RETRIEVAL OF IMAGE DATABASES

A THESIS

Submitted in fulfilment of the
requirements for the award of the degree
of
DOCTOR OF PHILOSOPHY
in
ELECTRONICS AND COMPUTER ENGINEERING

By

SHASHIKALA TAPASWI

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE–247 667 (INDIA)

SEPTEMBER, 2002

Candidate's Declaration

I hereby certify that the work, which is being presented in the thesis, entitled "Retrieval of Image Databases" in fulfillment of the requirement for the award of the Degree of Doctor of Philosophy and submitted in the Department of Electronics and Computer Engineering of the Institute is an authentic record of my own work carried out during a period from July-1999 to September-2002 under the supervision of Dr.R.C.Joshi.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute/university.

Tapaswi 10/9/2002

Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 10 · 9 · 2002

10/9/02

Signature of the Research Supervisor

The Ph.D. Viva-Voce examination of Shashikala Tapaswi, Research Scholar, has been held on 13/11/02.

Signature of Research Supervisor	13/11/2002 Signature of H.O.D	Signature of External Examiner

Abstract

With the spurt in computing and communication technology, more and more images are being captured, stored and are widely used in multimedia collections such as in medical imaging, geographic information systems, entertainment, education etc. Image Retrieval Systems are required to utilize such collections of images efficiently. The work presented in the thesis is an effort to propose different mechanisms and techniques for efficient retrieval of image databases using low level image features such as texture, color and shape.

In the first part, texture feature extraction method based on Haar Wavelet Transform is presented. The image is divided into sub-images and clustering is performed to cluster sub-images with similar characteristics. As texture feature extraction is very difficult, it is shown in research that neural network can be a useful tool for feature extraction. Another technique, which employs multi-layer perceptron based neural network for texture feature extraction and classification is proposed. The proposed schemes are able to extract texture features and the results obtained on images from Brodatz texture album are found to be reasonable and acceptable. Color is an important image feature, which is used in most of the image retrieval techniques. As user normally submit query in natural language and is not aware of low level image features. It is very difficult for a user to associate numeric values with the image features. So a technique for color based image

retrieval using fuzzy logic for defining the "imprecision" is proposed next. The algorithm has been tested on colored images from Kodak album.

For accuracy in retrieval, we need to extract multiple features for querying the database. A number of studies have been carried out which combine the various features for efficient and effective querying by image features. A technique, which computes the integrated feature vector, which combines the texture and color features, is presented in the next part and is able to extract regions, based on texture and color. Next, we have attempted to combine the color feature with the shape feature. The technique indexes the images on the basis of dominant color and then the local shape feature turning angle is used to perform shape based retrieval. This scheme automatically filters out the images on the basis of dominant color, which helps in fast retrieval of images. We compare the performance of the proposed technique with that of the Fourier descriptor based method, and Grid based method, and found that the precision Vs recall is improved.

Next, we address the issue of indexing the image databases. The variable bin allocation method can store the color information in compact form as compared to global color histogram. The method presented employs a parallel approach for the frame slice signature file using the variable bin allocation technique for representation of color image to speed up the retrieval of image databases. We have compared the performance of the proposed technique with that of the S-Tree parallel traversal implementation for color based image retrieval. The proposed approach has improved speedup performance.

We have carried out a case study on the real MRI images obtained from PGI, Chandigarh, biomedical images from National Technical Information Services Springfield U.S.A., and some medical images downloaded from the Internet on which the technique, which computes the integrated feature vector for texture and color, is implemented. This technique can be useful to the medical community for better diagnostics of abnormalities present in medical images based on texture and color.

Acknowledgments

I wish to express my whole hearted gratitude for the inspiration, encouragement and expert guidance that **Dr.R.C.Joshi** have given me throughout my stay at the Institute. I thank God Almighty who has given me such a guide on who's account this thesis has come to this shape. I consider myself fortunate for having been associated with him.

The co-operation and help extended by the Head and faculty members, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is gratefully acknowledged.

To follow the right path and to be able to concentrate, I cannot forget my loving mother **Late Smt. Pramila Tapaswi** who encouraged me like a spiritual teacher. My conscious throbs with spiritual sense that guided me all through the course of time.

I wish to convey my appreciation to my colleagues, and friends who have helped me in my difficult situations.

The acknowledgement would be incomplete without a mention of gratitude to **Amit Kush** for the help rendered to me during my research work.

My sincere thanks to **Mrs.Usha Joshi**, **Ira** and **Bakul** for the love and affection given to me through out my study period.

Last but not the least, with due humility and respect, I wish to express my sincere thanks to my **father** and **sister** for their love, motivation and constant encouragement, without which smooth sailing of the academic endeavor would not have been possible.

