Chapter 1
INTRODUCTION

Retrieval

The process of getting something from somewhere is termed as retrieval. The action of obtaining or consulting material stored in a computer system. Retrieval may be retrieving information, retrieving multimedia or an image retrieval and so on…

1.1 Image Retrieval

In this computer age, virtually all spheres of human life including commerce, government, academics, hospitals, crime prevention, surveillance, engineering, architecture, journalism, fashion and graphic design, and historical research use images for efficient services. A large collection of images is referred to as image database. An image database is a system where image data are integrated and stored. Image data include the raw images and information extracted from images by automated or computer assisted image analysis.

The police maintain image database of criminals, crime scenes, and stolen items. In the medical profession, X-rays and scanned image database are kept for diagnosis, monitoring, and research purposes. In architectural and engineering design, image database exists for design projects, finished projects, and machine parts. In publishing and advertising, journalists create image databases for various events and activities such as sports, buildings, personalities, national and international events, and product advertisements. In historical research, image databases are created for archives in areas that include arts, sociology, and medicine. In a small collection of images, simple browsing can identify an image. This is not the case for large and varied collection of images, where the user encounters the image retrieval problem. In an image retrieval problem is encountered when searching and retrieving images that are relevant to a user's request from a database. To solve this problem, text-based and content-based are the two techniques adopted for search and retrieval in an image database.

1.1.2 Text-Based and Content-Based Image Retrieval

In text-based retrieval, images are indexed using keywords, subject headings, or classification codes, which in turn are used as retrieval keys during search and retrieval. Text-based retrieval is non-standardized because different users employ different keywords for annotation. Text descriptions are sometimes subjective and incomplete because they cannot depict complicated image features very well. Examples are texture images that cannot be described by text. Textual information about images can be easily searched using existing technology, but requires humans to personally describe every image in the database. This is impractical for very large databases, or for images that are generated automatically, e.g. from surveillance cameras. It is also possible to miss images that use different synonyms in their descriptions. Systems based on categorizing images in semantic classes like "cat" as a subclass of "animal" avoid this problem, but still face the same scaling issues.

The Image Retrieval Based on content (IRBC) technique uses image content to search and retrieve digital images. Content-based image retrieval systems were introduced to address the problems associated with text-based image retrieval. Image retrieval based on content is a set of techniques for retrieving semantically-relevant images from an image database based on automatically-derived image features. The main goal of IRBC is efficiency during image indexing and retrieval, thereby reducing the need for human intervention in the indexing process. The computer must be able to retrieve images from a database without any human assumption on specific domain (such as texture vs. non-texture, or indoor vs. outdoor).

One of the main tasks for IRBC systems is similarity comparison; extracting feature signatures 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. An image is compared to other images by calculating the difference between their corresponding features.

1.2 About the Project

The design and development of effective and efficient IRBC systems are still a research problem, because the nature of digital images involves two well-known problems: the semantic gap and the computational load to manage large file collections. The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation. It has linguistic and contextual consequences, and mainly depends on the domain knowledge to represent images.

On the other hand, the computation load, when large image collections are managed, may make impractical use of IRBC systems. The aim of this thesis is to propose a new IRBC system; an important task of the system is

1) To reduce the "semantic gap" between low-level image features and the richness of human semantics and

2) To reduce the overall retrieval time. The system first segments images into regions that correspond to the objects in it. A combination of texture, and color features are extracted from each region in the segmented image.

The contribution of this work in the following directions:

a). Salient low-level texture features are extracted from regions using GLCM and color features using color moments, which have been a widely acclaimed natural and excellent tool in image classification, segmentation, and extraction.

b). Many of the existing systems attempt to compare the query image with every target image in the database to find the top matching images, resulting in an essentially linear search, which is prohibitive when the database is large. We believe that it is not necessary to conduct a whole database comparison. In fact, it is possible to exploit a priori information regarding the “organization” of the images in the database in the feature space before a query is posed, such that when a query is received, only a part of the database needs to be searched, while a large portion of the database may be eliminated in the search. This certainly saves significant query processing time without compromising the retrieval precision.

c).To further increase the performance of the system, we develop a global searching algorithm (referred to as global features based IRBC) that uses texture and color features from the whole image to compute the distance between two images. This algorithm is combined with the region based searching algorithm using weighted sum of the two distances, and by this we use properties of image regions associated with the general properties of the image for similarity computation between a query and database images.

d).We make a comparison between image retrieval using region based features and global based features. Results illustrate that the system developed in this thesis significantly improves the overall retrieval quality compared to the text based existing systems.

1.3 Fields of Application

Image retrieval based on content is extremely useful in a plethora of applications such as publishing and advertising, historical research, fashion and graphic design, architectural and engineering design, crime prevention, medical diagnosis, geographical information and remote sensing systems, etc.

- A typical image retrieval application example is a design engineer who needs to search his organization database for design projects similar to that required by his clients
- A police seeking to confirm the face of a suspected criminal among faces in the database of renowned criminals.
- In the commerce department, before trademark is finally approved for use, there is need to find out if such or similar ones ever existed.
- In hospitals, some ailments require the medical practitioner to search and review similar X-rays or scanned images of a patient before proffering a solution.
- The most important application, however, is the Web, as big fraction of it is devoted to images, and searching for a specific image is indeed a daunting task. Numerous commercial and experimental IRBC systems are now available, and many web search engines are now equipped with IRBC facilities, as for example Alta Vista, Yahoo and Google.

1.4 Principle of IRBC

IT uses the contents of images to represent and access the images. A typical content-based retrieval system is divided into off-line feature extraction and online image retrieval. A conceptual framework for content-based image retrieval is illustrated in Figure. In off-line stage, the system automatically extracts visual attributes (color, shape, texture, and spatial information) of each image in the database based on its pixel values and stores them in a different database within the system called feature database. The feature data (also known as image signature) for each of the visual attributes of each image is very much smaller in size compared to the image data, thus the feature database contains an abstraction (compact form) of the images in the image database. One advantage of a signature over the original pixel values is the significant compression of image representation. However, a more important reason for using the signature is to gain an improved correlation between image representation and visual semantics.

In on-line image retrieval, the user can submit a query example to the retrieval system in search of desired images. The system represents this example with a feature vector. The distances (i.e., similarities) between the feature vectors of the query example and those of the media in the feature database are then computed and ranked. Retrieval is conducted by applying an indexing scheme to provide an efficient way of searching the image database. Finally, the system ranks the search results and then returns the results that are most similar to the query examples. If the user is not satisfied with the search results, he can provide relevance feedback to the retrieval system, which contains a mechanism to learn the user's information needs.

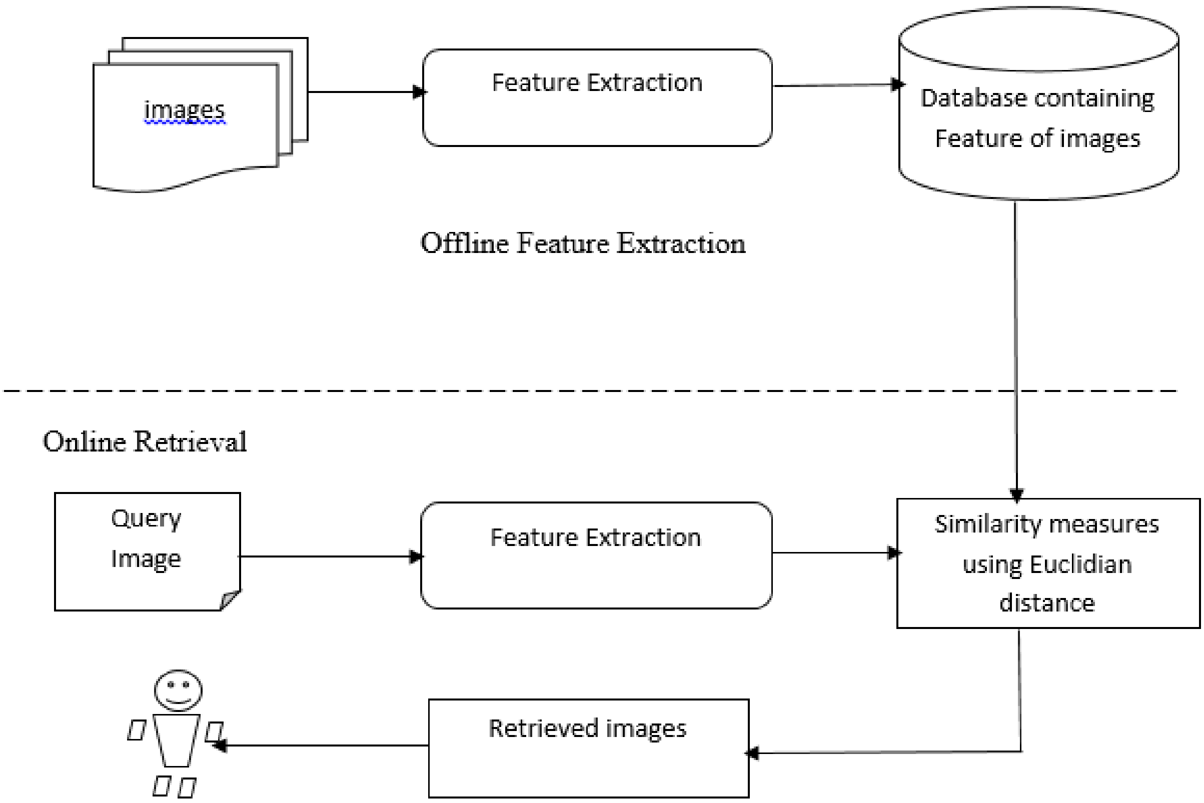


Figure 1.1: A Conceptual Framework for IRBC

1.5 Organization of the Project thesis

The rest of the project thesis is organized as follows.

- Chapter 2 summarizes some of the related works in the topic of IRBC and the primary research issues.
- In Chapter 3, color feature extraction is present which includes color spaces, color moments and how the moments are calculated globally and locally.
- Chapter 4 focuses on texture feature extraction i.e.. Overview of texture, GLCM, texture feature extraction.
- In chapter 5 image retrieval methodology is discussed which includes offline feature vector database and online image retrieval. IRBC system, its principle, and the techniques used for feature extraction, similarity measure. We also present the combination of the color and texture features.
- Simulation results and observation, are detailed in Chapter 6.

Finally, Chapter 7 concludes our work and Chapter 8 suggests future work.

Chapter 2

Literature Survey

IRBC for general-purpose image databases is a highly challenging problem because of the large size of the database, the difficulty of understanding images, both by people and computers, the difficulty of formulating a query, and the issue of evaluating results properly. A number of general-purpose image search engines have been developed. In the commercial domain, QBIC [7] is one of the earliest systems. Recently, additional systems have been developed such as T.J. Watson [26], VIR [10], AMORE [13], and Bell Laboratory WALRUS. In the academic domain, MIT Photobook is one of the earliest systems. Berkeley Blobworld [27], Columbia Visualseek and Webseek [9], Natra [16], and Stanford WBIIS are some of the recent well known systems. The common ground for IRBC systems is to extract a signature for every image based on its pixel values and to define a rule for comparing images. The signature serves as an image representation in the "view" of an IRBC system. The components of the signature are called features. One advantage of a signature over the original pixel values is the significant compression of image representation. However, a more important reason for using the signature is to gain an improved correlation between image representation and semantics. Actually, the main task of designing a signature is to bridge the gap between image semantics and the pixel representation, that is, to create a better correlation with image semantics [11]. Existing general-purpose IRBC systems roughly fall into three categories depending on the approach to extract signatures: histogram, color layout, and region-based search. There are also systems that combine retrieval results from individual algorithms by a weighted sum matching metric [6], or other merging schemes [18].

After extracting signatures, the next step is to determine a comparison rule, including a querying scheme and the definition of a similarity measure between images. For most image retrieval systems, a query is specified by an image to be matched. We refer to this as global search since similarity is based on the overall properties of images. By contrast, there are also "partial search" querying systems that retrieve results based on a particular region in an image [19].

2.1 Global Feature Based IRBC Systems

Some of the existing IRBC systems extract features from the whole image not from certain regions in it; these features are referred to as Global features.

