) Area of an Object: Number of pixels which reside within a closed boundary represents the area of that particular object in the image. It can be used as a descriptor to represent the shape feature as different shapes will have different area.

3) Horizontal and Vertical Distances: Horizontal distance vector describes the variance of the shape of the object from top to bottom. We calculate the width of the object at each row of pixels by calculating the distance between the two boundary lines.

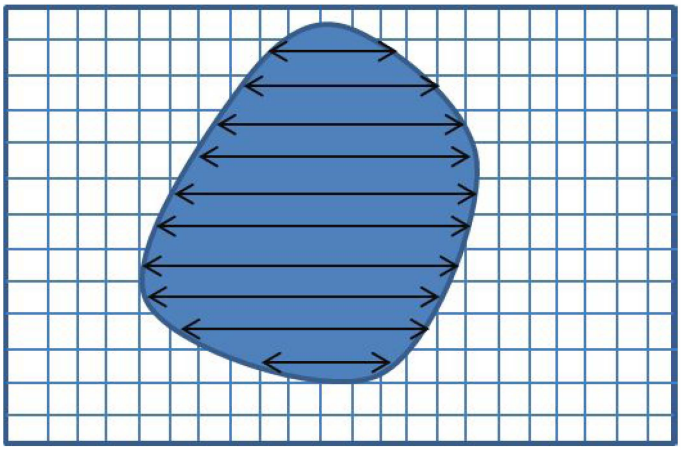

Fig. 4. Horizontal Distance Calculation

Vertical distance vector, as opposed to horizontal distance vector describes the variance of the shape of the object from left to right. The method is the same other than the orientation.

E. Similarity Measure and Feature Combination

Comparing and finding the most similar images using the extracted images is a key to a content based image retrieval project. There are a handful of similarities measures exist such as Euclidean Distance, Minkowski Distance, etc [28]. research and comparison on this part will result in choosing an appropriate method for each feature. Same goes with combining methods. There are several methods available; they

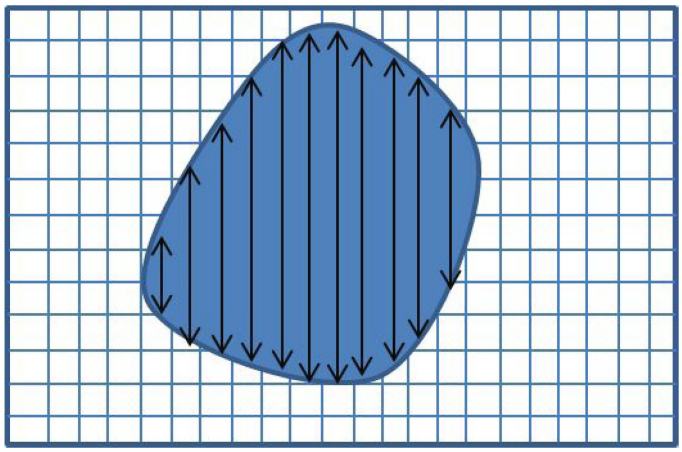

Fig. 5. Vertical Distance Calculation

are Neural Network-based Image Retrieval (NNIR), linear combining method, rank-based method, and BP-based method. We use a linear combining method which uses 3 coordinates which refers to colour, shape and texture. When a query image is given, euclidean distances with all the images in the database for all three features. Then, these distances are mapped on a 3 dimensional Cartesian space, and the euclidean distance between these points to origin is calculated. This refers to the distance by which an image differs from the query image. Weights are used to adjust the effect of each feature vector on calculating the distance to optimize the results for specific query images.

IV. Performance Evaluation

A. Experimental Setup

Waikato Environment for Knowledge Analysis (WEKA) [29] is a popular software suite for machine learning applications written in Java, developed at the University of Waikato, New Zealand. We have considered three learning algorithms in WEKA: Decision Trees (J48 in WEKA), Naive Bayes, and k-Nearest Neighbour (kNN) with k values of 3, 5, and 7. Comparison of these algorithms with 10-fold cross validation is carried out on 500 images split into 5 categories - Buses, Dinosaurs, Horses , Mountains, and Flowers - each containing 100 images.