Shashikala Tapaswi

List of Abbreviations

ANN	Artificial Neural Network
CBA	Constant Bin Allocation
CBIR	Content Based Image Retrieval
CCV	Color Coherence Vector
CNS	Color Naming System
CPU	Central Processing Unit
CT	Computerized Tomography
DBMS	Database Management System
FD	Fourier Descriptor
FSSF	Frame Slice Signature File
GCH	Global Color Histogram
GUI	Graphical User Interface
HSV	Hue Saturation Value
IR	Image Retrieval
ISEF	Infinite Symmetric Exponential Filter
MBR	Minimum Bounding Rectangle
MINDIST	Minimum Distance
MINMAXDIST	Minimum of Maximum Possible Distance
MLP	Multi Layer Perceptron

MRI	Magnetic Resonance Imaging
NN	Neural Network
QBIC	Query By Image Content
RGB	Red Green Blue
ROI	Region Of Interest
SAM	Spatial Access Method
SEC	Smallest Enclosing Rectangle
VBA	Variable Bin Allocation
WWW	World Wide Web

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Statement of the Problem

1.1 Introduction

Information is inherently multimodal. A user can efficiently and effectively process information simultaneously in multiple dimensions. With the explosion of the web page development and availability of color scanners, digital media and cheaper storage devices, people now have access to large image databases of thousands of images. Large image databases are encountered in a wide range of applications such as satellite images, diagnostic medical images, geographical maps, collection of logos and portraits etc. This is mainly due to the technological developments (such as powerful processors, high-speed networking, high-capacity storage devices, improvements in compression algorithms, and advances in processing of audio, speech, image, and video signals) which allow us to digitize, store, and transmit audiovisual content in a very cost effective and efficient manner. Multimedia systems are now not only economically feasible, but also highly desirable [33][126].

A number of commercial organizations have large image and video collections of images that are being digitized for convenient on-line access. These digital databases are not a dream of the future, but have become a reality. Organizing these digital libraries into a small number of categories and providing effective indexing is imperative for accessing, browsing, and retrieving useful data in "real-time". Image Retrieval has become an active research area in computer vision recently. The growing prevalence of digital images and video increases the need for effective and efficient searching techniques. Due to huge amount of potentially interesting documents available over the Internet, searching for relevant information has become very difficult. For example, it is currently not possible to automatically search for a particular picture. Current solutions for searching this enormous amount of data primarily deal with textual information. However, identifying "keywords" for audiovisual content is a difficult problem as no generally recognized description of this material exists. Numerous applications such as stock video, satellite imaging, medical imaging, education and distance learning have identified the need of a solution to the problem of efficiently searching for various types of multimedia material of interest to the user.

A person can look over a few hundred thumbnail images to find a specific image, it is much harder to do the same with several thousand images. Exhaustive searching is not feasible when the database becomes sufficiently large. The traditional approach is to represent image contents by manual annotation. The annotated keywords are used to index images for retrieval. The annotation procedure relies on some human intervention and interpretation. This method of annotating each image is not feasible in a collection of millions of images in an image database. Two users may provide different annotations to describe the same image. Some visual aspects of the image may be inherently difficult to describe in terms of keywords. It is also difficult to capture all the visual aspects depicted in the image.

An alternative approach is to extract image contents automatically from the images [4] [31][32]. Research in image processing and computer vision has enabled us to capture salient aspects and identify objects in an image to some extent [13][72]. The present approach is to use features to capture the image contents. This technique has been used in many Content Based Image Retrieval (CBIR) systems developed in recent times such as Query By Image Content (QBIC) [32], NETRA [70], Photobook [85], VisualSEEK [112]. These CBIR systems enable users to pose queries by providing visual information using Graphical User Interface (GUI). A user can query by selecting example images, constructing sketches and drawings, selecting color and texture patterns, etc.

In the case of image databases, a feature vector that describes various visual cues, such as shape, texture, color or spatial information is computed for each image in the database. Given a query image, its feature vector is calculated and those images, which are most similar to this query image based on an appropriate distance measure in the feature space, are retrieved. Most image retrieval systems use shape, texture, color and spatial constraints to represent an image and retrieval is based on the similarity of features derived from these cues. Although, color seems to be highly reliable attribute for image retrieval, situations where color information is not present in the images require the use of shape and / or texture attributes for image retrieval.

For improving the retrieval accuracy, still there is a need to improve on the feature representation methods. Moreover, retrieval based on single image attribute might lack sufficient discriminatory information warranting a need for the use of multiple low-level image attributes. For more efficient retrieval the image attributes can be integrated together.

Thus, it is evident from the above discussions that the image retrieval techniques implemented for large image databases should be efficient in terms of response time, image query results should be up to the user's satisfaction, all the features of the image such as color, shape, texture and spatial information should be considered. We can therefore say that though there have been efforts in providing different image retrieval techniques, still there is a scope to propose new techniques with a view to improve the performance of image retrieval.

1.2 Statement of the Problem

In the present work we have attempted to investigate and propose techniques for efficient retrieval of image databases. The main objective of the present research work can be described by the statement of the problem expressed as follows "To investigate and propose schemes to extract low level image features and employ techniques on these features, which will lead to an efficient and fast image retrieval". We have several issues to contend within retrieving images from very large image databases, which are described and subdivided into small number of objectives. These objectives are specified as follows:

1. To investigate the techniques needed for Retrieval of Image Databases catering to user satisfaction.

2. To improve techniques for representing High level concepts and Low level visual image features.
3. To explore and propose new techniques for increasing the retrieval speed of image database.
4. To study and propose new techniques for representation of image content within the range of human perception.
5. To examine and propose techniques by integrating different image features for better retrieval results.
6. To investigate key application areas and validate the techniques experimentally on various image databases.

The efforts in meeting the above goals resulted in development of the following :

- Feature extraction methods.
- Region extraction methods.
- Integrated Texture feature and Color, Color and Shape feature representation.
- Representation of user image query in natural language using fuzzy logic.
- Training the neural network using backpropagation algorithm and incorporating incremental learning.
- Parallel Frame Sliced Signature files approach for indexing and fast retrieval of image databases.