Histogram search algorithms [7] characterize an image by its color distribution or histogram. Many distances have been used to define the similarity of two color histogram representations. Euclidean distance and its variations are the most commonly used. The drawback of a global histogram representation is that information about object location, shape and texture is discarded. Color histogram search is sensitive to intensity variations, color distortions, and cropping. The color layout approach attempts to overcome the drawback of histogram search. In simple color layout indexing [7], images are partitioned into blocks and the average color of each block is stored. Thus, the color layout is essentially a low resolution representation of the original image. A relatively recent system, WBIIS [17], uses significant Daubechies' wavelet coefficients instead of averaging. By adjusting block sizes or the levels of wavelet transforms, the coarseness of a color layout representation can be tuned. Hence, we can view a color layout representation as an opposite extreme of a histogram. At proper resolutions, the color layout representation naturally retains shape, location, and texture information. However, as with pixel representation, although information such as shape is preserved in the color layout representation, the retrieval system cannot perceive it directly. Color layout search is sensitive to shifting, cropping, scaling, and rotation because images are described by a set of local properties [6].Image retrieval using color features often gives disappointing results, because in many cases, images with similar colors do not have similar content.

D. Zhang [20] proposed a method combining both color and texture features to improve retrieval performance. By computing both the color and texture features from the images, the database images are indexed using both types of features. During the retrieval process, given a query image, images in the database are firstly ranked using color features. Then, in a second step, a number of top ranked images are selected and re-ranked according to their texture features. Two alternatives are provided to the user, one is the retrieval based on color features, and the other is retrieval based on combined features. When the retrieval based on color fails, the user will use the other alternative which is the combined retrieval. Since the texture features are extracted globally from the image; they are not an accurate description of the image in some cases, which degrades the system performance.

2.2 Region Based IRBC Systems

Region-based retrieval systems attempt to overcome the deficiencies of global feature based search by representing images at the object-level. A region-based retrieval system applies image segmentation to decompose an image into regions, which correspond to objects if the decomposition is ideal [12]. The object-level representation is intended to be close to the perception of the human visual system (HVS). Since the retrieval system has identified what objects are in the image, it is easier for the system to recognize similar objects at different locations and with different orientations and sizes. Region based retrieval systems include the Natra system [16], and the Blobworld system [27]. The Natra and the Blobworld systems compare images based on individual regions. The motivation is to shift part of the comparison task to the users. To query an image, a user is provided with the segmented regions of the image and is required to select the regions to be matched and also attributes, e.g., color and texture, of the regions to be used for evaluating similarity. Such querying systems provide more control to the user. However, the user's semantic understanding of an image is at a higher level than the region representation. For objects without discerning attributes, such as special texture, it is not obvious for the user how to select a query from the large variety of choices. Thus, such a querying scheme may add burdens on users without significant reward.

Recently, Natsev et al. considered the similarity model WALRUS [14], which is a robust model for scaling and translation of objects within an image. Each image is first decomposed into regions. The similarity measure between two images is then defined as the fraction of the area of the two images covered by matching regions. However, WALRUS focuses on the development of a fast and effective segmentation method instead of an image-to-image similarity measure. Consequently, region matching should be necessary before image matching. The authors proposed a greedy heuristic for computing the similar region pair set with the maximum area. The basic idea is to iteratively choose the best pair of matching regions that maximizes the area covered by the regions. The time complexity of the above greedy algorithm is O(n2), where n is the number of matching pairs obtained by the R*-tree search. In [21], the mean shift algorithm is used for segmentation of images and interested regions are indexed using cluster-based R*-tree to increase the efficiency of the retrieval process. However, this system uses only color as image signature, which is sensitive to shifting, cropping, scaling, and rotation. Also, query is by image region matching, while a user’s semantic understanding of an image is at a higher level than region representation.

Region based image retrieval [22] uses low-level features including color, texture, and edge density. For color, the histogram of image regions are computed, for texture co-occurrence matrix based entropy, energy, etc., are calculated, and for edge density it is Edge Histogram Descriptor (EHD) that is found. To decrease the retrieval time of images, an idea is developed based on greedy strategy to reduce the computational complexity. In this strategy, the query image is compared to each of the target images in the database based on region matching in term of Euclidean distance between them. The system then arranges the segments of each image in decreasing order based on the size of each segment. When a query is presented to the system, it starts comparing from the first region, if the distance between the query region and the target region is less than the threshold value, the system continues to check the other regions; otherwise it exits and does not check the other segments marking the target image as an irrelevant one.

To measure the similarity between images, Li and Wang et al [11], proposed the Integrated Region Matching (IRM) algorithm, which allows matching a region of one image to several regions of another image. That is, the region mapping between any two images is a many-to-many relationship. As a result, the similarity between two images is defined as the weighted sum of distances in the feature space, between all regions from different images. Compared with retrieval systems based on individual regions, such as Blob world, the IRM approach decreases the impact of inaccurate segmentation by smoothing over the imprecision in distances. IRM incorporates the properties of all the segmented regions so that information about an image can be fully used. To increase the robustness against segmentation errors, IRM allows a region to be matched to several regions in another image. Each matching is assigned a significance credit, which corresponds to the importance of the matching. There are several ways to assign the importance of a region. One can assume that every region is equally important. IRM views that important objects in an image tend to occupy larger areas, called an area percentage scheme. This scheme is less sensitive to inaccurate segmentation than the uniform scheme. If one object is partitioned into several regions, the uniform scheme raises its significance improperly, whereas the area percentage scheme retains its significance to the region. Fuzzy Club [28] addresses the issue of effective and efficient content based image retrieval by presenting an indexing and retrieval system that integrates color, texture, and shape information for the indexing and retrieval, and applies these region features obtained through unsupervised segmentation, as opposed to applying them to the whole image domain. Fuzzy Club emphasizes improving on a color feature “inaccuracy” problem in the region based literature – that is color histogram bins are not independent. For instance, if the color spectrum is divided into 10 bins, these bins are not independent some are closer or farther away from each other in the original color space. Fuzzy logic is applied to the traditional color histogram to solve this problem to some degree. Fuzzy Club first segments an image into regions of 4x4 blocks and extracts color and texture features on each block. The k-means algorithm is used to cluster similar pixels together to form a region. The Lab color space is used to extract color features and Haar wavelet transform is used to extract three texture features. A secondary clustering is performed to reduce query processing time. Regions with similar features are grouped together in the same class. This secondary clustering is performed offline, and each region's indexing data along with its associated class ID are recorded in the index files. The distances between each query region and all class centroids in the database are computed to determine to which class these query regions belong. The similar regions in the database are returned and all the images that have any of the member regions are assigned as candidate. The query image is compared to the candidate image set.

2.3 Research Issues:

1. In the RBIR systems, the texture features are obtained during segmentation from small blocks. Such features do not properly represent the property of an entire region; thus it is necessary to study texture feature extraction from the whole region after segmentation.
2. Even though RBIR systems increased the retrieval accuracy, they require high complex computations to calculate similarity, since these systems need to consider each region in the database images, resulting in high retrieval response time. Thus, we need a solution to reduce the number of database regions included in the similarity computation.
3. The existing IRBC systems use either global features, or region based features to represent the content of an image. Each type of these features can be significant in representing images with certain semantics. For example, global features are useful for retrieving textured images that have no specific regions in accordance to the user, such as natural scenes used as backgrounds. Thus, utilizing an integration of both types of features can improve the performance of the retrieval system.

Chapter 3
Color Feature Extraction

IRBC is the retrieval of images based on visual features such as color and texture. In IRBC, each image that is stored in the database has its features extracted and compared to the features of the query image. It involves two steps:

Feature Extraction: The first step in the process is extracting image features to a distinguishable extent.

Matching: The second step involves matching these features to yield a result that is visually similar.

In this system we used global color moments in extracting the color features of images. The main issue regarding the use of color for image retrieval involves the choice of color space.

3.1 Color Spaces

A color space is a mathematical representation of a set of colors. The three most popular color models are RGB (used in computer graphics); YIQ, YUV, or YCbCr (used in video systems); and CMYK (used in color printing). However, none of these color spaces are directly related to the intuitive notions of hue, saturation, and brightness. This resulted in the temporary pursuit of other models, such as HSI and HSV, to simplify programming, processing, and end-user manipulation.

All of the color spaces can be derived from the RGB information supplied by devices such as cameras and scanners.

3.1.1 RGB Color Space

An **RGB color space** is any additive color spaces based on the RGB color model. A particular RGB color space is defined by the three chromaticity's of the red, green, and blue additive primaries and can produce any chromaticity that is the triangle defined by those primary colors. The complete specification of an RGB color space also requires a white point chromaticity and a gamma correction curve. As of 2007, RGB is by far the most commonly used RGB color space.

RGB is an abbreviation for red–green–blue.

An RGB color space can be easily understood by thinking of it as "all possible colors" that can be made from three colorants for red, green and blue. Imagine, for example, shining three lights together onto a white wall: one red light, one green light, and one blue light, each with dimmer switches. If only the red light is on, the wall will look red. If only the green light is on, the wall will look green. If the red and green lights are on together, the wall will look yellow. Dim the red light and the wall will become more of a yellow-green. Dim the green light instead, and the wall will become more orange. Bringing up the blue light a bit will cause the orange to become less saturated and more whitish. In all, each setting of the three dimmer switches will produce a different result, either in color or in brightness or both. The set of all possible results is the gamut defined by those particular color lamps. Swap the red lamp for one of a different brand that is slightly more orange, and there will be a slightly different gamut, since the set of all colors that can be produced with the three lights will be changed.

An LCD display can be thought of as a grid of thousands of little red, green, and blue lamps, each with their own dimmer switch. The gamut of the display will depend on the three colors used for the red, green and blue lights. A wide-gamut display will have very saturated, "pure" light colors, and thus be able to display very saturated, deep colors.

Results:

Here is a sample image:

Figure 3.1 Sample Image

In Java, image is read by using a method in predefined class: javax.imageio;

The method prototype is as follows:

BufferedImage img = ImageIO.read (new File("image name"));

The above image is 32*32.The height and width are found using getWidth() and getHeight() methods of BufferedImage class. An image is represented as set of pixels. This is shown as follows:

Image of dimensions 8*5

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

Table 3.1 Image Representation

Pixel values are stored in an array as follows:

0	**1**	**2**	**.**	**.**	**.**	**.**	**.**	**39**

RGB values of every pixel is extracted using a predefined method in BufferedImage class

img.getRGB (0, 0, w, h, pixels, 0, w); where w is width, h is height and pixels[] is an array containing rgb components.

For the above image rgb values are:

Pixel	R	G	B
0	155	160	163
1	160	155	159
2	123	100	108

Table 3.2 RGB Values for certain pixels

3.1.2 HSV Color Spaces

In the literature, there is no optimum color space known for image retrieval, however certain color spaces such as HSV, Lab, and Luv have been found to be well suited for the content based query by color. We adopt to use the HSV (Hue, Saturation, and Value) color space for its simple transform from the RGB (Red, Green, Blue) color space, in which all the existing image formats are represented. The HSV color space is a popular choice for manipulating color, it is developed to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color. RGB to HSV is a nonlinear, but reversible transformation. The hue (H) represents the dominant spectral component (color in its pure form), as in red, blue, or yellow. Adding white to the pure color changes the color: the less white, the more saturated the color is. This corresponds to the saturation (S). The value (V) corresponds to the brightness of color. The hue (color) is invariant to the illumination and camera direction, and thus suitable for object recognition. Figure 3, shows the cylindrical representation of the HSV color space. The angle around the central vertical axis corresponds to “hue” denoted by the angle from 0 to 360 degrees, the distance from the axis corresponds to “saturation” denoted by the radius, and the distance along the axis corresponds to “lightness”, “value” or “brightness” denoted by the height.

