Feature Extraction Techniques Based on Color Images

Dr. Sanjay Kumar,
Jaipur National University
Ankur Chauhan,
Jaipur National University

ABSTRACT

Nowadays various applications are available that claim to extract the correct information from such colored image databases which have different kinds of images and their own semantics. During information extraction based on the content of images various kinds of feature extraction techniques are available. This presented work focus on the information reflected by various feature extraction techniques and where they can be easily adaptable.

Keywords— content based filtering, face recognition, feature extraction, survey

I. INTRODUCTION

In machine learning process data is recognized using their meaningful patterns and extracted using the similarity between these patterns. To find the recognizable patterns among the data required to reduce the amount of data and extract the actual relationship or difference between two data instances. These relationships or differences are computed using the content of the data. Therefore that is a complex domain; where uncertainty and randomness nature of the data can be misguide the actual decision or recognition pattern.

The presented work in this paper is an evaluation of techniques by which the optimal properties between data can be evaluated. To find and form the optimum properties by which the nature of data and pattern of data can be recognize. The presented work is evaluation of the image data and finding the most appropriate feature extraction method, to utilize them in various applications.

For proper understanding of the relation between the data processing and image processing first we take an example, suppose we have a set of random documents, for categorizing or proper arrangement of these documents according to their domain, required to find some knowledge about the document contents, therefore first required to read a document and then evaluate the domains and topic reside in the given document. In the same way for finding the appropriate patterns over the given data, pre-processing, data model construction and implementation in problem is required.

Image is a different kind of data which includes a huge amount of information, such as color information, objects, edges, pixel definition, dimensions and others. Therefore the treatment of image data is a sensitive concern to preserve the complete information. this paper address the various key features and properties of image data by which the information from the image is extracted and utilized for different applications of face recognition, image retrieval and others.

II. BACKGROUND STUDY

A local feature is an image pattern which differs from its immediate neighborhood. It is usually associated with a change of an image property or several properties, though it is not localized exactly on this change. The image properties considered are intensity, color, and texture. Figure 1 shows some examples of local features in a contour image (left) as well as in a gray value image (right). Local features can be points, but also edges or small image patches. Some measurements are taken from region centered on a local feature and converted into descriptors. The descriptors can then be used for various applications [1].

Figure 1 Image Features

Good features should have the following properties:
Repeatability: If two images of the same object or scene are taken under different viewing conditions, a high percentage of similar features visible on both the images should be actually present in both the images.

Distinctiveness/in formativeness: The intensity patterns underlying the detected features should show a lot of variations, such that features can be distinguished and matched.

Locality: The features should be local, to reduce the probability of occlusion and to allow simple model approximations of the geometric and photometric deformations between two images taken under different viewing conditions (example: based on a local planarity assumption).

Quantity: The number of detected features should be large enough, so that a reasonable number of features are detected even on small objects. However, the optimal number of features depends on the application. Ideally, the number of detected features should be controllable over a large range by a simple and intuitive threshold. The density of features should reflect the information content of the image to provide a compact image representation.

Accuracy: The detected features should be localized accurately, with respect to scale and shape in both image locations.

Efficiency: Preferably, the detection of features in a new image should allow for time-critical applications. Repeatability, the most important property of all, can be achieved in two different ways: either by invariance or by robustness.

Invariance: When large deformations are to be expected, the preferred approach is to model these mathematically if possible, and then develop methods for feature detection that are unaffected by these mathematical transformations.

Robustness: In case of relatively small deformations, it often suffices to make feature detection methods less sensitive to such deformations, i.e., the accuracy of the detection may decrease, but not drastically. Typical deformations that are tackled using robustness are image noise, discretization effects, compression artefacts, blur, etc. Also geometric and photometric deviations from the mathematical model used to obtain invariance are often overcome by including more robustness.

III. CONTENT BASED IMAGE RETRIEVAL

An image retrieval system can be defined as searching, browsing, and retrieving images from massive databases consisting of digital images. Although Conventional and common techniques of retrieving images make use of adding metadata namely captioning keywords so as to perform annotation of words. However image search can be described by dedicated technique of search which is mostly used to find images. For searching images user provides the query image and the system returns the image similar to that of query image [2].