B. Colour

In order to input the colour histogram to WEKA, we converted it into a one-dimensional feature vector, which contains the same number of elements as the number of bins used for colour quantization. For example, a feature vector for 14 bin colour quantization contains 14 elements in which the first 8 elements are for Hue (0 to 7), the next 3 elements are for Saturation (0 to 2), and the last 3 elements are for Value (0 to 2). We add the feature vectors for each image into a single Comma Separated Value (CSV) file, and load it into WEKA. The feature vector for 24 bin quantization was created using a similar approach.

According to the results obtained, it can be seen that even though the difference is not much, the 24 bin quantization has higher accuracy for most of the algorithms. In the meantime, 5-Nearest Neighbour performed well with both quantization methods. Therefore we chose 5-Nearest Neighbour as our classifier.

TABLE I
COMPARISON OF ACCURACY FOR DIFFERENT ALGORITHMS WITH BOTH QUANTIZATION METHODS

Algorithm	Accuracy (14 bins)	Accuracy (24 bins)
J48 Decision Tree	97.0%	97.0%
Naive Bayes	97.0%	97.5%
3–Nearest Neighbour	96.0%	97.5%
5–Nearest Neighbour	97.5%	97.5%
7–Nearest Neighbour	97.0%	96.5%

C. Texture

The algorithms we consider are Decision Trees (J48), Naive Bayes, and k-Nearest Neighbour with k values of 3, 5, and 7. In order to input the GLCM to WEKA, we classified the images with 256 grey levels, 0° angle (horizontal) and 12 matrices for offsets of 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, and 23 pixels. And we calculated contrast, correlation, energy, and homogeneity for each matrix. Finally, there are 48 features for each image. In order to input Laws Texture Feature we applied laws method for each image and generated 15 features for each image. We used 10 fold cross validation method for the classification.

TABLE II
COMPARISON OF ACCURACY FOR DIFFERENT TEXTURE EXTRACTION ALGORITHMS

Algorithm	Accuracy (GLCM)	Accuracy (Law's)
J48 Decision Tree	85.4%	75.6%
Naive Bayes	77.0%	70.6%
3–Nearest Neighbour	86.8%	82.8%
5–Nearest Neighbour	87.6%	81.0%
7–Nearest Neighbour	87.4%	80.2%

According to the results obtained, GLCM has the higher classification accuracy and faster than Law's texture method. GLCM took 2 minutes for 500 images and Law's texture took about 2 hours. Therefore we chose GLCM for the texture extraction. According to the result obtained, 5-Nearest Neighbour classifier has higher classification accuracy compared to the other four classifiers. So we chose 5-Nearest Neighbour as our classifier.

D. Shape

We use the chain codes method to calculate the number of corners for threshold angles varying from 120 degrees to 180 degrees by 10 degrees step size. In addition, we use the area, horizontal distance, and vertical distance of the images, and we tried different combination of these methods and evaluated them using Decision Trees and 5-Nearest Neighbour algorithms.

TABLE III
COMPARISON OF ACCURACY FOR DIFFERENT SHAPE EXTRACTION METHODS

Descriptor	Accuracy (J48)	Accuracy (5NN)
Chain Method	50.8%	53.2%
Vertical Distance	51.2%	46.2%
Horizontal Distance	49.0%	52.6%
Area	57.4%	54.2%
Chain Method + Area	65.8%	67.8%
Chain Method + Area + Horizontal Distance	72.3%	69.7%
Chain Method + Area + Vertical Distance	62.5%	62.5%

According to the results in Table III, the combination of chain method, area and horizontal distance has the acceptable accuracy level for each algorithm compared to other methods. Hence, we choose that method as our shape descriptor.

V. CONCLUSION

In this paper, we proposed combining colour, shape and texture features for content based image retrieval. We have experimentally investigated several feature extraction methods with several learning algorithm. We identified appropriate extraction method in colour, shape and texture feature and selected 5-Nearest Neighbour as the learning algorithm.