1.3 Organization of the Thesis

The outline of the thesis is as follows: A Literature review in the related area and general considerations for retrieval of image databases is given in Chapter 2. Briefly the tools and techniques needed to extract image features and processing these image features for image retrieval has been discussed. Chapter 3 contains the texture feature extraction methods and proposes technique for region extraction using sub-images and clustering technique, also a neural network based technique for extraction and classification of texture feature is presented. In Chapter 4, the color feature of Image is considered and techniques are proposed for retrieval of images based on color which lies within the human perception, a neuro-fuzzy approach has been proposed, which accepts the user query in terms of natural language and the results of user query is presented as images having similar colors after computing the similarity factor. In Chapter 5 a technique which computes the feature vector integrating both texture and color features of image is presented. Region extraction using these integrated color texture feature vectors can be performed. Also a method which integrates dominant colors, which lies within the range of human perception, with the local shape feature turning angle approach is presented. Chapter 6 discusses the indexing method for speeding up the retrieval performance of the Image Databases. The frame sliced signature file technique employing parallel approach is presented. In Chapter 7 a case study of bio-medical Magnetic Resonance Imaging (MRI) images on which, the algorithm presented in Chapter 5 has been tested and the experimental results are reported. Finally, in Chapter 8 the contribution of the thesis, conclusions and scope for future work are presented.

Chapter 2

Review and General Considerations

2.1 Introduction

In recent years much emphasis has been placed on developing systems which make effective use of multi-dimensional data. This data may take the form of images, graphics, video sequences, satellite images etc. To manage multidimensional data in a database, various techniques were proposed like K-D tree, K-D-B tree [93], Quad tree [117], R-tree [37] etc. The complex nature of two dimensional image data has presented problems for traditional information systems designed strictly for alphanumeric data.

The objective of Image Retrieval Systems is to enable efficient and effective retrieval of relevant images from the database [5][10][18][36][46][98]. Image Retrieval Systems use features to retrieve relevant images [45][94][114]. Visual features describe the visual contents in the images such as color, texture and shape [64][78][80][84][109][110][120]. Non-Visual features describe the information associated with images such as date, place and time. The user viewing the images cannot easily identify these features, nor they can be extracted from the images automatically.

However, visual features can be seen and extracted from the images. Image retrieval systems are mainly classified into four categories based on user queries, and describe their characteristics.

2.2 Classification of Image Retrieval Systems

2.2.1 Annotation based Image Retrieval Systems

Image Retrieval Systems that use only non-derivable or non-visual attributes such as textual description are placed in this class. Many commercial imaging systems typically use relational database technology with enhancements for image data types. In these systems, the features stored in the database are date, time, image format and free text annotations. The user provides this information while loading images as associated keywords or annotations, which are then stored in the database. These features can be indexed using traditional indexing methods. The user can submit queries by typing a textual description. Images are then retrieved after searching in the text database for the matching keywords. It does not involve processing images, and deriving information from them. Such methods present major challenges for the retrieval system.

- The method is quite tedious for large image databases. Annotating each image in a collection of million of images may consume a good deal of time.
- The other problem is the user's perceptions. Two users may describe the same image very differently. They may use different words, emphasize different aspects of the image and describe at different levels of detail.

Research has been carried out in combining text based approaches with knowledge bases to facilitate abstract query composition. This allows the user to submit queries based not only on the keywords, but also on the system provided knowledge.

2.2.2 Image Feature based Image Retrieval Systems

Retrieval based on image features is an alternative to the annotation based approach. CBIR systems that use primitive features such as color, texture and shape are placed in this class. Most of the recently developed CBIR systems belong to this class. Features are extracted from images and stored in the database. Many academic prototype and commercial systems have been developed based on primitive features, among them are QBIC [3][32], Virage [36], NETRA [70] and VisualSEEK [112]. The advantages of such systems over non-visual systems are automatic derivation of information, use of perceptual properties, and scalability to very large databases. The systems in this class are based on similarity. The system measures the similarity between the query and database features using some distance functions. The result of a query is a ranked list of images based on similarity. The user can submit queries using visual languages where he/she can specify feature values (picking a color palette) or by providing example images. The kinds of queries that are supported by this type of system are "find all images containing red and blue regions", "find images similar in shape to this sketch", etc. The main focus on this class of CBIR systems is use of features to retrieve images from the database. The most primitive and widely used features are color, texture and shape.

Histograms are commonly used for color feature representation. Texture, that contains information about the structural arrangement of surfaces and their relationship to the surrounding environment is another important feature. Shape representations are useful in application areas such as logo collections and machine tools. Multidimensional indexing structures such as R-tree [37], R*-Tree [6], S-Tree [29], TV-Tree [67] are used to index such feature vectors.

2.2.3 Semantic based Image Retrieval Systems

CBIR systems that use logical features for retrieval are categorized in this class. It is also referred to as retrieval by semantic content. The systems in this class can answer queries of the form “find images that have the sunset”, “find images that have flowers” etc. There has not been much progress in this class of systems. Retrieval not only involves attributes that can be derived from the images, but also abstract concepts such as “sunset”, “mountain” and “flower” (extracting such concepts is beyond the capacity of current object recognition algorithms). A problem of determining whether an image represents a flower, for example, is a difficult one.