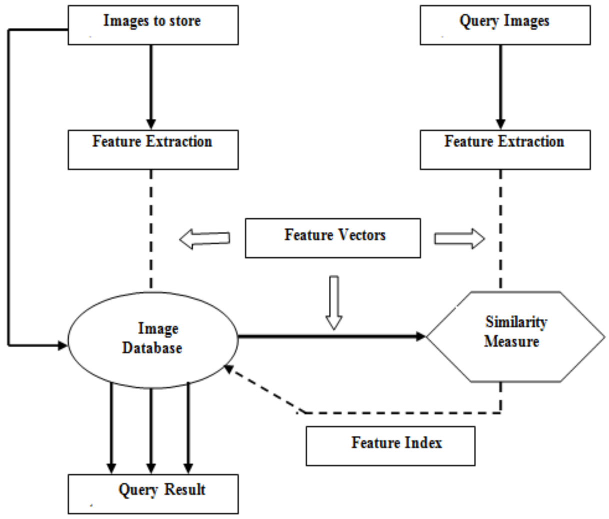


Figure 2 General Image Retrieval System

With the development of the Internet, and the availability of image capturing devices such as digital cameras, huge amounts of images are being created every day in different areas including remote sensing, fashion, crime prevention, publishing, medicine, architecture, etc. For this purpose, the need for the development of efficient and effective methodologies to manage large image databases for retrieval is urgent; so many general- purpose image retrieval systems have been developed. There are three methods for image retrieval: text-based method, content-based method and hybrid method. This section explains each method in details. Image retrieval system can be classified as:

- Text based Image retrieval system
- Content Based Image retrieval system

Text Based Image Retrieval (TBIR) is currently used in almost all general-purpose web image retrieval systems. This approach uses the text associated with an image to determine what the image contains. This text can be text surrounding the image, the image's filename, a hyperlink leading to the image, an annotation to the image, or any other piece of text that can be associated with the image. Google, Yahoo Image Search engines are examples of the systems using this approach. These search engines having indexed over one billion images. Although these search engines are fast and robust, they sometimes fail to retrieve relevant images, this is because of many reasons

- Firstly, there are too many irrelevant words in the surrounding textual descriptions, which results in low image search precision rate.
- Secondly, the surrounding text does not seem to fully describe the semantic content of Web images, which results in low image search recall rate.
- The third problem is polysemy problem (same word can be used to refer to more than one object). Due to

the query polysemy, the result searcher will fail to find images tagged in Chinese, and a Dutch searcher will fail to find images tagged in English. This means the query must match the language of the text associated with the images.

Content Based Image Retrieval is a set of techniques for retrieving semantically-relevant images from an image database based on automatically-derived image features [3]. This aims at avoiding the use of textual descriptions and instead retrieves images based on their visual similarity to a user-supplied query image or user-specified image features.

The main goal of CBIR is maintaining efficiency during image indexing and retrieval, thereby reducing the need for human intervention in the indexing process [4]. The computer must be able to retrieve images from a database without any human assumption on specific domain (such as texture vs. non texture). One of the main tasks for CBIR systems is similarity comparison, extracting feature of every image based on its pixel values and defining rules for comparing images. These features become the image representation for measuring similarity with other images in the database. Images are compared by calculating the difference of its feature components to other image descriptors. Research Center, extracts several features from each image, namely colour, texture and shape features [5]. These descriptors are obtained globally by extracting information on the means of colour histograms for colour features; global texture information on coarseness, contrast, and direction; and shape features about the curvature, moments invariants, circularity, and eccentricity. Similarly, the Photo book system features to represent image semantics [6]. These global approaches are not adequate to support the queries looking for images where specific objects in an image having particular colours and/or texture are present, and shift/scale invariant queries, where the position and/or the dimension of the query objects may not relevant [7].