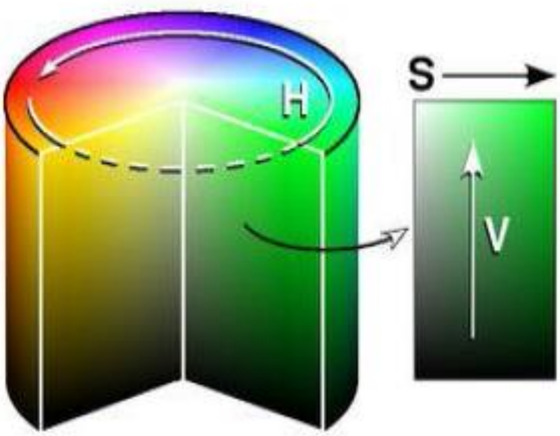


Fig 3.2 HSV Color Space

The HSV values of a pixel can be transformed from its RGB representation according to the following formulas:

$$\boldsymbol{H = arctan\ \frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}}$$

$$\boldsymbol{S = 1 - \frac{min\{R,G,B\}}{V}}$$

$$\boldsymbol{V = \frac{R+G+B}{3}}$$

Note: Refer RGB values from the table 3.2

Results:

For the above mentioned image r,g,b components are converted into hsv color space using above formulae.These are some sample hsv component values of pixels for above image:

Pixel	H	S	V
0	**202.5**	**0.031**	**163**
1	**-48**	**0.019**	**160**
2	**-20.869**	**0.0901**	**123**

Table 3.3 HSV Values for certain pixels

3.2 Color Moments

Color moments are measures that can be used differentiate images based on their features of color. Once calculated, these moments provide a measurement for color similarity between images. These values of similarity can then be compared to the values of images indexed in a database for tasks like image retrieval.

The basis of color moments lays in the assumption that the distribution of color in an image can be interpreted as a probability distribution. Probability distributions are characterized by a number of unique moments (e.g. Normal distributions are differentiated by their mean and variance). It therefore follows that if the color in an image follows a certain probability distribution, the moments of that distribution can then be used as features to identify that image based on color.

Stricker and Orengo use three central moments of an image's color distribution. They are Mean, Standard deviation and Skewness. A color can be defined by 3 or more values. (Here we will restrict ourselves to the HSV scheme of Hue, Saturation and brightness, although alternative encoding could just as easily be used.) Moments are calculated for each of these channels in an image. An image therefore is characterized by 9 moments 3 moments for each 3 color channels. We will define the i^{th} color channel at the j^{th} image pixel as p_{ij}. The three color moments can then be defined as:

MOMENT 1 – Mean:

$$\boldsymbol{E_i = \sum_{j=1}^{N} \frac{p_{ij}}{N}}$$

Mean can be understood as the average color value in the image.

MOMENT 2- Standard Deviation:

$$\boldsymbol{\sigma_i = \sqrt{\left(\frac{\sum_{j=1}^{N}(p_{ij} - E_i)^2}{N}\right)}}$$

The standard deviation is the square root of the variance of the distribution.

MOMENT 3 – Skewness:

$$\boldsymbol{s_{i=} \sqrt[3]{\left(\frac{\sum_{j=1}^{N}(p_{ij} - E_i)^3}{N}\right)}}$$

Skewness can be understood as a measure of the degree of asymmetry in the distribution.

3.2.1 Global Retrieval

In Global Retrieval, color moments are calculated from three color channels R, G, B. Therefore, 3 moments for each channel results in 9 moments. Global moments form the feature vector. Global moments are calculated for every image in the database and during retrieval global moments of query image are calculated. A match is obtained by using a similarity measure.

3.2.2 Local Retrieval

In Local Retrieval, every image is segmented into same number of regions. For example, in this project image is divided into 6 regions and this may be extended to any number of regions. The number of features extracted from each region is same. For every region 9 moments are extracted. Hence, entire image constitute 9*(no. of regions) moments. This forms the feature vector. During retrieval query image is segmented into regions and moments for every region are calculated. In order to find a match, the distances between regions must be computed.
Image is divided into 6 and 9 regions as follows:

1	**2**	**3**
4	**5**	**6**

1	**2**	**3**
4	**5**	**6**
7	**8**	**9**

3.3 Constructing Feature Vector

In the case of Global retrieval 9 moments forms the feature vector where a in the case of Local retrieval region based moments i.e.… 9 moments for every region forms the feature vector.

Feature Vector Size = number of regions X number of color channels X 3 floating point numbers.
In our case, the image is divided horizontally into six regions and we have to store 54 floating point numbers per image as well as 81 floating point numbers for 9 equal regions.
So, the feature vector f_c of length 54 is given by:

$f_c = \{E_{11}, \sigma_{11}, s_{11}, E_{12}, \sigma_{12}, s_{12}, E_{13}, \sigma_{13}, s_{13}, \ldots, E_{23}, \sigma_{23}, s_{23}\}$

Note: $f_c = \{E_{ri}, \sigma_{ri}, s_{ri}\}$

Here ‘r’ represents the region and ‘i’ represents the color channel.

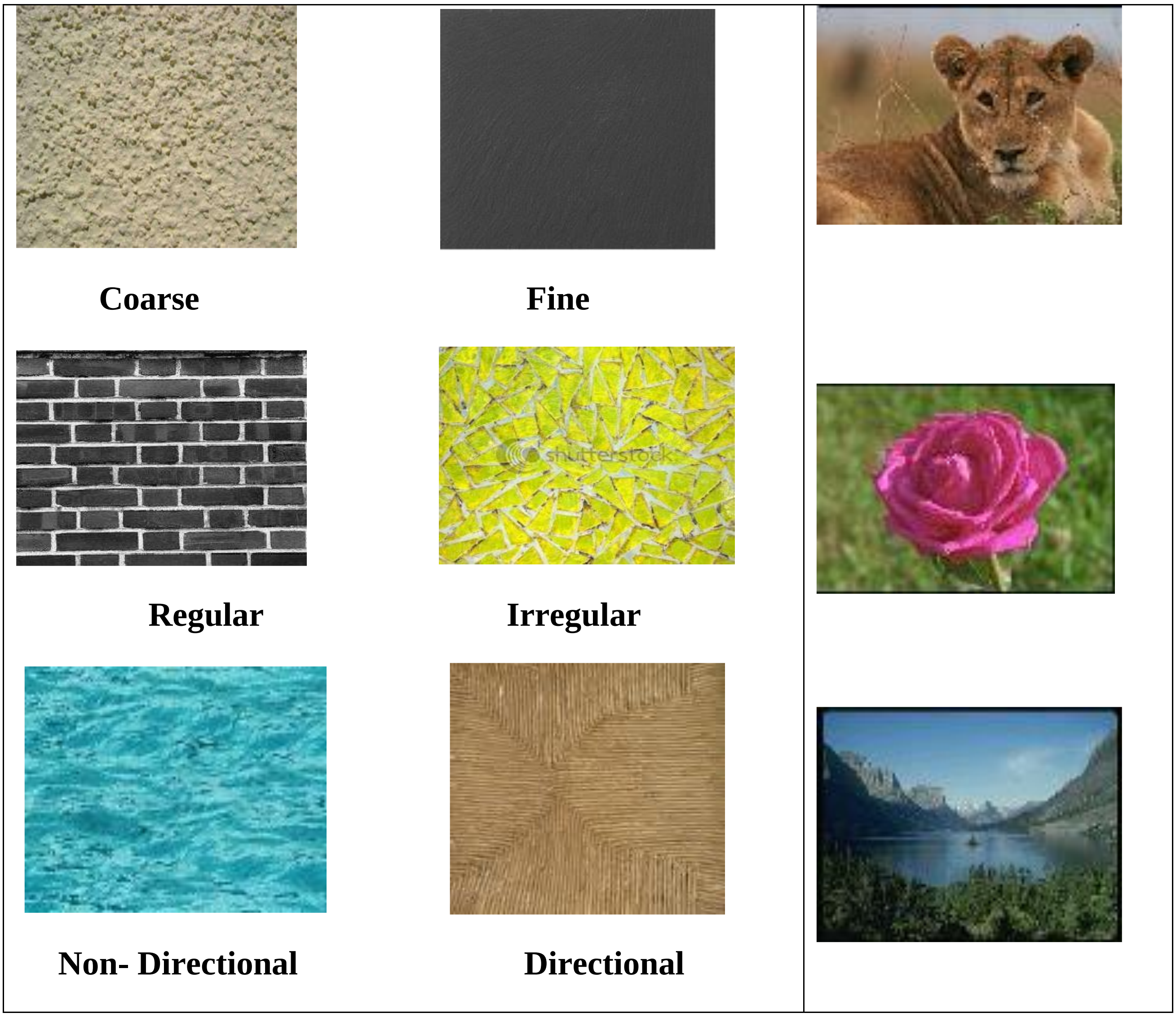
Chapter 4

Texture Feature Extraction

4.1 Texture

In the field of computer vision and image processing, there is no clear-cut definition of texture. This is because available texture definitions are based on texture analysis methods and the features extracted from the image. However, texture can be thought of as repeated patterns of pixels over a spatial domain, of which the addition of noise to the patterns and their repetition frequencies results in textures that can appear to be random and unstructured. Texture properties are the visual patterns in an image that have properties of homogeneity that do not result from the presence of only a single color or intensity. The different texture properties as perceived by the human eye are, for example, regularity, directionality, smoothness, and coarseness, see fig

a)Simple Texture Images **b)Complex Texture**



Images

Fig 4.1 Examples of Simple and Complex Texture Images

In real world scenes, texture perception can be far more complicated. The various brightness intensities give rise to a blend of the different human perception of texture as shown in Figure 4(b). Image textures have useful applications in image processing and computer vision. They include: recognition of image regions using texture properties, known as texture classification, recognition of texture boundaries using texture properties, known as texture segmentation, texture synthesis, and generation of texture images from known texture models.

Since there is no accepted mathematical definition for texture, 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. According to Manjunath and Ma, the commonly used methods for texture feature description are statistical, model-based, and transform-based methods. The texture feature description categories are explained below.

4.1.1 Statistical Methods

Statistical methods analyze the spatial distribution of grey values by computing local features at each point in the image, and deriving a set of statistics from the distribution of the local features. They include co-occurrence matrix representation, statistical moments, gray level differences, autocorrelation function, and grey level run lengths. The most commonly used statistical method is the Gray-level Co-occurrence Matrix (GLCM). It is a two-dimensional matrix of joint probabilities $P_{d,r}(i, j)$ between pairs of pixels, separated by a distance, d, in a given direction, r. It is popular in texture description and is based on the repeated occurrence of some gray level configuration in the texture; this configuration varies rapidly with distance in fine textures and slowly in coarse textures. Haralick defined 14 statistical features from gray-level co-occurrence matrix for texture classification, such as energy, entropy, contrast, maximum probability, autocorrelation, and inverse difference moment. .

Gray-level co-occurrence matrix method of representing texture features has found useful applications in recognizing fabric defects, and in rock texture classification and retrieval.

4.1.2 Model Based Approaches

Model-based texture methods try to capture the process that generated the texture. By using the model-based features, some part of the image model is assumed and an estimation algorithm is used to set the parameters of the model to yield the best fit. To describe a random field, assume the image is modeled as a function f (r, ω), where r is the position vector representing the pixel location in the 2-D space and ω is a random parameter. For a given value of r, f (r, ω) is a random variable (because ω is a random variable). Once a specific texture ω is selected, f (r, ω) is an image, which is a function over the two-dimensional grid indexed by r. Function f (r, ω) is called as a random field. There are currently three major model based methods: Markov random fields by Dubes and Jain, fractals by Pentland, and the multi-resolution autoregressive features introduced by Mao and Jain.

4.1.3 Transform Domain Features

The word transform refers to a mathematical representation of an image. There are several texture classifications using transform domain features in the past, such as discrete Fourier transform, discrete wavelet transforms, and Gabor wavelets. Transform methods analyze the frequency content of the image to determine texture features. Fourier analysis consists of breaking up a signal into sine waves of various frequencies. On the other hand, wavelet analysis breaks up a signal into shifted and scaled versions of the original wavelet (mother wavelet), which refers to decomposition of a signal into a family of basic functions obtained through translation and dilation of a special function. Moments of wavelet coefficients in various frequency bands have been shown to be effective for representing texture. Gabor filter (or Gabor wavelet) has been shown to be very efficient. Manjunath and Ma [15] have shown that image retrieval using Gabor features outperforms that using other transform features. Therefore, in this research we use Gabor wavelet transform as our technique to extract the texture features. Gabor wavelet and its implementation for texture feature extraction are detailed in Chapter 4.