Image feature based CBIR systems focus on retrieving “stuff” (such as color, texture and shape), whereas semantics based systems focus on retrieving “things” (such as “sunset”, “mountains” and “flowers”). The main problem is to bridge the gap between these two. PIQ image DBMS where one can incorporate the semantic information using a data model is presented in [108]. Similar approaches have been used in [41].

SEMOG database management system, which supports semantics and image feature based approaches to support semantic level queries has been described in [66]. It supports modeling semantic information while loading images using user interaction.

2.2.4 Abstract Feature based Image Retrieval Systems

The ideal CBIR systems are placed in this class. Such CBIR systems should use abstract features, including reasoning about the meaning of the objects depicted or describing the events of the image objects in the image. Queries such as, "find pictures of Bill Clinton playing", and "find pictures of Bill Clinton with Tony Blair" are supported by such systems. Processing such queries not only involve identification of real life objects, but also events and interaction between them. The present research has not succeeded in parroting human perception in a general context. Identifying real life objects such as "Bill Clinton" and "Tony Blair" or events such as "playing" and "walking" cannot be achieved with the present technology. In order to process such queries, the system has to derive the meanings of events that are depicted in the image.

CBIR systems have attracted attention of other research communities such as traditional Information Retrieval (IR), signal processing and artificial intelligence. Much of the recent research is focused on improving the retrieval effectiveness and efficiency of the first two classes of CBIR systems. There is not much research on CBIR systems based on semantics, and none on abstract features. A summary of the various CBIR systems is presented in Table 2.1.

Type of CBIR system	Feature Used	Example Systems
Annotation based	Non-derivable attributes	Kodak Picture Exchange, PressLink Library
Image Feature based	Image features such as color, texture and shape	QBIC, Virage, VisualSEEK, NETRA etc.
Semantics based	Logical features such as "sunset" and "mountains"	SEMOG, CHABOT etc.
Abstract Feature based	Abstract features such as "playing" etc.	None

Table 2.1 Classification of Image Retrieval Systems

2.3 Basics of Image Retrieval Systems

2.3.1 Image Features

Color, texture and shape are the dominant image features used. Ideally, the features should be able to capture the full semantics portrayed in the images [28]. Achieving such features is difficult, given the current state of the art in image processing and computer vision research. Thus, the trend of using features to represent the semantic contents in the images is likely to continue in future. In this section color, texture and shape features are discussed to give an idea of the nature of features.

Color

Color has been recognized as the most discriminating feature in identifying the semantically relevant images from the database. Each CBIR system has used different color feature representation and showed the merits of using it. Color histograms, prominent colors and salient colors are examples of color feature representation [22][34][110]. The most common representation of color feature is the color histogram. Some of the advantages of using color histograms are their robustness with respect to scaling, orientation, perspective and occlusion. Various indexing schemes have been proposed for color representation [24][42][116][118]. Digitized images are normally represented as intensity values in RGB (Red, Green, Blue) color space [75]. Each color is a point in a three dimensional space. In a digitized color image, the color information of each pixel is encoded by 24 bits (8 bits for each color component), giving 16 Million possible colors. This can be reduced to a reasonable size by quantizing the color space. This coarsely quantized color histogram is then used to retrieve images using a suitable similarity measure. This technique has been used in the existing CBIR systems. The RGB color space does not reflect the colors that are actually perceived by humans. Many other color spaces, which are suitable for human perception, such as Munsell and HSV (Hue, Saturation, and Value) have been developed and used in CBIR systems [87]. Since the retrieval in CBIR systems is based on similarity measure, the distance function (or similarity measure) used in the color space also plays an important role in the performance of the systems. The retrieval performance of color histograms depends on the following factors: the color space in which the image colors are represented, the type of color quantization scheme used to reduce the color resolution, and the similarity metric used to compare color histograms. The IBM's QBIC [3][32] uses the *k* element color histogram, where *k* (typically, $k = 64$) is chosen by user. The *k* color space is first determined by transforming each color to the Munsell color space, and clustering them into *k* groups using clustering algorithms. The RGB color space is then quantized into *16* levels, giving a color space of *4096* cells. Each color cell is then transformed into one of the *k* elements (determined earlier) using its center of Munsell coordinates. The color histograms of the images (or objects within the images) are a normalized count of the number of pixels that fall in each of *k* colors. The retrieval is then done by computing the distances between the two color histograms $X = [x_1, \ldots, x_k]$ and $Y = [y_1, \ldots, y_k]$ as follows:

$$d(X,Y) = (X-Y)^t A(X-Y) = \sum_{i}^{k} \sum_{j}^{k} a_{ij} (x_i - y_i)(x_j - y_j) \tag{2.1}$$

where the superscript *t* indicates matrix transposition. Each a_{ij} of matrix *A* describes the similarity between color *i* and color *j*. The superiority of this distance function over Euclidean distance function is due to this matrix *A*, whose elements account for "cross-talk" between the colors. For example, the user perception of an orange image being similar to a red image is represented in A by making the values of the corresponding a_{ij} values higher. These techniques are not efficient for representing local color contents [13]. Here, the term local refers to the homogeneous regions of features extracted from the images as image objects. It has been observed that the color content within the homogeneous regions is much more sparsely distributed in the color space than in the whole image. Fewer colors can thus be enough to represent regions without affecting the perceptual quality. Motivated by this fact, the prototype CBIR system NETRA uses a *256*-color codebook to represent color features [70]. The color codebook is context sensitive. A different color codebook may exist for different applications. The codebook is constructed from training images using the Generalized Lloyd algorithm. NETRA uses this color codebook to represent the color feature for homogeneous regions extracted from the images. Experimental observation has shown that ten to fifteen colors from the color codebook are enough to represent the regions. This representation scheme has several advantages over the color histogram. First, the color feature is very compact, which reduces the amount of feature data for storage and indexing. Second, this method considers the properties of behavior about human perception. Human eyes cannot distinguish similar colors very well. The segmented regions contain only few colors that can be distinguished by human eyes. This method ignores the colors that are beyond the human perception by extracting the most dominant and distinctive colors from the regions. For each region, the color feature is then defined as:

$$f_c = \left\{ \left(I_j, P_j\right) \middle| I_j \in \{1,2,\ldots,256\}, 0 \le P_j \le 1 \sum_{1 \le j \le N} P_j = 1, and\, 1 \le j \le N \right\} \tag{2.2}$$

where I_j is the index into the color codebook, P_j is the corresponding percentage of colors present in the image, and N is the total number of colors in the region. The distance between the two regions say $A(\{(I_a, P_a) \mid 1 \le a \le N_a\})$ and $B(\{(I_b, P_b) \mid 1 \le b \le N_b\})$, where N_a and N_b denote the number of color features in the corresponding regions, is calculated as follows.

1. The distance between any two colors from the color codebook *($W(I_a, I_b)$)* is computed using the Euclidean distance as:

$$W(I_a, I_b) = \sqrt{(rI_a - rI_b)^2 + (gI_a - gI_b)^2 + (bI_a - bI_b)^2} \tag{2.3}$$

where rI_a, bI_a and gI_a are the red, blue and green components of the color indexed by I_a in the color codebook.

2. The best matched color l from the region B that has a minimum distance to the color I_a is first calculated as:

$$l = \arg\min_{1 \le b \le N_b} W(I_a, I_b) \tag{2.4}$$

where *argmin* returns the value of b having minimum $W(I_a, I_b)$.

3. The distance of a color element (I_a, P_a) in region A with color elements in region B $D[(I_a, P_a), B]$ is given by:

$$D[(I_a, P_a), B] = |P_a - P_l| . W(I_a, I_l). \tag{2.5}$$

$D[(I_b, P_b), A]$ can be computed in a similar way.

4. The distance between regions A and B, $d(A,B)$, is then given by:

$$d(A,B) = \sum_{1 \le a \le N_a} D[(I_a, P_a), B] + \sum_{1 \le b \le N_b} D[(I_b, P_b), A]. \tag{2.6}$$

The prototype CBIR system VisualSEEK also uses a similar approach to represent images or regions or objects in the images by color feature [109][111]. It uses the HSV color space to better represent the perceptual colors. The HSV color space is quantized to produce a compact set of colors. In the HSV color space, hue (H) represents the tint or tone, saturation (S) the amount of color and the value (V) blackness or whiteness. The hue is quantized into *18* levels to provide a fine representation, as it is the most significant characteristic of the color. Both saturation and value are quantized at *3* levels. This gives total *166* different colors in the set, represented as a *166* dimensional binary set. The color histogram for the image is then generated as follows:

$$h[m]=\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1} 1 \; if \;\; Q(T(I[x,y]))=m; \qquad (2.7)$$
$$0 \; otherwise$$

where h is the color histogram, $I[x, y]$ is the color image of width X and height Y, T is the transformation and Q is the quantization. The color histogram also represents the index to the *166* dimensional binary space. The similarities between given images are then calculated using Euclidean distance of points in binary space.

In order to provide a textual query language to retrieve images from a very large collection, a different technique to represent colors in images is used in [63][73]. Thirteen colors are chosen as prominent colors in images of natural scene: red, orange, yellow, green, blue-green, light blue, blue, purple, pink, brown, white, gray and black. These were chosen based on experimental results, which showed that they fall in the human perceptual range and identify the interesting objects from the images. The color image is first transformed into the HSV color space. The HSV color channels are then mapped into the *13* perceptual color channels. The image is then sent to a set of filters. The percentages of colors present in the image are stored for each image in the database. The image feature is then stored as a text string and retrieval is done using a text matching technique as in text retrieval systems [103].

We just describe the various ways of quantizing color space and representing color information of an image in the histograms. A histogram is a coarse characterization of an image. Images with different contents can have the same (or similar) histograms. Pass and Zabih proposed a technique called histogram refinement [81]. Histogram refinement splits the pixels in the image into several buckets based on the local property. Within each bucket, only pixels in the same class are compared. An experiment based on Color Coherence Vector (CCV), which partition each histogram bucket based on spatial coherence, was conducted and showed that the split histogram generated from CCV performs better [82]. Applying constraints such as "center pixel" for positional refinement does the split of pixels [81]. An improved version of CCV called color correlograms is also used and the experimental results are reported in [42][43]. This method has considered the spatial distribution of colors in the images. As we know, color is one of the main characteristics of natural images. To support retrieval of relevant images from the database, there is a need for a semantically rich representation of the color feature. To achieve this goal, many advanced forms of color representations are being developed and used in current CBIR systems. Though it is not possible to encode the large range of semantics depicted in the images within color representation, a certain kind of improvement in color feature representation has been observed in the recent times. Encoding color distribution within the images in feature representations is addressed by researcher in [82][124]. There is a need for encoding spatial information of colors within color representation [42][43][109]. The development of color feature representations that are compact, simple to index and semantically rich is a need for the next generation CBIR systems.