Most of the existing CBIR systems consider each image as a whole; however, a single image can include multiple regions/objects with completely different semantic meanings. A user is often interested in only one particular region of the query image instead of the image as a whole. Therefore, rather than viewing each image as a whole, it is more reasonable to view it as a set of regions. The features employed by the majority of Image Retrieval systems include colour, texture, shape and spatial layout. Such features are apparently not effective for CBIR, if they are extracted from a whole image, because they suffer from the different backgrounds, overlaps, occlusion and cluttering in different images and do not have adequate ability to capture important properties of objects, as a result most popular approaches in recent years are to change the focus from the global content description of images into the local content description by regions or even the objects in images. RBIR is a promising extension of the classical CBIR: rather than deploying global features over the entire content, RBIR systems divide an image into number of homogenous regions and extract local features for each region then features of various regions are used to represent and index images in RBIR. For RBIR, The user supplies a query object by selecting a region of a query image and then the corresponding similarity measure is computed between features of region in the query and a set of features of segmented regions in the features database and the system returns a ranked list of images that contain the same object. The content-based approach can be summarized as follows:

1. Computer vision and image processing techniques are used to extract content features from the image.
2. Images are represented as collections of their prominent features. For a given image, an appropriate representation of the feature and a notion of similarity are determined.
3. Image retrieval is performed based on computing similarity or Dissimilarity in the feature space, and results are ranked based on the similarity measure.

IV. LOW LEVEL FEATURE EXTRACTION TECHNIQUES

This section includes the various feature vector calculation methods that are consumed to design algorithm for image retrieval system.

Grid Color Moment

Color feature is one of the most widely used features in low level feature. Compared with shape and texture feature, color feature shows better stability and is more insensitive to the rotation and zoom of image. Color not only adds beauty to objects but also adds more information, which is used as powerful tool in content-based image retrieval. In color indexing, given a query image, the goal is to retrieve all the images whose color and texture compositions are similar to that of query image. In color image retrieval there are various methods, but here we will discuss some prominent methods.

The feature vector we will use is called "Grid-based Color Moment". Here is an example which shows how to compute this feature vector for a given image: [8]

- Convert the image from RGB for HSV color space (Hint: use the function rgb2hsv in Matlab for this operation)
- Uniformly divide the image into 3x3 blocks
- For each of these nine blocks
- Compute its mean color (H/S/V)

$$x' = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Where N is the number of pixels within each block, x_i is the pixel intensity in H/S/V channels.

- Compute its variance (H/S/V)

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - x')^2$$

- Compute its skewness (H/S/V)

$$\gamma = \frac{\frac{1}{n}\sum_{i=1}^{N}(x_i - x')^3}{(\frac{1}{n}\sum_{i=1}^{N}(x_i - x')^2)^{3/2}}$$

- Each block will have 3+3+3=9 features, and thus the entire image will have 9x9=81 features. Before we use SVM to train the classifier, we first need to normalize the 81 features to be within the same range, in order to achieve good numerical behavior. To do the normalization, for each of the 81 features:
- Compute the mean and standard deviation from the training dataset

$$\mu = \frac{1}{M}\sum_{i=1}^{M} f_i$$

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=0}^{M}(f_i - \mu)^2}$$

- M is the number of images in the training dataset, and f_i is the feature of the i^{th} training sample.
- Perform the "whitening" transform for all the data (including both the training data and the testing data), and get the normalized feature value:

$$f'_i = \frac{f_i - \mu}{\sigma}$$

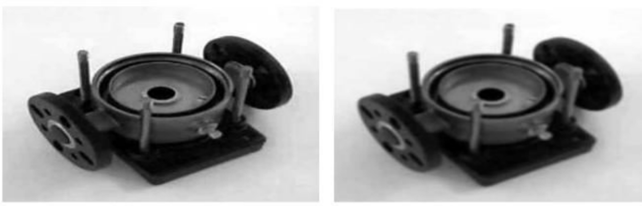
Canny Edge Detection

The purpose of edge detection in general is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Several algorithms exists, and this worksheet focuses on a particular one developed by John F. Canny (JFC) in 1986. [9, 10]

The algorithm runs in 5 separate steps:

1. **Smoothing:** Blurring of the image to remove noise.
2. **Finding gradients:** The edges should be marked where the gradients of the image has large magnitudes.
3. **Non-maximum suppression:** Only local maxima should be marked as edges.
4. **Double thresholding:** Potential edges are determined by thresholding.
5. **Edge tracking by hysteresis:** Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

Smoothing



(a) Original (b) Smoothed

Figure 3 Smoothing effect on image

It is inevitable that all images taken from a camera will contain some amount of noise. To prevent that noise is mistaken for edges, noise must be reduced. Therefore the image is first smoothed by applying a Gaussian filter where the kernel of Gaussian filter with a standard deviation is σ = 1.4. The effect of smoothing the test image with this filter is shown in Figure.