4.2 GLCM (Gray Level Co-occurrence Matrix)

Texture is one of the important characteristics used in identifying objects or regions of interest in an image. Texture contains important information about the structural arrangement of surfaces. The textural features based on gray-tone spatial dependencies have a general applicability in image classification. The three fundamental pattern elements used in human - interpretation of images are spectral, textural and contextual features. Spectral features describe the average tonal variations in various bands of the visible and/or infrared portion of an electromagnetic spectrum. Textural features contain information about the spatial distribution of tonal variations within a band. The fourteen textural features proposed by Haralick et al contain information about image texture characteristics such as homogeneity, gray-tone linear dependencies, contrast, number and nature of boundaries present and the complexity of the image. Contextual features contain information derived from blocks of pictorial data surrounding the area being analyzed. Haralick et all first introduced the use of co-occurrence probabilities using GLCM for extracting various texture features. GLCM is also called as Gray level Dependency Matrix. It is defined as "A two dimensional histogram of gray levels for a pair of pixels, which are separated by a fixed spatial relationship." GLCM of an image is computed using a displacement vector d, defined by its radius δ and orientation θ. Consider a

4x4 image represented by figure 1a with four gray-tone values 0 through 3. A generalized GLCM for that image is shown in figure 1b where # (i , j) stands for number of times gray 8 tones i and j have been neighbors satisfying the condition stated by displacement vector d.

0	**0**	**1**	**1**
0	**0**	**1**	**1**
0	**2**	**2**	**2**
2	**2**	**3**	**3**

Fig 4.2.1 Test Image

Gray tone	**0**	**1**	**2**	**3**
0	**#(0,0)**	**#(0,1)**	**#(0,2)**	**#(0,3)**
1	**#(1,0)**	**#(1,1)**	**#(1,2)**	**#(1,3)**
2	**#(2,0)**	**#(2,1)**	**#(2,2)**	**#(2,3)**
3	**#(3,0)**	**#(3,1)**	**#(3,2)**	**#(3,3)**

Fig 4.2.2 General form of GLCM

The four GLCM for angles equal to 0°, 45°, 90° and 135° and radius equal to 1 are shown in figure 4.3.1 to 4.3.4.

4	**2**	**1**	**0**
2	**4**	**0**	**0**
1	**0**	**6**	**1**
0	**0**	**1**	**2**

Fig 4.3.1 GLCM for δ**=1 and** θ**=0°**

6	**0**	**2**	**0**
0	**4**	**2**	**0**
2	**2**	**2**	**2**
0	**0**	**2**	**0**

Fig 4.3.2 GLCM for δ**=1 and** θ**=45°**

4	**1**	**0**	**0**
1	**2**	**2**	**0**
0	**2**	**4**	**1**
0	**0**	**1**	**0**

Fig 4.3.3 GLCM for δ**=1 and** θ**=90°**

2	**1**	**3**	**0**
1	**2**	**1**	**0**
3	**1**	**0**	**2**
0	**0**	**2**	**0**

Fig 4.3.4 GLCM for δ**=1 and** θ**=0°**

These are symmetric matrices hence evaluation of either upper or lower triangle serves the purpose. Frequency normalization can be employed by dividing value in each cell by the total number of pixel pairs possible. Hence the normalization factor for 0° would be $(N_x-1) \times N_y$ where N_x represents the width and N_y represents the height of the image. The quantization level is an equally important consideration for determining the co-occurrence texture features. Also, neighboring co-occurrence matrix elements are highly correlated as they are measures of similar image qualities. Each of these factors is discussed ahead in detail.

Choice of radius δ

Various research studies show δ values ranging from 1, 2 to 10. Applying large displacement value to a fine texture would yield a GLCM that does not capture detailed textural information. From the previous studies, it has been concluded that overall classification accuracies with δ = 1, 2, 4, 8 are acceptable with the best results for δ = 1 and 2. This conclusion is justified, as a pixel is more likely to be correlated to other closely located pixel than the one located far away. Also, displacement value equal to the size of the texture element improves classification.

Choice of angle θ

Every pixel has eight neighboring pixels allowing eight choices for θ, which are 0°, 45°, 90°, 135°, 180°, 225°, 270° or 315°. However, taking into consideration the definition of GLCM, the co-occurring pairs obtained by choosing θ equal to 0° would be similar to those obtained by choosing θ equal to 180°. This concept extends to 45°, 90° and 135° as well. Hence, one has four choices to select the value of θ. Sometimes, when the image is isotropic, or directional information is not required, one can obtain isotropic GLCM by integration over all angles.

Choice of quantized gray levels (G)

The dimension of a GLCM is determined by the maximum gray value of the pixel. Number of gray levels is an important factor in GLCM computation. More levels would mean more accurate extracted textural information, with increased computational costs. The computational complexity of GLCM method is highly sensitive to the number of gray levels and is proportional to $O(G^2)$.

Thus for a predetermined value of G, a GLCM is required for each unique pair of δ and θ. GLCM is a second-order texture measure. The GLCM's lower left triangular matrix is always a reflection of the upper right triangular matrix and the diagonal always contains even numbers. Various GLCM parameters are related to specific first-order statistical concepts. For instance, contrast would mean pixel pair repetition rate, variance would mean spatial frequency detection etc. Association of a textural meaning to each of these parameters is very critical. Traditionally, GLCM is dimensioned to the number of gray levels G and stores the co-occurrence probabilities g_{ij}.To determine the texture features, selected statistics are applied to each GLCM by iterating through the entire matrix. The textural features are based on statistics which summarize the relative frequency distribution which describes how often one gray tone will appear in a specified spatial relationship to another gray tone on the image.

Following notations are used to explain the various textural features:

$g_{ij} = (i, j)^{th}$ entry in GLCM

g $_x$ (i) = ith entry in marginal probability matrix obtained by summing rows of $g_{ij} = \sum_{j=1}^{N_g} g(i,j)$

N_g = Number of distinct gray levels in the image

$$\sum_i = \sum_{i=1}^{N_g}$$

$$\sum_j = \sum_{j=1}^{N_g}$$

$$g_{ij} = \sum_{i=1}^{N_g} g(i,j)$$

$$g_y = \sum_{i=1}^{N_g} g(i,j)$$

$$g_{x+y} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i,j)\, where\;\; i + j = k = 1,2,3,4, \ldots., N_g$$

$$g_{x-y} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} g(i,j)\, where\;\; |i - j| = k = 1,2,3,4, \ldots., N_g - 1$$

Few of the common statistics applied to co-occurrence probabilities are discussed ahead.

1) Energy:

$$Energy(ene) = \sum_i \sum_j g_{ij}^2$$

This statistic is also called Uniformity or Angular second moment. It measures the textural uniformity that is pixel pair repetitions. It detects disorders in textures. Energy reaches a maximum value equal to one. High energy values occur when the gray level distribution has a constant or periodic form. Energy has a normalized range. The GLCM of less homogeneous image will have large number of small entries.

2) Entropy:

$$Entropy(ent) = -\sum_i \sum_j g_{ij} \log_2 g_{ij}$$

This statistic measures the disorder or complexity of an image. The entropy is large when the image is not texturally uniform and many GLCM elements have very small values. Complex textures tend to have high entropy. Entropy is strongly, but inversely correlated to energy.

3) Contrast:

$$Contrast(con) = \sum_{i} \sum_{j} (i-j)^2 \, g_{ij}$$

This statistic measures the spatial frequency of an image and is difference moment of GLCM. It is the difference between the highest and the lowest values of a contiguous set of pixels. It measures the amount of local variations present in the image. A low contrast image presents GLCM concentration term around the principal diagonal and features low spatial frequencies.

4) Variance:

$$Variance(var) = \sum_{i} \sum_{j} (i-\mu)^2 \, g_{ij} \; where\ \mu\ is\ the\ mean\ of\ g_{ij}$$

This statistic is a measure of heterogeneity and is strongly correlated to first order statistical variable such as standard deviation. Variance increases when the gray level values differ from their mean.

5) Homogeneity:

$$Homeogenity(ene) = \sum_{i} \sum_{j} \frac{1}{1+(i-j)^2} \, g_{ij}$$

This statistic is also called as Inverse Difference Moment. It measures image homogeneity as it assumes larger values for smaller gray tone differences in pair elements. It is more sensitive to the presence of near diagonal elements in the GLCM. It has maximum value when all elements in the image are same. GLCM contrast and homogeneity are strongly, but inversely, correlated in terms of equivalent distribution in the pixel pairs population. It means homogeneity decreases if contrast increases while energy is kept constant.

6) Correlation:

$$Correlation(cor) = \frac{\sum_i \sum_j (ij) g_{ij} - \mu_x \, \mu_y}{\sigma_x \, \sigma_y}$$

where μ_x, μ_y, σ_x and σ_y are the means and standard deviations of g_x and g_y

The correlation feature is a measure of gray tone linear dependencies in the image. The rest of the textural features are secondary and derived from those listed above.

7) Sum Average:

$$Sum\ Average = \sum_{i=1}^{2N_g} i\ g_{x+y}(i)$$

8) Sum Entropy:

$$Sum\ Entropy(se) = -\sum_{i=2} g_{x+y}(i) \log\{g_{x+y}(i)\}$$

9) Sum Variance:

$$Sum\ Variance(sv) = \sum_{i=2} (i - sa)^2\ g_{x+y}(i)$$

10) Difference Variance:

$$Difference\ Variance = Variance\ of\ g_{x+y}$$

11) Difference Entropy:

$$Difference\ Entropy = -\sum_{i=1}^{N_{g-1}} g_{x-y}(i) \log\{g_{x-y}(i)\}$$

12) Maximum Correlation Coefficient:

Maximum Correlation Coefficient (MCC) = (second largest eigen value of Q $)^{0.5}$

$$where\ Q(i,j) = \sum_k \frac{g(i,k)g(j,k)}{g_x(i)g_y(k)}$$

13),14) Information Measures of Correlation:

$$Information\ measure\ of\ correlation\ 1(IMC1) = \frac{HXY - HXY1}{max\{HX, HY\}}$$

$$Information\ measure\ of\ correlation\ 2(IMC2) = \sqrt{1 - (HXY2 - HXY)^{-2}}$$

$$HXY = -\sum_i \sum_j g_{ij} \log_2 g_{ij}$$

Where HX and HY are entropies of $g_x$and g_y

$$HXY1 = -\sum_i \sum_j g_{ij} \log_2\{g_x(i)g_y(j)\}$$

$$HXY2 = -\sum_i \sum_j g_x(i)g_y(j) \log_2\{g_x(i)g_y(j)\}$$

The question what exactly the textural features represent from a human perception point of view can be a subject for a thorough experimentation. Of the textural features described above, the angular second moment, the entropy, the sum entropy, the difference entropy, the information measure of correlation and the maximal correlation features have the invariance property. Earlier studies cite “Energy” and “Contrast” to be the most efficient parameters for discriminating different textural patterns. The general thumb rules used in the selection of the texture features can be stated as follows:

• Energy is preferred to entropy as its values belong to normalized range.

• Contrast is associated with the average gray level difference between neighbor pixels. It is similar to variance however preferred due to reduced computational load and its effectiveness as a spatial frequency measure.

• Energy and contrast are the most significant parameters in terms of visual assessment and computational load to discriminate between different textural patterns.

4.3 Texture Features

Image textures are defined as images of natural textured surfaces and artificially created visual patterns, which approach, within certain limits, these natural objects. Image sensors yield additional geometric and optical transformations of the perceived surfaces, and these transformations should not affect a particular class of textures the surface belongs. It is almost impossible to describe textures in words, although each human definition involves various informal qualitative structural features, such as fineness - coarseness, smoothness, granularity, lineation, directionality, roughness, regularity - randomness, and so on.

These features, which define a spatial arrangement of texture constituents, help to single out the desired texture types, e.g. fine or coarse, close or loose, plain or twilled or ribbed textile fabrics. It is difficult to use human classifications as a basis for formal definitions of image textures, because there is no obvious ways of associating these features, easily perceived by human vision, with computational models that have the goal to describe the textures. Nonetheless, after several decades of research and development of texture analysis and synthesis, a variety of computational characteristics and properties for indexing and retrieving textures have been found. The textural features describe local arrangements of image signals in the spatial domain or the domain of Fourier or other spectral transforms. In many cases, the textural features follow from a particular random field model of textured images.