Texture

Texture has been described in many ways in the literature, and there is no formal definition for it. It can be thought of as repeated patterns that appear in the images in a structured manner. It has been widely used in computer vision and image processing research, mainly in segmentation and classification [30][113]. Since texture captures the human perception of the image, it has been used as one of the important cues in CBIR systems [3][109][113]. Many texture feature representations have been proposed and used in CBIR systems [16][70][71][79][99][109]. Wavelet decomposition [99][109][115] and Gabor decomposition [72] has been used in CBIR systems. In most cases texture regions are characterized by the responses of the filters to an image. Texture representations are derived from the gray scale representation of images, and have largely ignored the importance of the color intensity. There have been attempts to combine color and texture [109]. Serge and Malik [8] have combined intensity with texture feature representation.

Shape

Shape features describe the shape of the objects in an image. Applications of shape features are searching catalogs of consumer goods such as hand-tools, searching the database of symbols and logos, searching inventories of mechanical parts, etc. There are usually a small number of dominant objects in natural images. Shape features can be extracted from such objects and stored into the database. The images then can be retrieved comparing shape features based on the similarity. In some applications such as logo databases and aircraft databases, the system would provide a better retrieval performance using the shape features. Many shape features have been developed and used in CBIR systems [2][99][100]. The Shape Photobook models the physical "interconnectedness" of the shape [85]. The idea is to capture the deformation of the real objects quantitatively. The stiffness matrix is first generated using the finite element method. The stiffness matrix describes how each point in the object is connected to every other point. The eigenvectors of the stiffness matrix are then calculated and used to encode deformation related to some base or average shape. The retrieval is done by simply comparing the amplitudes of the eigenvectors. The Smallest Enclosing Circle (SEC) to describe the shape of an object in the image is also used. A signature is derived for an object from the SEC, which is invariant of translation, rotation and scale. For an object in the image, the SEC is first determined. The center of the SEC is used as a reference point. Since the SEC of a shape is unique, it is translation invariant. Rotation invariance is obtained by using a vector from the SEC center to the centroid of the object. Scale invariance is obtained by dividing the SEC by n-radial lines with equal angles, starting from the orientation vector (vector from the center of the circle to the centroid of the object). The distances from the center of the SEC to the boundary of the objects are calculated. The ratios of these distances to the SEC radius are scaling invariant. Then an n dimensional shape vector (or signature) is obtained by taking the n ratios starting from the orientation vector in counter-clockwise order. Contour representation is another form of shape representation. The boundary pixels of the images are used to derive the contour representation. NETRA uses three types of contour representations: curvature function, centroid distance and complex coordinate function [70]. The other shape descriptors used by CBIR systems and found in the literature of pattern recognition are Fourier descriptors [101].

2.4 Feature Indexing

A simple way of processing user queries is to evaluate the similarity with each of the images stored in the database. This is not viable for very large databases. Indexes enable efficient access to a subset of the database (rather than all of it). Indexing in the context of relational databases is a well researched area and many indexing structures such as B-trees and B+-trees have been proposed and used in commercial product [90]. Insertion, deletion, and search algorithms for such indexing structures are based on keys, which are usually numbers or are (1-dimensional) strings. High dimensional feature vectors represent image features such as color, texture and shape. Traditional indexing structures are not suitable for high dimensional vectors. Many novel indexing structures have been proposed such as Quad trees, and R-trees [37]. Among the indexing structures, the most relevant to CBIR systems are R-trees [37] and their variants such as R*-tree [6], R+-trees [106]. A 2D string based representation for a picture is presented in [16][17]. This approach allows an efficient and natural way to construct iconic indexes for pictures.