Finding gradients

The gradient magnitudes (also known as the edge strengths) can be determined as an Euclidean distance measure by applying the law of Pythagoras.

$$|G| = \sqrt{G_x^2 + G_y^2}$$

It is sometimes simplified by applying Manhattan distance measure to reduce the computational complexity.

$$|G| = |G_x| + |G_y|$$

G_x and G_y are the gradients in the x- and y-directions respectively.

The Euclidean distance measure has been applied to the test image. The computed edge strengths are compared to the smoothed image in Figure (4).

(a) Smoothed (b) Gradient magnitudes

Figure 4 Gradient magnitudes of image

The image of the gradient magnitudes often indicates the edges quite clearly. However, the edges are typically broad and thus smoothing do not indicate exactly where the edges are. To make it possible to determine this, the direction of the edges must be determined and stored as.

$$\theta = arcTan\left(\frac{|G_y|}{|G_x|}\right)$$

Non-maximum suppression

The purpose of this step is to convert the “blurred” edges in the image of the gradient magnitudes to “sharp” edges. Basically this is done by preserving all local maxima in the gradient image, and deleting everything else. The algorithm is for each pixel in the gradient image:

1. Round the gradient direction θ to nearest 45°, corresponding to the use of an 8-connected neighborhood.
2. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient direction i.e. if the gradient direction is north

(θ = 90◦), compare with the pixels to the north and south.
3. If the edge strength of the current pixel is largest; preserve the value of the edge strength. If not, suppress (i.e. remove) the value.

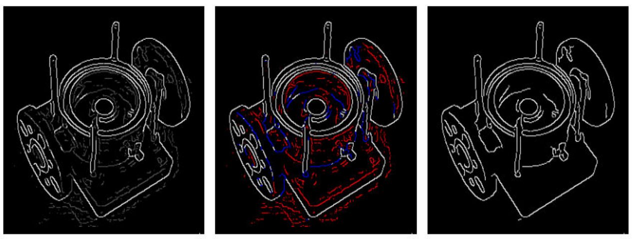
Double thresholding

The edge-pixels remaining after the non-maximum suppression step are (still) marked with their strength pixel-by-pixel. Many of these will probably be true edges in the image, but some maybe caused by noise or colour variations for instance due to rough surfaces. The simplest way to discern between them would be to use a threshold, so that only edges stronger that a certain value would be preserved. The Canny edge detection algorithm uses double thresholding. Edge pixels stronger than the high threshold are marked as strong, whereas, edge pixels weaker than the low threshold are suppressed and those between the two thresholds are marked as weak.

Edge tracking by hysteresis

Strong edges are interpreted as “certain edges”, and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges. The logic is of course that noise and other small variations are unlikely to result in a strong edge (with proper adjustment of the threshold levels). Thus strong edges will (almost) only be due to true edges in the original image. The weak edges can either be due to true edges or noise/colour variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges.

Edge tracking can be implemented by BLOB-analysis (Binary Large Object). The edge pixels are divided into connected BLOB’s using 8-connected neighbourhood. BLOB containing at least one strong edge pixel is then preserved, while other BLOB’s are suppressed. The effect of edge tracking on the test image is shown in Figure.



(a) Double thresholding (b) Edge tracking by hysteresis (c) Final output

Figure 5 Blob Analysis

Local Binary Pattern

Given a pixel in the image, an LBP [11] code is computed by comparing it with its neighbours:

$$LBP_{P,R} = \sum_{P=0}^{P=1} s(g_p - g_e)2^p$$

$$s(x) = \begin{cases} 0 & x \geq 0 \\ 1 & x < 0 \end{cases}$$

g_e is the gray value of the central pixel , g_p is the value of its neighbors, P is the total number of involved neighbours and R is the radius of the neighborhood. Suppose the coordinate of g_e is (0, 0), then the coordinates of g_p are

$$\left(Rcos\left(\frac{2\pi p}{P}\right), Psin\left(\frac{2\pi P}{P}\right)\right)$$