Texture features are extracted from gray level co-occurrence matrix. Texture features are described above in the section4.2.In this project features extracted are Energy, Entropy, Homogeneity, Contrast and Variance. So, these features form the feature vector.

4.3.1 Global Retrieval

In global retrieval, GLCM for whole image is calculated. Texture features are extracted from the resulted GLCM. The features extracted forms the feature vector. During retrieval, query image GLCM is calculated and texture features are extracted from it. This forms the query image feature vector. For a match a similarity measure should be computed between query image and all other images in the image dataset.

4.3.2 Local Retrieval

In local retrieval, image is segmented into various number of regions. GLCM is calculated for every region and features mentioned above in section 4.2 are extracted from the calculated GLCM. So, for every region a count of 5 features forms the feature vector. Similarly, during retrieval, query image is segmented into regions and features are extracted from every region .A match is found based on the similarity between the query image and image dataset.

4.4 Feature Vector

In global retrieval, five (5) texture features which are extracted forms the feature vector. In local retrieval, every region has 5 features i.e. the feature vector consists of 5*(no. of regions into which image is segmented).

$F_c=\{ene_1,ent_1,var_1,cor_1,con_{1,}\ldots\ldots\ldots\ldots\ldots\ ene_r\ ,ent_r\ ,var_r\ ,cor_r\ ,con_r\}$

(1 <= r) r represents the region, ene is energy, ent is entropy, var is variance, con is contrast, cor is correlation. When considering the whole image r=1.

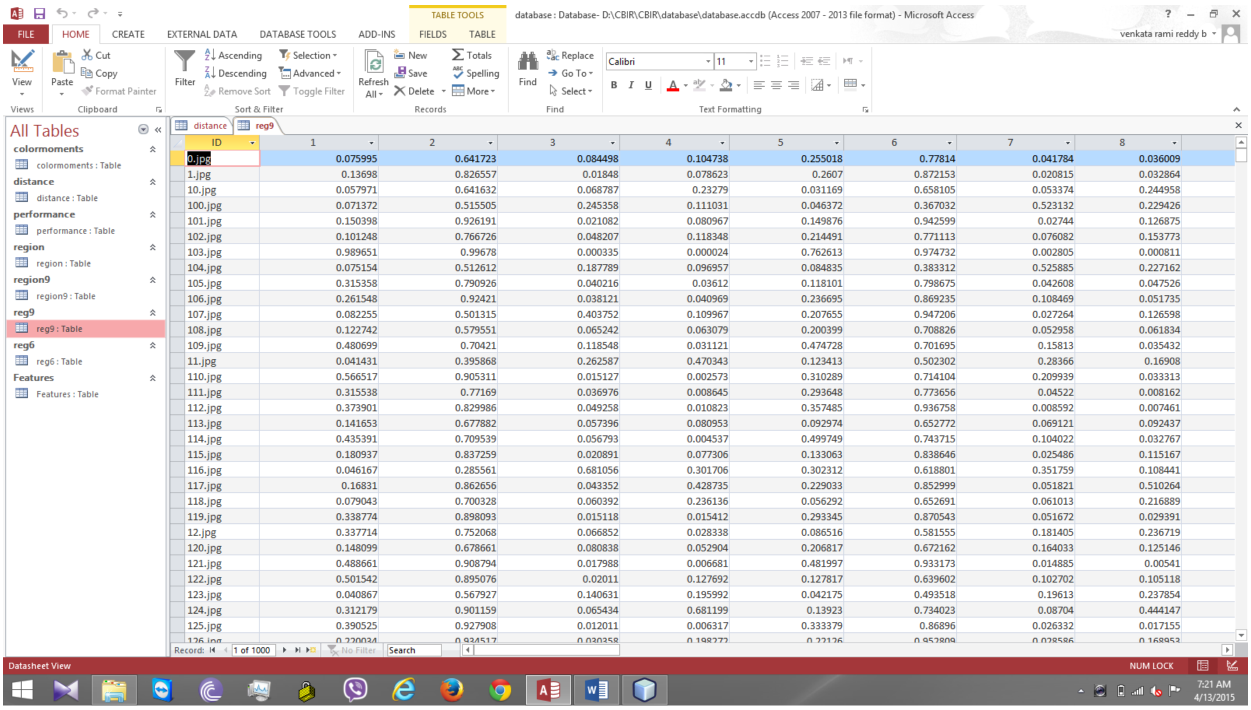
Chapter 5

Image Retrieval Methodology

5.1 Offline Feature Extraction

Features are extracted from all the images in the dataset. These features are stored in the database. The attributes of the features vector may have different ranges (one of very small value and one of very high value), therefore a normalization method should be applied to make all the texture features have the same effect in measuring image similarity. The Min-Max algorithm is employed as a normalization technique; it performs a linear transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of the attribute, the Min-Max normalization maps a value, v, of A to $\tilde{v}$ in the range [0, 1] by computing:

$\tilde{v} = \frac{v - min_A}{max_A - min_A}$ Weka is used to normalize the feature vectors of all images in dataset. So, the color features and texture features in normalized form are available in the database. Following is the sample database.



ID	1	2	3	4	5	6	7	8
0.jpg	0.075995	0.641723	0.084498	0.104738	0.255018	0.77814	0.041784	0.036009
1.jpg	0.13698	0.826557	0.01848	0.078623	0.2607	0.872153	0.020815	0.032864
10.jpg	0.057971	0.641632	0.068787	0.23279	0.031169	0.658105	0.053374	0.244958
100.jpg	0.071372	0.515505	0.245358	0.111031	0.046372	0.367032	0.523132	0.229426
101.jpg	0.150398	0.926191	0.021082	0.080967	0.149876	0.942599	0.02744	0.126875
102.jpg	0.101248	0.766726	0.048207	0.118348	0.214491	0.771113	0.076082	0.153773
103.jpg	0.989651	0.99678	0.000335	0.000024	0.762613	0.974732	0.002805	0.000811
104.jpg	0.075154	0.512612	0.187789	0.096957	0.084835	0.383312	0.525885	0.227162
105.jpg	0.315358	0.790926	0.040216	0.03612	0.118101	0.798675	0.042608	0.047526
106.jpg	0.261548	0.92421	0.038121	0.040969	0.236695	0.869235	0.108469	0.051735
107.jpg	0.082255	0.501315	0.403752	0.109967	0.207655	0.947206	0.027264	0.126598
108.jpg	0.122742	0.579551	0.065242	0.063079	0.200399	0.708826	0.052958	0.061834
109.jpg	0.480699	0.70421	0.118548	0.031121	0.474728	0.701695	0.15813	0.035432
11.jpg	0.041431	0.395868	0.262587	0.470343	0.123413	0.502302	0.28366	0.16908
110.jpg	0.566517	0.905311	0.015127	0.002573	0.310289	0.714104	0.209939	0.033313
111.jpg	0.315538	0.77169	0.036976	0.008645	0.293648	0.773656	0.04522	0.008162
112.jpg	0.373901	0.829986	0.049258	0.010823	0.357485	0.936758	0.008592	0.007461
113.jpg	0.141653	0.677882	0.057396	0.080953	0.092974	0.652772	0.069121	0.092437
114.jpg	0.435391	0.709539	0.056793	0.004537	0.499749	0.743715	0.104022	0.032767
115.jpg	0.180937	0.837259	0.020891	0.077306	0.133063	0.838646	0.025486	0.115167
116.jpg	0.046167	0.285561	0.681056	0.301706	0.302312	0.618801	0.351759	0.108441
117.jpg	0.16831	0.862656	0.043352	0.428735	0.229033	0.852999	0.051821	0.510264
118.jpg	0.079043	0.700328	0.060392	0.236136	0.056292	0.652691	0.061013	0.216889
119.jpg	0.338774	0.898093	0.015118	0.015412	0.293345	0.870543	0.051672	0.029391
12.jpg	0.337714	0.752068	0.066852	0.028338	0.086516	0.581555	0.181405	0.236719
120.jpg	0.148099	0.678661	0.080838	0.052904	0.206817	0.672162	0.164033	0.125146
121.jpg	0.488661	0.908794	0.017988	0.006681	0.481997	0.933173	0.014885	0.00541
122.jpg	0.501542	0.895076	0.02011	0.127692	0.127817	0.639602	0.102702	0.105118
123.jpg	0.040867	0.567927	0.140631	0.195992	0.042175	0.493518	0.19613	0.237854
124.jpg	0.312179	0.901159	0.065434	0.681199	0.13923	0.734023	0.08704	0.444147
125.jpg	0.390525	0.927908	0.012011	0.006317	0.333379	0.86896	0.026332	0.017155

Fig 5.1 Offline Database

5.2 On-line Image Retrieval

During image retrieval, query image is selected. Features are extracted from query image so that a feature vector for query image is obtained. The features include color features using color moments and texture features using GLCM. Let T_Q and T_I denote the feature vector of a query image and an image in the database respectively. To test the similarity between a query image *Q* and a database images *I* based on their color or texture feature we proposed to use the Euclidean distance for its simplicity.

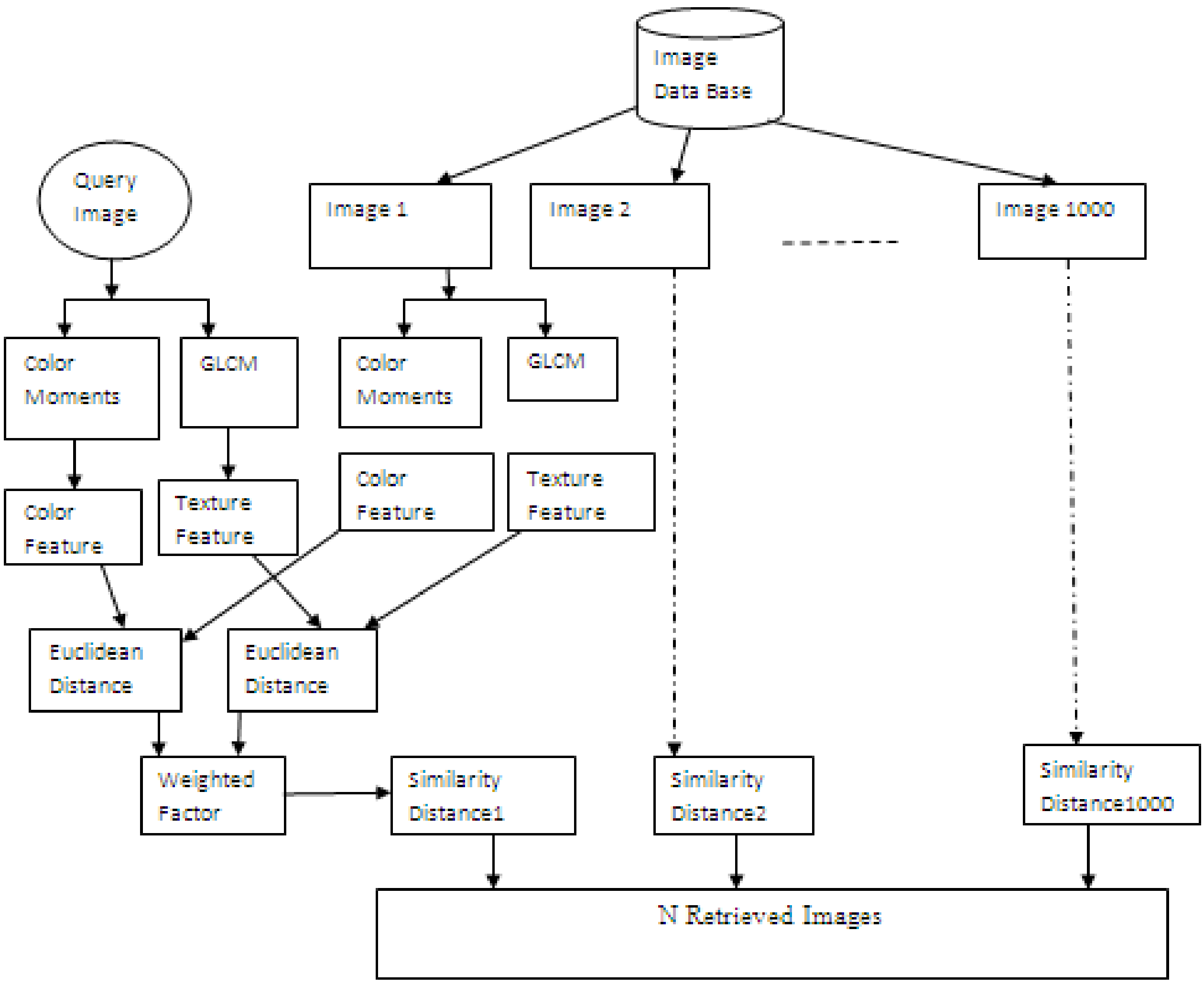


Fig 5.2 Online Image Retrieval

We define the distance between the two vectors denoted as $d_T(Q, I)$ by the Euclidean distance as:

$$d_T(Q, I) = \sqrt{\sum_{i=1}^{n} \left(T_{Qi} - T_{Ii}\right)^2} \quad where\ n\ is\ the\ size\ of\ feature\ vector$$

Images for which the distance between the images with the query image is least will be the most similar image.