In the future, we plan to implement this system as a web-based system and try to improve the consistency of the results using relevance feedback from the user. In addition to content based retrieval, meta data can be used as a secondary method to support the retrieval of the image and furthermore, image clustering can be used to filter the images in the database according to their class.

REFERENCES

[1] "Image Retrieval," http://en.wikipedia.org/wiki/Image_retrieval, [Online; accessed 10-Jun-2012].

[2] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years," *IEEE Transactions on pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, December 2000.

[3] "TinEye Reverse Image Search," http://www.tineye.com, [Online; accessed 11-Feb-2012].

[4] "FAQ - TinEye," http://www.tineye.com/faq#what, [Online; accessed 11-Feb-2012].

[5] "Flexible Image Retrieval Engine," http://thomas.deselaers.de/fire/, [Online; accessed 11-Feb-2012].

[6] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: An experimental comparison," *Information Retrieval*, 2008.

[7] H. Tamura, "Textural features corresponding to visual perception," *IEEE Trans. Syst., Man and Cybern, 8*, pp. 460–472, 1978.

[8] "The GNU Image-Finding Tool," http://www.gnu.org/software/gift/, [Online; accessed 11-Feb-2012].

[9] M. M. Zloof, "Query-by-example: A data base language," *IBM Systems Journal, vol. 16, no. 4*, p. 324, 1977.

[10] "Google Images," http://images.google.com/, [Online; accessed 10-Jun-2012].

[11] "List of CBIR engines," http://en.wikipedia.org/wiki/List_of_CBIR_engines, [Online; accessed 10-Jun-2012].

[12] A. del Bimbo, *Visual Information Retrieval.* Morgan Kaufmann Publishers, San Francisco, CA, USA., 1999.

[13] S. Wang, "A robust cbir approach using local color histograms," *Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada, Tech. Rep. TR 01-13*, October 2001.

[14] J. K. C.-C. Vellaikal, A. and S. Dao, "Content-based retrieval of colour and multispectral images using joint spatial-spectral indexing," *SPIE Vol. 2606*, pp. 232–243, 2001.
[15] S. Jeong, "Histogram-based color image retrieval," 2001.
[16] E. Jessee and E. Wiebe, "Visual perception and the hsv color system: Exploring color in the communications technology classroom," 2008.
[17] S. Kodituwakku and S.Selvarajah, "Comparison of color features for image retrieval," *Indian Journal Of Computer Science And Engineering, 1(3)*, pp. 207–211, 2010.
[18] M. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory, 8*, pp. 179–187, 1962.
[19] R. P. F. Liu, "Detecting and segmenting periodic motion," *Media lab vision and modelling tr*, p. 400, 1996.
[20] J. K. Zijun Yang, "Survey on content-based analysis, indexing and retrieval techniques and status report of mpeg-7," *Tamkang journal of science and engineering, vol.2, No.3*, pp. 101–118, 1996.
[21] M. P. T. Ojala, "Texture classfication," *Machine vision and Media signal Processing Unit, University of Oulu, Finland.*
[22] M. Zachary, "An information theoretic approach tocontent based image retrieval," 2000.
[23] I. R. M. Haralick, K. Shanmugam, "Textural features for image classification," *IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-3, No.6*, pp. 610–621, November 1973.
[24] R. E. W. R. C. Gonzalez, "Digital image processing," 1993.
[25] A. R. J. S. Weszka, C. R. Dyer, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst. ManCybern, vol. SMC-6, no. 4*, 1976.
[26] F. P. A. Baraldi, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters," *IEEE Trans. On Geoscience and Remote Sensing, vol. 33, no. 2*, pp. 294–304, 1995.
[27] P. J. Belongie S., Malik J., "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 4*, p. 509522, 2002.
[28] R. C. Veltkamp, "Shape matching: Similarity measures and algorithms," 2001.
[29] "Weka 3 - data mining with open source machine learning software in java," http://www.cs.waikato.ac.nz/ml/weka/, [Online; accessed 10-Jun-2012].