The gray values of neighbours that are not in the image grids can be estimated by interpolation. Suppose the image is of size I*J after the LBP pattern of each pixel is identified, a histogram is built to represent the texture image:

$$H(k) = \sum_{i=1}^{I}\sum_{J=1}^{j} f\left(LBP_{p,r}(i,j), k\right), k \in [0,k]$$

$$f(x,y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases}$$

K is the maximal LBP pattern value. The U value of an LBP pattern is defined as the number of spatial transitions (bitwise 0/1 changes) in that pattern

$$U\left(LBP_{P,R}\right) = \left|s\left(g_{p-1} - g_e\right) - s(g_0 - g_e)\right| + \sum_{p=1}^{p-1}\left|s\left(g_p - g_e\right) - s\left(g_{p-1} - g_e\right)\right|$$

The uniform LBP patterns refer to the patterns which have limited transition or discontinuities ($U \leq 2$) in the circular binary presentation. In practice, the mapping from$LBP_{P,R}$to$LBP_{P,R}^{u2}$which has P*(P-1)+3 distinct output values, is implemented with a lookup table of 2^p elements. To achieve rotation invariance, a locally rotation invariant pattern could be defined as:

$$LBP_{P,r}^{riu2} = \begin{cases} \sum_{p=0}^{p-1} s\left(g_p - g_e\right) & if U\left(LBP_{P,R}\right) \leq 2 \\ P + 1 & otherwise \end{cases}$$

The mapping from$LBP_{P,R}$to$LBP_{P,R}^{u2}$which has P+2 distinct output values.

GABOR filter

In the one-dimensional case, the Gabor function consists of a complex exponential (a cosine or sine function, in real case) localized around x = 0 by the envelope with a Gaussian window shape [10].

$$g_{\alpha,\varepsilon}(x) = \sqrt{\alpha/\pi}\, e^{-\alpha x^2} e^{-i\varepsilon x}$$

$\alpha \epsilon R^+$ and $\varepsilon, x \epsilon R$, where $\alpha = (2\sigma^2)^{-1}, \sigma^2$ is variance and ε is frequency. Dilation of the complex exponential function and shift of the Gaussian window, when the dilation is fixed form kernel of a Gabor transform. The Gabor transform (a special case of the short-time Fourier transform) employs such kernel for time-frequency signal analysis. The mentioned Gaussian window is the best time frequency localization window in a sense of the Heisenberg uncertainty principle [12].

In a two-dimensional case, the absolute square of the correlation between an image and a two-dimensional Gabor function provides the spectral energy density concentrated around a given position and frequency in a certain direction. Moreover, the two-dimensional convolution with a circular (non-elliptical) Gabor function is separable to series of one-dimensional ones

$$g_{\alpha,\varepsilon}(x) = g_{\alpha,\varepsilon 0}(x_0) g_{\alpha,\varepsilon 1}(x_1)$$

For $\varepsilon = (\varepsilon_0, \varepsilon_1)$ and $x = (x_0, x_1)$Here, the actual frequency of the two-dimensional function is determined by$\varepsilon = (\varepsilon_0^2 + \varepsilon_1^2)^{1/2}$ Furthermore $\vartheta = \arctan\left(\frac{\varepsilon_1}{\varepsilon_0}\right)$is an angle between x-axis and a linear perpendicular to the ridges of a wave.

Gabor Wavelet

Elements of a family of mutually similar Gabor functions are called wavelets when they are created by dilation and shift from one elementary Gabor function (mother wavelet), i.e.

$$g_{\alpha,\varepsilon,a,b}(x) = |a|^{-1/2} g_{\alpha,\varepsilon}\left(\frac{x-b}{a}\right)$$

for $a\epsilon R^+$(scale) and$b\epsilon R$(shift). By convention, the mother wavelet has the energy localized around x = 0 as well as all of the wavelets are normalized$\|g\| = 1$. Although the Gabor wavelets do not form orthonormal bases, the discrete set of them form a frame.

The used notation is in accordance with [13]. The first order partial derivative of image I with respect to variable x is denoted by I_x. Analogously I_{XX}denotes the second order partial derivative with respect to x and I_{xy} is the second order mixed derivative. Furthermore$I_x(x, \sigma_D)$) denotes a partial derivative obtained at the location of an point x and calculated by using a Gabor wavelet with scale $a \propto \sigma_D$

Edge Detection

For the edge detection, the convolution in two perpendicular directions is performed with variously dilated wavelets (e.g., separately in row and column directions). It is necessary to use a wavelet which serves as the first order partial differential operator (e.g., a first derivative of a Gaussian function).Consequently, local maxima of module is

$$M(x, \sigma_D) = \sqrt{I_x^2(x, \sigma_D) + I_y^2(x, \sigma_D)}$$

found. Only the maxima above a given threshold are considered (due to noise and slight edges). As a result, the edges for each scale are obtained.