Here is an example of a query image. The features of query image are extracted. The feature vector of query image is shown below.

Fig 5.3 Query Image

Features here considered are combination of Color and texture in RGB color space.

Feature Vector of the query image is as follows:

Image	M r	M g	M b	Sdr	Sdg	SDb	Skr	Skg	Skb	Ene	H	Con	Var
330.jpg	0.390	0.332	0.369	0.626	0.862	0.896	0.648	0.905	0.907	0.171	0.669	0.268	0.142

Table 5.1 Feature Vector of Query Image

where

- M r is the mean of red-channel
- M g is the mean of green-channel
- M b is the mean of blue-channel
- Sdr is the standard deviation of red-channel
- Sdg is the standard deviation of green-channel
- Sdb is the standard deviation of blue-channel
- Skr is the Skewness of red-channel
- Skg is the Skewness of green-channel

- Skrb is the Skewness of blue-channel
- Ene is energy
- Hom is homogeneity

The below table shows some images, their corresponding feature vectors and distance measured between the feature vector of image and feature vectors of the query image

Image	M r	M g	M b	Sdr	Sdg	SDb	Skr	Skg	Skb	Ene	H	Con	Var	Ent	Dist
100.jpg	0.42	0.42	0.45	0.48	0.65	0.71	0.70	0.76	0.78	0.08	0.40	0.52	0.12	0.82	1.788
305.jpg	0.33	0.38	0.42	0.53	0.74	0.72	0.76	0.81	0.74	0.18	0.58	0.42	0.12	0.55	1.792
313.jpg	0.41	0.40	0.47	0.57	0.92	0.95	0.31	0.84	0.79	0.11	0.61	0.36	0.17	0.68	1.721
315.jpg	0.29	0.32	0.37	0.62	0.82	0.82	0.84	0.93	0.88	0.27	0.59	0.57	0.14	0.70	1.783
326.jpg	0.35	0.38	0.27	0.61	0.75	0.72	0.74	0.80	0.88	0.17	0.58	0.43	0.12	0.46	1.683
354.jpg	0.46	0.37	0.43	0.58	0.71	0.86	0.25	0.83	0.87	0.12	0.64	0.26	0.11	0.81	1.729
368.jpg	0.44	0.37	0.42	0.41	0.83	0.91	0.26	0.77	0.84	0.13	0.63	0.30	0.12	0.38	1.766
372.jpg	0.42	0.34	0.32	0.72	0.91	0.83	0.26	0.92	0.88	0.18	0.56	0.65	0.16	0.53	1.387
377.jpg	0.49	0.30	0.34	0.62	0.70	0.72	0.19	0.84	0.86	0.12	0.56	0.33	0.08	0.65	1.663

Table 5.2 Demonstration of distance measurement

The following table shows the images that are most similar on sorting the distances. It also shows the image corresponding to the image name

Img Name	Distance	Image
330.jpg	0.0	
372.jpg	1.3870851050746669	
377.jpg	1.6633258119707033	
326.jpg	1.683497933365527	
313.jpg	1.7219367169440345	
354.jpg	1.7290968111436098	
368.jpg	1.766829481295238	
315.jpg	1.783910679097751	
100.jpg	1.7880449846060922	
305.jpg	1.792081005885058	

Table 5.3 Similarity order of images after sorting distances

Chapter 6

Test Cases and Results

6.1 WANG Dataset

The database we used in our evaluation is WANG database. The WANG database is a sub-set of the Corel database of 1000 images, which have been manually selected to be a database of 10 classes of 100 images each.

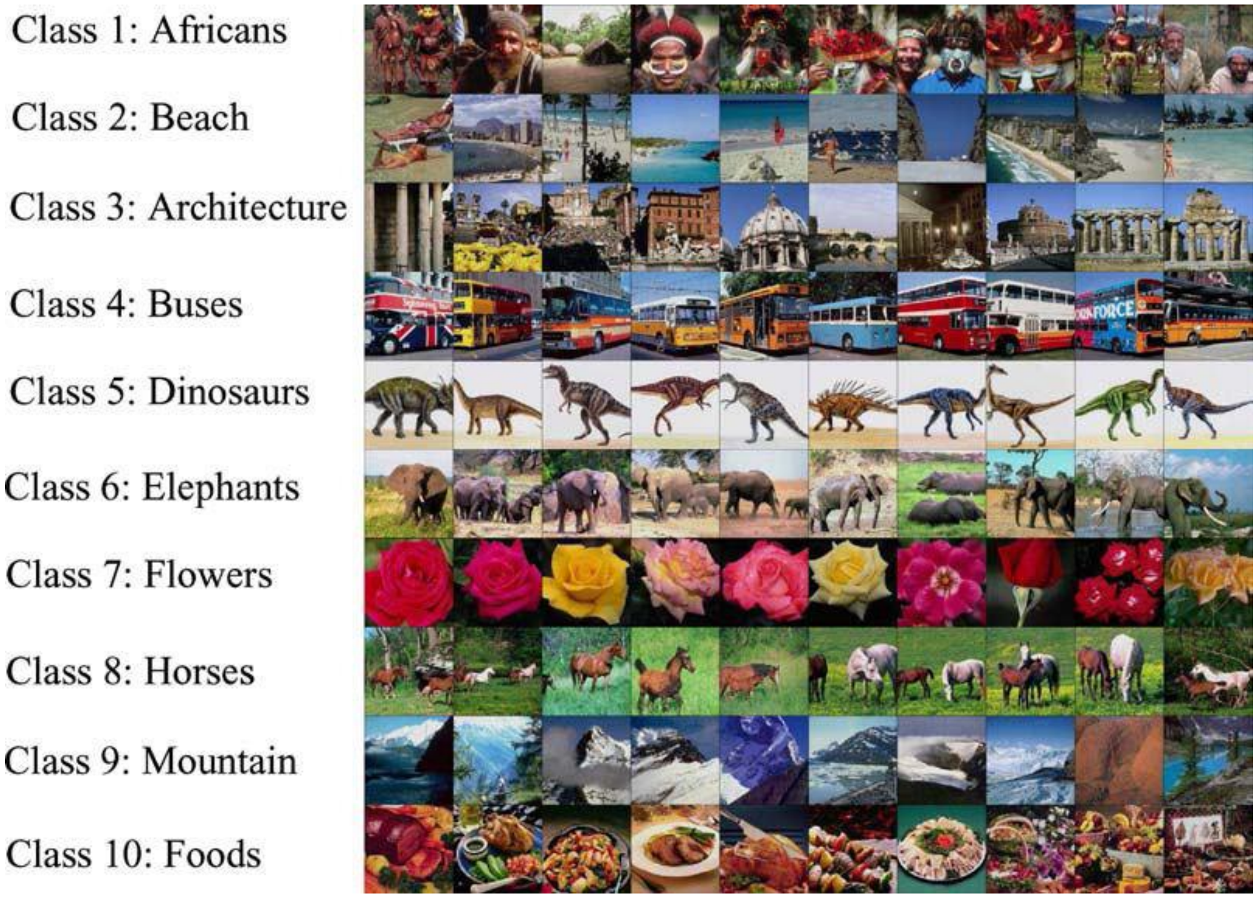


Fig 6.1 Sample WANG Dataset

The images are subdivided into 10 classes, such that it is almost sure that a user wants to find the other images from a class if the query is from one of these 10 classes. This is a major advantage of this database because due to the given classification it is possible to evaluate retrieval results. The images are of size 384 × 256 or 256 × 384 pixels, Figure shows 10 sample images in each image class.

6.2 Implementation Environment

The image retrieval system is implemented in JAVA. During the implementation, we use a platform of Intel Core 2 Due Processing power of 2.4 GHz CPU with 4GB RAM. WEKA is used in-order to normalize the feature vector. Microsoft Access in-order to store the feature vectors of database of images.

6.3 Precision and Recall

The level of retrieval accuracy achieved by a system is important to establish its performance. If the outcome is satisfactory and promising, it can be used as a standard in future research works. In IRBC, precision-recall is the most widely used measurement method to evaluate the retrieval accuracy. We have found some recent literature that uses this pair to measure the retrieval performance.

Precision, P, is defined as the ratio of the number of retrieved relevant images to the total number of retrieved images.

$$\boldsymbol{P = \frac{No. of\ relevant\ images\ retrieved}{No\ . of\ images\ retrieved}}$$

Let the number of all retrieved images be n, and let r be the number of relevant images according to the query then the precision value is: P = r / n. Precision P measures the accuracy of the retrieval.

Recall, R, is defined as the ratio of the number of retrieved relevant images to the total number of relevant images in the whole database.

$$\boldsymbol{R = \frac{No. of\ relevant\ images\ retrieved}{No\ . ofrelevant\ images\ in\ the\ database}}$$

Let r be the number of relevant images among all retrieved images according to the query, and M be the number of all relevant images to the query in the whole database then the Recall value is: R = r / M. Recall R measures the robustness of the retrieval.

To evaluate the system top 100 images are retrieved. Precision is calculated by considering the top 10 images i. e… how many images are of the same category in the top 10. Recall is calculated by considering all the 100 that are retrieved i.e. how many matches in the top 100 that are retrieved…

So, here

Precision (n=10%) = (No. of relevant images/10)*100%

Recall (n=10%) = (No. of relevant images/No. of relevant images in database)*100%

No. of relevant images in database=100 for every category

In order to assess the discriminating power of the techniques proposed, we carried out the experiments based on Color Moments- in RGB(CMRGB) color space and HSV color space (CMHSV).These are done for whole image and also image divided into 6 parts as well as 9 parts,

Only texture features(GLCM) this is also done for whole image and image divided into parts, combining both color and texture features Globally and locally (CMRGB+GLCM), (CMHSV +GLCM).

Some of the results using the same query image of Fig. 6.2.1 are shown in Fig. 6. From Table 4. , it is seen that the average precision (%) based on (CMRGB+GLCM) is 61.0 and the average precision (%) based on (CMW+GTF) is 58.2. Thus the proposed method demonstrates clearly that our encoding of spatial information in the color index from different regions of the image significantly increases the discriminating power compared to the color moment (based on whole image) + GLCM texture features in which color moments are extracted from the entire image. It is also seen that the value of the average precisions (%) based on single features i.e. only Global texture features or only Color moments are less than the average precisions (%) of combined features of color moments and GLCM texture features as shown in Table 4. and Table 5. This also shows that there is considerable increase in retrieval efficiency when both color and texture features are combined for IRBC.

6.4 Image Retrieval

A query image is given in-order to test the system .The features of query image are extracted based on the consideration. In the first case only color feature is used for retrieval. In the second case only texture feature is considered. To improve the efficiency both the color and texture features are considered. The results in the above mentioned cases are shown below.

6.4.1 Color Moments

The retrieval is done considering global features and local features. In case of global features whole image is considered. Local features are calculated by dividing an image into regions and for every region features are calculated.