2.4.1 Index Structures

The role played by keys in organizing B-tree is carried out by n-dimensional rectangles in R-trees, called Minimum Bounding Rectangles (MBRs). MBRs are arranged in a hierarchical manner. The actual data (or the pointer to the actual data) is stored at the leaf nodes together with its minimum enclosing rectangle. The MBR at the parent nodes encloses the smaller rectangles of its children. That is, the higher the level of the node, the greater the size of the MBR. A non-leaf node contains entries of the form *(p,R),* where *p* is a pointer to its child node and *R* is the MBR of all rectangles which appear in the child node. A leaf node contains entries of the form *(oid, R)*, where *oid* refers to a record in a database, describing the object and *R* is the enclosing rectangle of that object. The R-tree index structure allows overlapping in the non-leaf node rectangles. This may give multiple search paths for a query point. Therefore, R-tree does not guarantee that only one search path is required for a search. Due to insertion and deletion of records, its search performance can gradually degrade. To overcome the deficiencies, a set of new index structures that belong to the same family such as Packed R-trees, R+-trees [106], and R-trees [37] have been proposed. The Packed R-tree is designed to overcome the deficiency of R-trees where the data to be stored are known beforehand. It exploits the nearest neighbor knowledge of known data to group them together. It is experimentally found that the Packed R-tree performs better than R-tree for two-dimensional data. The R-trees allow overlapping of bounding rectangles at the non-leaf nodes, which may yield multiple search paths. The R+-tree does not allow overlapping of rectangle [106]. The same record may be stored in different leaf nodes. This may introduce several paths from the root leading to the same record. The R+-trees outperform the R-trees when there are just a few large rectangles and quite a few small ones. These are designed for spatial databases, and thus named SAMs (Spatial Access Methods). As SAMs are not designed for multimedia databases, their use has been limited by various underlying assumptions such as use of Euclidean distance to measure the similarity between objects. Many other similarity and metric-based access methods have been proposed to overcome the problems. Similarity indexing with the SS-tree has been proposed. The structure of SS-tree is similar to that of the R-tree, but the SS-tree uses spheres instead of rectangles. The SS-tree uses one base distance metric defined by a vector, which is equal in size to the feature vector. The SS-tree exploits the properties of the distance functions and the features (provided by the domain expert) at the time of creation. Thus, the index structure can be tuned for a particular domain and distance function. It therefore appears superior to R-trees in similarity indexing applications. SS+-tree [62] has been proposed to improve the performance of SS-tree. The SS+-tree uses a tighter bounding sphere compared to that of the SS-tree. The splitting algorithms used by most of the R-tree family access structure are based on topological information such as the minimization of perimeter, area and overlapping. The approach used in similarity based access mechanisms like the SS-tree is to collect the nearby vectors at the same node, i.e., the minimum value of variance at the node is the better. The SS+-tree uses the *k*-means clustering algorithm while splitting the nodes. As a consequence of this, one node with bigger variance might overlap with a smaller one. To overcome this problem, the SS+-tree uses an overlap reduction technique where the child nodes are created by applying *k*-means clustering to the grandchildren. Theoretically, the decomposition of space into spherical regions is optimal (in the sense that it minimizes the expected number of regions touched) for the nearest neighbor query in Euclidean space. However, this is not actually the case, since the volume occupied by the space is very high, resulting in the overlapping. Furthermore, due to the isotropic nature of the spheres, extending the sphere in one direction will extend it in all directions, whereas this is not the case in boxes. To overcome the problem of high overlapping due to the inherent nature of spheres, SS+-tree uses the smallest enclosing sphere calculated by a spatial search. As mentioned above, the main problem associated with the SS-tree is that it employs bounding spheres, which occupy much larger volumes compared to bounding rectangles. The SS-tree addresses this problem using minimum enclosing sphere [6]. An index structure called the SR-tree (Sphere / Rectangle-tree) which integrates bounding spheres and bounding rectangles is proposed in [58]. A region in the SR-tree is specified by the intersection of a bounding sphere and bounding rectangle. The introduction of a bounding rectangle reduces the regions into smaller sizes compared to those of SS-tree. Other high dimensional index structures proposed for spatial and image databases are the Hybrid tree [15], M-tree [21], HB-tree [61], X-tree [7], TV-tree [67], K-D-B tree [93], etc.

2.4.2 Index Search Algorithms

So far, we have discussed the index structures. Insertion, deletion and search are elementary algorithms used by query processor on the index structures. This section provides details of various search algorithms.

(a) Exact Match Search

This algorithm returns results from the database that match exactly with the user query. This method is used in Geographical Information Systems where the system returns all objects containing the given query point. For example, the user can pose a query "retrieve all cities at this altitude level", selecting an altitude from the map, the system then performs the exact match search and returns the result. However, such queries are irrelevant to CBIR systems, where the retrieval is based on the similarity.

(b) Nearest Neighbor Search

For the CBIR system, what is relevant is a similarity match, rather than the exact match. The processing of even an elementary query requires the nearest k images. This is formally defined as, K-nearest Neighbor Search: Given a collection $C=\{I_1,\ldots,I_n\}$ of n images and a feature vector q_f, $R=\{J_1,\ldots,J_n\}$ is an ordering of C such that the distance of q_f with the corresponding feature of J_i is less than the distance of q_f with the feature of J_{i+1}. The subset $R_k=\{J_1,\ldots,J_k\}$ is the set of k nearest neighbors.

The algorithms for the above searches have been studied well for various indexing structures. Roussopoulos et. al. proposed a k nearest neighbor algorithm for R-trees in [95]. It used a "branch and bound" R-tree traversal algorithm. The tree traversal is guided by two metrics:

MINDIST (minimum distance) and MINMAXDIST (minimum of maximum possible distance). The MINDIST is the minimum distance of a query point from the rectangle. If the point is inside the rectangle, then the MINDIST is zero. If the point is outside the rectangle, then the MINDIST is calculated by using the Euclidean distance. The MINMAXDIST is the minimum value of the maximum distances between the query point and the points on each of the n axes respectively. The Minimum Bounding Rectangles (MBRs) are then ordered based on the above two metrics. The MINDIST ordering is the optimistic choice, while the MINMAXDIST is pessimistic. It is experimentally found that this method scales up well with respect to both the numbers of nearest neighbor requested and with the size of the data sets. However, such *k*-nearest neighbor algorithms have two major drawbacks.

The first is that they are designed for Geographical Information Systems and do not scale up well for high dimensional image feature vectors. The second is that they are designed for a fixed distance function, i.e., the processing algorithm works well for a predefined distance function.

A fast nearest neighbor search algorithm for high dimensional space based Voronoi cell is proposed by Berchtold et. al. In this approach, the result of any nearest neighbor search, which corresponds to a computation of the Voronoi cell of each data point, is precomputed. Such computed Voronoi cells are then stored into the high dimensional index structure for efficient processing. As a result, the nearest neighbor search corresponds to a simple point query on the index structure. Thus, the approach can be used in very high dimensional space. One of the disadvantages of this technique is that one needs to precompute the result beforehand.