Corner Detection

The key idea is to obtain the partial derivatives needed for a construction of an autocorrelation matrix

$$\mu(x, \sigma_I, \sigma_D) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(x, \sigma_D) & I_x I_y(x, \sigma_D) \\ I_x I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix}$$

by using the convolution with the Gabor wavelets. A Gaussian window of SI scale is used to determine the average of the derivatives. On this matrix, detectors are based. Also here, it is necessary to use such a Gabor wavelet which serves as the first order partial differential operator.

Blob Detection

Following the same principle, BLOB'S can be detected [14] from the second order partial derivatives using a Hessian matrix

$$H(x, \sigma_D) = \begin{bmatrix} I_{XX}(x, \sigma_D) & I_{xy}(x, \sigma_D) \\ I_{xy}(x, \sigma_D) & I_{yy}(x, \sigma_D) \end{bmatrix}$$

V. CONCLUSION AND FUTURE WORK

In this presented study paper, a survey is conducted for finding methods of content based image retrieval process. In addition of that, the feature vector estimation and various frequently used techniques are also evaluated. In near future this technique is utilized to introduce a new image feature calculation technique, which is used for color image recognition and clearer and efficient edge detection.

REFERENCES

[1] KRYSTIAN MIKOLAJCZYK, TINNE TUYTELAARS, "Local Image Features", Universiteit Leuven, Kasteelpark Arenberg 10, Leuven, Belgium

[2] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee,"Reweighted Random Walks for Graph Matching", Computer Vision–ECCV 2010, 2010 – Springer

[3] Priti Maheswary, Dr. Namita Srivastava, "Retrieval of Remote Sensing Images Using Colour & Texture Attribute",(IJCSIS) International Journal of Computer Science and Information Security,Vol. 4,No. 1&2, 2009

[4] S. Oraintara, T. T. Nguyen, "Using Phase and Magnitude Information of the Complex directional Filter Bank for Texture Image Retrieval", IEEE International Conference on Image Processing, Vol. 4, Pages 61-64, Oct. 2007

[5] P. S. Hiremath and Jagadeesh Pujari. "Content Based Image Retrieval based on Color, Texture and Shape features using Image and its complement", International Journal of Computer Science and Security, Volume (1) : Issue (4)

[6] Philipp Sandhaus, Mohammad Rabbath, and Susanne Boll,"Employing Aesthetic Principles for Automatic PhotoBook Layout",Advances in Multimedia Modeling, 2011 – Springer

[7] Michele Saad, "Low-Level Color and Texture Feature Extraction for Content-Based Image Retrieval", Final Project Report, EE 381K: Multi-Dimensional Digital Signal Processing, Pages: 20-28, May 2008

[8] Yu-Gang Jiang, Jun Yang, Chong-Wah Ngo, Alexander G. Hauptmann, “Representations of Key point-Based Semantic Concept Detection: A Comprehensive Study”, IEEE, 2008
[9] Canny Edge Detection, March 23, 2009
[10] Masoud Nosrati, Ronak Karimi, Mehdi Hariri, “Detecting Circular Shapes From Areal Images Using Median Filter and CHT”, World Applied Programming, Vol (2), Issue (1), . 49-54, January 2012
[11] Zhenhua Guo,Lei Zhang,David Zhang, “A Completed Modeling of Local Binary Pattern Operator for Texture Classification”, IEEE transaction on image processing, 2010
[12] S. Mallat, “A Wavelet Tour of Signal Processing: The Sparse Way”, 3 edition, 2009
[13] Joni-Kristian Kamarainen, “Gabor Features in Image Analysis”, Machine Vision and Pattern Recognition Laboratory, Lappeenranta University of Technology (LUT Kouvola)
[14] Larnaca, Cyprus, “An MPEG-7 Image Retrieval Systemof Atherosclerotic Carotid Plaque Images”, IEEE 12th International Conference on Bioinformatics& Bioengineering (BIBE), 11-13 November 2012