Results for the following query images are shown below

Query Image is:

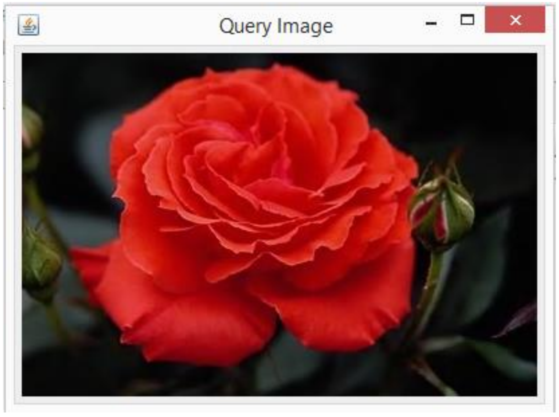


Fig 6.2.1 Query Image1

Fig 6.2.2 Query Image2

The top10 images are shown here using different techniques:

Fig 6.3.1 Results using only Color Features for Query Image1

Observation on above results:

The above figure shows the results obtained by using Color Moment features (RGB and HSV values) to find the images. It also shows the variation in results while using global, 6region and 9region comparisons. We obtained 10 similar images of roses but of this only 3 roses are of same color and the remaining results contains images of roses but with different colors with respect to query image by using global Color Moments. Similarly we are getting all the 10 images which are similar to query image in 6 and 9 regions cases with variation in color of roses.

	RGB			HSV	
Global	Local6	Local9	Global	Local6	Local9

Fig 6.3.2 Results using only Color Features for Query Image

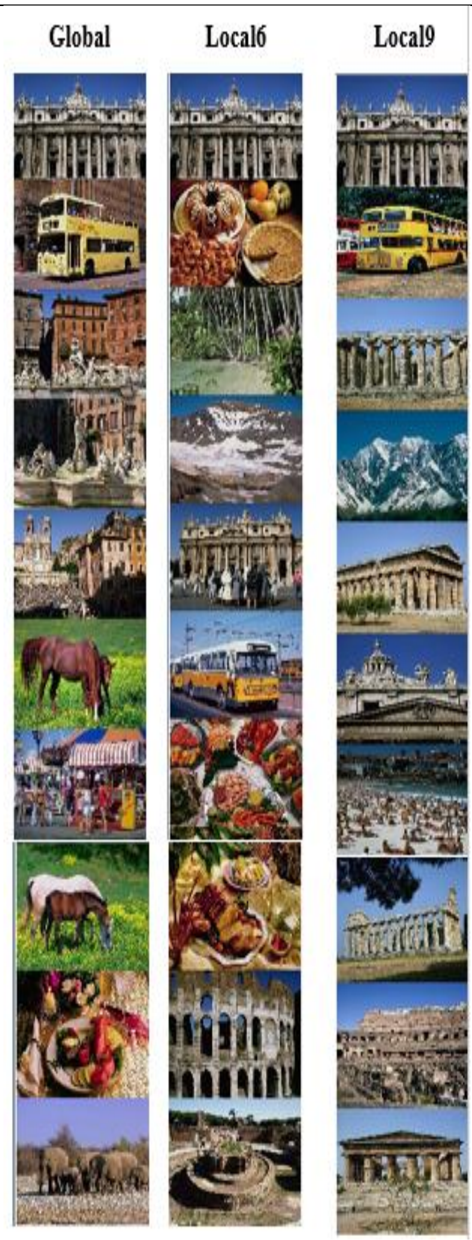
Observation on above results:

The above figure shows the results obtained by using Color Moment features (RGB and HSV values) to find the images. It also shows the variation in results while using global, 6region and 9region comparisons. We obtained 5 similar images by using global RGB Color Moments, 4 similar images by using 6region RGB Color Moments and 4 similar images by using 9region RGB Color Moments. The number of similar images remained same even though 6region and 9region Color Moments are applied. We obtained 6 similar images by using global HSV Color Moments, 5 similar images by using 6region HSV Color Moments and 5 similar images by using 9region HSV Color Moments. In this case there is a decrease in the number of similar images from global Moments to 6region Moments and 9region Moments.

From this observation it is understood that for query images like the above, global feature case is useful for getting more similar images compared to 6 and 9region case.

6.4.2 Texture

Only if texture features are constructed the following are the results:

GLCM for Query 1	**GLCM for Query 2**
	
Fig 6.4.1 Considering Texture Features Query1	Fig 6.4.2 Considering texture features Query2

Observation on the above results:

The above figures shows the results obtained by using Texture Moment features (GLCM Matrix) to find the similar images. It also shows the variation in results while using global, 6region and 9region comparisons.

Query1:

We obtained 9 similar images by using global Texture Moments, 9 similar images by using 6region Texture Moments and 9 similar images by using 9region Texture Moments. The number of similar images remained same in results from global, 6region and 9region Texture Moments but there is variation in the colors of roses.

Query2:

We obtained 4 similar images by using global Texture Moments, 4 similar images by using 6region Texture Moments and 7 similar images by using 9region Texture Moments. . The number of similar images in results increased from global, 6region to 9region Texture Moments.

6.4.3 Combining Color and Text features

The retrieval result using only single feature may be inefficient. It may either retrieve images not similar to query image or may fail to retrieve images similar to query image. Hence, to produce efficient results, we use combination of color and texture features. The similarity between query and target image is measured from two types of characteristic features which includes color and texture features. Two types of characteristics of images represent different aspects of property. So, during similarity measure, appropriate weights are considered to combine the features. The distance between the query image and the image in the database is calculated as follows:

$$d = w1*d1+w2*d2$$

Here, w1 is the weight of the color features, w2 is the weight of the texture features and d1 and d2 are the distances calculated using color moment and texture features. Experiments show that better retrieval performances are achieved when we set w1=0.80 and w2=0.20. The weight factor of color feature distance is higher than the weight factor of texture feature distance because our database consists of mostly natural images.

The above distance, d is calculated between the query image and all the images in the database and it is sorted in ascending order. The image corresponding to the first element of d is the most similar image compared with the query image. The first 10 top most similar images are then displayed.

Color Moments +Texture Feature(RGB) **Color Moments +Texture Feature(HSV)**

Global **Local6** **Local9** **Global** **Local6** **Local9**

Fig 6.5.1 Results on combining the features for Query Image1

Observation on the above result:

The above figure shows the results obtained by combining Texture Moments
With RGB Color Moment features and HSV Color Moments respectively. It also shows the variation in results while using global, 6region and 9region comparisons.

We obtained 10 similar images when global RGB Color Moments are Combined with Global Texture Moments, 10 similar images when 6region RGB Color Moments are Combined with 6 region Texture Moments and 10 similar images when 9region RGB Color Moments are Combined with 9region Texture Moments. Results are same in case of HSV too.

Color Moments +Texture Feature(RGB) **Color Moments +Texture Feature(HSV)**

Global **Local6** **Local9** **Global** **Local6** **Local9**

Fig 6.5.2 Results on Combining the Features for Query Image2

Observation on the above results:

The above figure shows the results obtained by combining Texture Moments with RGB Color Moment features and HSV Color Moments respectively. It also shows the variation in results while using global, 6region and 9region comparisons.

We obtained 7 similar images when global RGB Color Moments are combined with Global Texture Moments, 3 similar images when 6region RGB Color Moments are Combined with 6 region Texture Moments and 4 similar images when 9region RGB Color Moments are combined with 9region Texture Moments.

We obtained 8 similar images when global HSV Color Moments are combined with Global Texture Moments, 7 similar images when 6region HSV Color Moments are Combined with 6 region Texture Moments and 6 similar images when 9region HSV Color Moments are combined with 9region Texture Moments.

Class	*CMRGB*			*CMHSV*			*GLCM*			*CMRGB+GLCM*			*CMHSV+GLCM*		
	G	***R6***	***R9***	***G***	***R6***	***R9***	***G***	***R6***	***R9***	***G***	***R6***	***R9***	***G***	***R6***	***R9***
Africans	90	80	70	90	100	90	40	30	40	70	90	70	100	90	90
Beach	90	80	90	80	70	90	70	60	60	90	90	90	70	90	90
Building	50	40	40	60	50	50	40	40	70	70	40	50	80	70	60
Buses	80	80	90	100	100	100	100	100	100	100	100	100	100	100	100
Dinosaur	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Elephant	60	80	90	70	60	90	60	50	90	60	80	80	90	60	90
Flower	100	100	100	100	100	100	90	90	90	100	100	100	100	100	100
Horse	100	100	100	100	100	100	70	90	70	100	100	100	100	100	100
Mountain	40	50	20	20	90	80	20	50	10	50	50	30	50	90	90
Food	90	80	60	80	70	80	10	10	30	90	90	50	80	80	80
Average Precision(%)	80	79	76	80	84	88	58	58	62	81	81	75	87	88	90

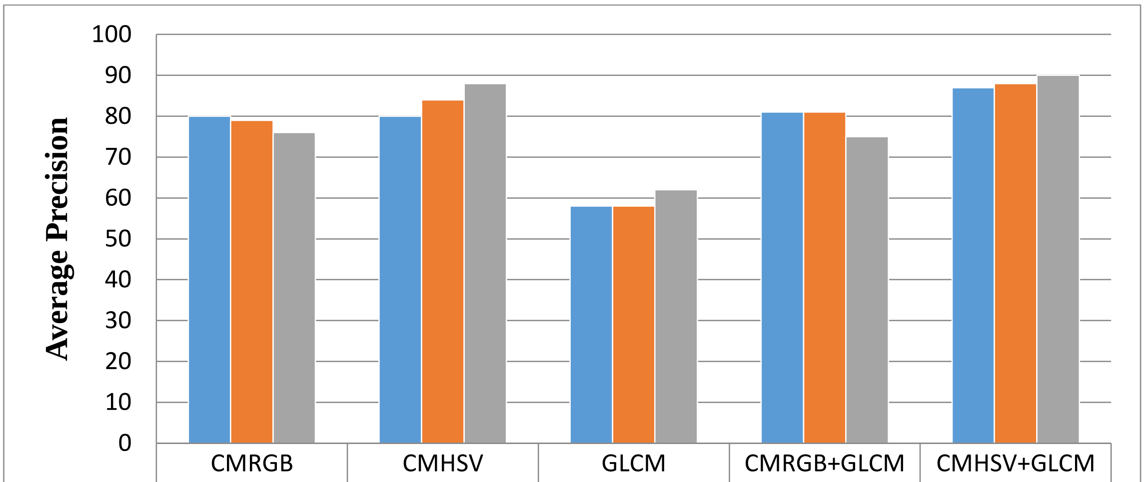
Table 6.1 Average Precision

The following table shows the average recall using different techniques:

Class	*CMRGB*			*CMHSV*			*GLCM*			*CMRGB+GLCM*			*CMHSV+GLCM*		
	G	***R6***	***R9***	***G***	***R6***	***R9***	***G***	***R6***	***R9***	***G***	***R6***	***R9***	***G***	***R6***	***R9***
Africans	41	45	38	55	51	49	14	21	21	41	39	35	50	44	46
Beach	47	50	47	50	48	43	30	35	35	51	52	54	51	51	48
Building	26	21	15	22	35	27	31	18	23	27	23	18	24	37	30
Buses	38	39	42	41	63	59	42	34	32	38	41	43	41	63	64
Dinosaur	92	89	88	84	82	83	97	97	98	97	91	92	96	89	90
Elephant	54	46	53	50	47	49	40	32	33	52	45	52	51	47	51
Flower	74	82	74	58	62	62	79	79	80	76	84	83	60	70	67
Horse	57	60	64	56	77	80	27	35	43	55	64	68	55	77	76
Mountain	18	31	27	42	57	58	21	28	23	21	31	31	42	54	56
Food	43	39	36	40	47	47	14	13	18	46	39	35	41	48	47
Avg Recall	49	50.2	48.4	49.8	56.9	55.7	39.5	39.2	40.6	50.4	50.9	51.1	50.5	58	57.5

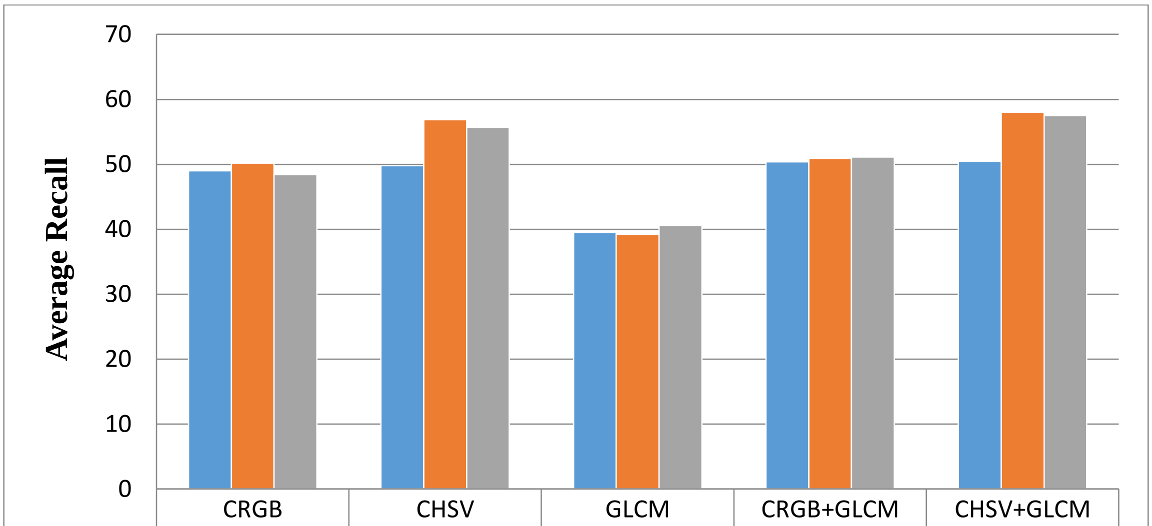
Table 6.2 Average Recall

Plots for Average Precision and Recall:



	CMRGB	CMHSV	GLCM	CMRGB+GLCM	CMHSV+GLCM
Global	80	80	58	81	87
Local6	79	84	58	81	88
Local9	76	88	62	75	90

Fig 6.6.1 Plot for Average Precision



	CRGB	CHSV	GLCM	CRGB+GLCM	CHSV+GLCM
Global	49	49.8	39.5	50.4	50.5
Local6	50.2	56.9	39.2	50.9	58
Local9	48.4	55.7	40.6	51.1	57.5

Fig 6.6.2 Plot for Average Recall

6.5 Output Screens

Fig 6.7.1 Query image

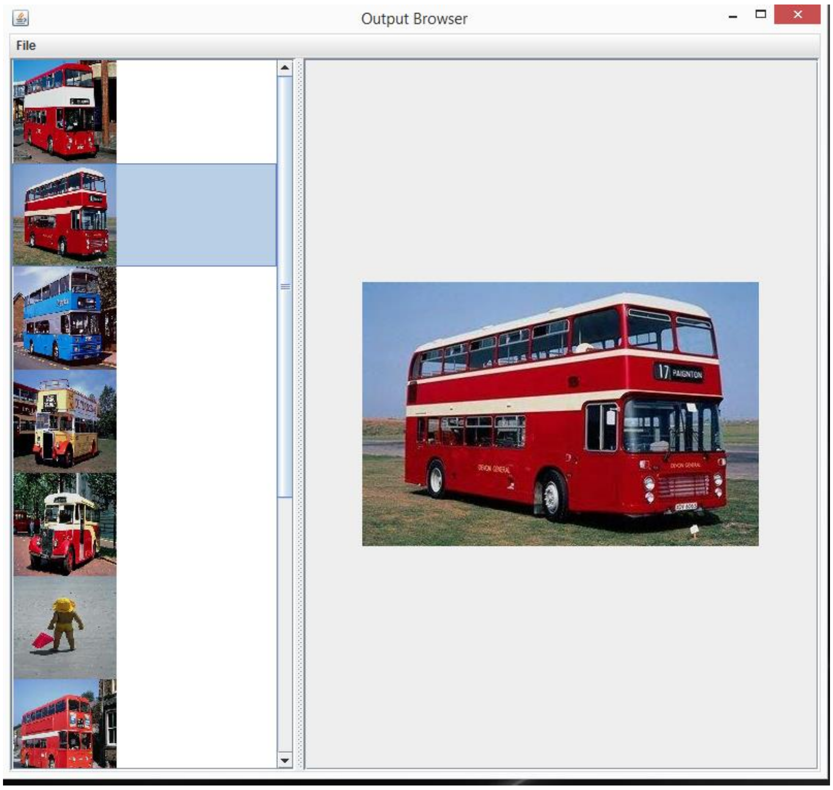


Fig 6.7.2 Output of similar images for the given query image

Figure 6.7.1 is the query image given to the system and the respective output for the query image is shown in figure 6.7.2 containing similar images.

Chapter 7

Conclusion

Conclusion

Image retrieval based on content is a challenging method of capturing relevant images from a large storage space. Although this area has been explored for decades, no technique has achieved the accuracy of human visual perception in distinguishing images. Whatever the size and content of the image database is, a human being can easily recognize images of same category.

From the very beginning of IRBC research, similarity computation between images used either region based or global based features. Global features extracted from an image are useful in presenting textured images that have no certain specific region of interest with respect to the user. Region based features are more effective to describe images that have distinct regions. Retrieval systems based on region features are computationally expensive because of the need of segmentation process in the beginning of a querying process and the need to consider every image region in similarity computation. In this research, we presented a content based image retrieval that introduces three alternatives to answer an image query, which are to use either region based, global based features, or a combination of them. We use GLCM, which is a powerful texture extraction technique either in describing the content of image regions or the global content of an image. Color moments as a region wise color feature taken as color similarity metric combined with GLCM texture features have been proved to give approximately as good retrieval results as that of global based retrieval systems. Precision and Recall values are high using segmentation process.

Chapter 8

Future Work

Future Work

The following developments can be made in the future:

1. To further improve the performance of the retrieval system, the study of taking shape features into account during similarity distance computation can be considered.
2. To obtain better performance, the system can automatically pre-classify the database into different semantic images (such as outdoor vs. indoor, landscape vs. cityscape, texture vs. non texture images) and develop algorithms that are specific to a particular semantic image class.
3. Demonstration of using different color and texture weights adaptively during retrieval has an effect on the retrieval results.

Bibliography

[1] R.Datta, D. Joshi, J.Li, J.Z. Wang, " Image retrieval: ideas, influences, and trends of the new age ", ACM Computing Surveys 40(2), 2008, pp. 1-60.

[2]V.N.Gudivada, and V.V.Raghavan, “Content based image retrieval systems", IEEE Computer, Vol. 28, No. 9, 1995, pp. 18-22.

[3] B.S. Manjunath, and W.Y. Ma, " Texture Features for browsing and retrieval of image data", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, 1996, pp. 837-842.

[4] Y.Rui, T.S. Huang, M. Ortega, S. Mehrotra, " Relevance feedback : a power tool for interactive content based image retrieval ",IEEE Circuits and Systems for Video Technology , Vol. 8, No. 5, 1998, pp. 644-655.

[5] D. Swets, and J. Weng, “Hierarchical discriminant analysis for image retrieval”, IEEE “PAMI, Vol. 21, No. 5, 1999, pp. 386-400.

[6] F. Long, H. Zhang, H. Dagan, and D. Feng, “Fundamentals of content based image retrieval,” in D. Feng, W. Siu, H. Zhang (Eds.): “Multimedia Information Retrieval and Management. Technological Fundamentals and Applications,” Multimedia Signal Processing Book, Chapter 1, Springer-Verlag, Berlin Heidelberg New York, 2003, pp.1-26.

[7] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, and P. Yanker, “Query by image and video content: The QBIC system,” IEEE Computer, vol. 28, no 9, pp.23-32, Sep. 1995. [8] A. Pentland, R. Picard, and S. Sclaroff , “Photobook: Content based manipulation of image databases,” International Journal of Computer Vision, vol.18, no 3, pp.233–254, June 1997.

[9] J. Smith and S. Chang, “Visualseek: A Fully Automated Content-Based Image Query System,” Proceedings of the 4th ACM international conference on Multimedia table of contents, Boston, Massachusetts, United States, Nov. 1996, pp. 87-98.

[10] A. Gupta, and R. Jain, “Visual information retrieval,” Comm. Assoc. Comp. Mach., vol. 40, no. 5, pp. 70–79, May. 1997.

[11] J. Li, J. Wang, and G. Wiederhold, “Integrated Region Matching for Image Retrieval,” In Proceedings of the 2000 ACM Multimedia Conference, Los Angeles, October 2000, pp. 147-156.

[12] J. Wang, J. Li, G. Wiederhold, “Simplicity: semantics-sensitive integrated matching for picture libraries,” IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 23, no. 9, pp. 947–963, Sep. 2001.

[13] S. Mukherjea, K. Hirata, and Y. Hara, “AMORE: A World Wide Web Image Retrieval Engine,” Proc. World Wide Web, vol. 2, no. 3, pp. 115-132, June. 1999.

[14] A. Natsev, R. Rastogi, and K. Shim, “WALRUS: A Similarity Retrieval Algorithm for Image Databases,” IEEE Trans. On Knowledge and Data Engineering, vol.16, pp. 301-318, Mar. 2004.

[15] R. Picard and T. Kabir, “Finding Similar Patterns in Large Image Databases,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 5, pp. 161-164, New York 1993.

[16] W. Ma and B. Manjunath, “Natra: A Toolbox for Navigating Large Image Databases,” Proc. IEEE Int'l Conf. Image Processing, Santa Barbara, 1997, pp. 568- 571.

[17] J. Wang, G. Wiederhold, O. Firschein, and X. Sha, “Content-Based Image Indexing and Searching Using Daubechies' Wavelets,” Int. J. Digital Libraries, vol. 1, no. 4, pp. 311-328, 1998.

[18] M. Ghahroudi, M. Sarshar, and R. Sabzevari, “A Novel Content-Based Image Retrieval Technique Using Tree Matching,” Proceedings of The World Congress on Engineering, pp1797-180, May 2008.

[19] C. Chiang, Y. Hung, H. Yang, and G. Lee, “Region-based image retrieval using color-size features of watershed regions,” Journal of Visual Communication and Image Representation , vol. 20, no. 3, pp. 167-177, April 2009.

[20] D. Zhang, "Improving Image Retrieval Performance by Using Both Color and Texture Features," In Proc. of IEEE 3rd International Conference on Image and Graphics (ICIG04), Hong Kong, China, Dec.18-20, 2004, pp.172-175.

[21] M. Sudhamani, and C. Venugopal, “Image Retrieval from Databases: an Approach using Region Color and Indexing Technique,” International Journal of Computer Science and Network Security, vol.8 no.1, pp. 54-64, January 2008.

[22] S. Nandagopalan, B. Adiga, and N. Deepak, “A Universal Model for Content-Based Image Retrieval,” Proc. of world academy of science, engineering and technology, vol. 36, pp. 659-662, DEC. 2008.

[23] A. M. Eskicioglu, “Quality measurement for monochrome compressed images in the past 25 years”, Acoustics, Speech and Signal Processing, IEEE International Conference, Volume 6, June 2000.

[24] R. M. Haralick, K. Shanmugam and I. Dinstein “Textural features for Image Classification”, IEEE Transactions on Systems, Man and Cybernetics, Vol.3, pp. 610-621, November 1973.

[25] David A. Clausi, “An analysis of co-occurrence texture statistics as a function of gray level quantization”, Can. J. Remote Sensing, Vol. 28, No. 1, pp. 45-62, 2002

[26] J. Smith and C. Li, “Image Classification and Querying Using Composite Region Templates,” Int. J. Computer Vision and Image Understanding, vol. 75, no. 2, pp. 165-174, July 1999.

[27] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik, “Blobworld: A System for Region-Based Image Indexing and Retrieval,” Proc. Visual Information Systems, pp. 509-516, June 1999.

[28] R. Zhang, and Z. Zhang, “A Clustering Based Approach to Efficient Image Retrieval,” Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’02), Washington, DC, Nov. 2002, pp. 339-346.

List of Figures

List of Tables