As mentioned above, the traditional approaches of *k*-nearest neighbor search are well suited for a medium-dimensional similarity distance function. But they do not meet the efficiency requirements of complex high dimensional and adaptable distance functions. To address this problem, an optimal multi-step *k*-nearest neighbor search algorithm is proposed. The multi-step algorithm has two steps, filtering and measuring similarity. In the filtering step, the filtering distance function is applied to the index structure and retrieves the images that need to be considered for the second stage. The main aim at the filtering stage is to generate a fewer number of candidates for the second stage. The second stage measures the similarity among images using the exact similarity measuring function. The optimal algorithm generates the exact number of images from the first stage.

(c) Range Search

The searching of relevant images using similarity is discussed above using a nearest neighbor search algorithm. In the nearest neighbor approach, the user gives a feature vector and retrieves the requested (say *k*) number of the nearest neighbor images. Instead of using this, a range search can be performed, where the user specifies a range of similarity values and retrieves images that fall within the range. For example, retrieve all images having similarity values in between *0.9* and *1.0* with the feature vector *V*. This approach looks promising if a large number of similar feature vectors reside in the same physical blocks. This is formally defined as, Range Query: Given a collection $C=\{I_1,\ldots,I_n\}$ of n images and a feature distance function D_f, find the images $R \subset C$ such that $D_f(q_f, I_{i_f}) \le t_f$, where q_f is the query feature vector, I_{i_f} is a feature vector of a database image I_i and t_f is a threshold of feature distance.

For image retrieval different mathematical models have been used and they are being reviewed in the following sections.

2.5 Mathematical Techniques for Image Processing

2.5.1 Wavelet Background

Recently developed theory of wavelets provides an extremely useful mathematical toolkit for hierarchically decomposing functions in ways that are both efficient and theoretically sound. A wavelet representation of a function consists of a coarse overall approximation together with detail coefficients that influence the function at various scales.

Storing the wavelet transform of the image, rather than the image itself, has a number of advantages. One advantage of the wavelet transform is that often a large number of detail coefficients turn out to be very small in magnitude.

Some of the properties of wavelets are mentioned below:

- Linear-time complexity: Transforming to and from a wavelet representation can generally be accomplished in linear time, allowing for very fast algorithms.
- Sparsity: For functions typically encountered in practice, many of the coefficients in a wavelet representation are either zero or negligibly small. This property offers the opportunity both to compress data and to accelerate the convergence of iterative solution techniques.
- Adaptability: Unlike Fourier techniques, wavelets are remarkably flexible in that they can be adapted to represent a wide variety of functions, including functions with discontinuities, functions defined on bounded domains, and functions defined on domains of arbitrary topological type. Consequently, wavelets are equally well suited to problems involving images, open or closed curves, and surfaces of just about any variety.

2.5.2 Gabor Functions and Wavelets

Gabor features have been used in several image analysis applications including texture classification and segmentation. A two dimensional Gabor function $g(x,y)$ and its Fourier transform $G(u,v)$ can be written as :

$$g(x,y)=\left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{2.8}$$

$$G(u,v)=\exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\} \tag{2.9}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$, Gabor functions form a complete but nonorthogonal basis set. Expanding a signal using this basis provides a localized frequency description. A class of self-similar functions, referred to as Gabor wavelets in the following discussion, is now considered. Let *g(x,y)* be the mother Gabor wavelet, then this self similar filter dictionary can be obtained by appropriate dilations and rotations of *g(x,y)* through the generating function :

$$\begin{aligned} g_{mn}(x,y) &= a^{-m}G(x',y'),\ a\rangle 1, m,n=\text{integer} \\ x' &= a^{-m}(x\cos\theta+y\sin\theta),\ \text{and } y'=a^{-m}(-x\sin\theta+y\cos\theta) \end{aligned} \tag{2.10}$$

where $\theta = n\pi/K$ and K is the total number of orientations. The scale factor a^{-m} in Equation 2.10 is meant to ensure that the energy is independent of m.

Given an image *I(x,y)*, its Gabor wavelet transform is then defined to be

$$W_{mn}(x,y)=\int I(x_1,y_1)g_{mn}{}^{*}(x-x_1,y-y_1)dx_1\,dy_1 \tag{2.11}$$

where * indicates the complex conjugate. It is assumed that the local texture regions are spatial, homogeneous, and the mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the transform coefficients are used to represent the region for classification and retrieval purposes:

$$\mu_{mn} = \iint|W_{mn}(xy)|dxdy,\ \text{and } \sigma_{mn}=\sqrt{\iint(|W_{mn}(x,y)|-\mu_{mn})^2\,dxdy} \tag{2.12}$$

A feature vector is constructed using μ_{mn} and σ_{mn} as feature components.

2.5.3 Neural Computing

Neural computing has proved to be a useful solution in many application areas that are difficult to tackle using conventional computing [60][96][119]. As neural networks provide rather general techniques for modeling and recognition, they have found applications in many diverse engineering fields. In early days of neural computing, the first applications were in pattern recognition, but since then neural computing has spread to many other fields of computing. Still, the relative impact of neural network techniques is perhaps largest in the area of pattern recognition. Artificial Neural Networks (ANNs) are biologically inspired, having a large number of biological neural cells in the brain. ANNs learn by previous experience and can make decisions. The ANN architecture is different to conventional architectures found in the computing. They possess the property of learning and decision making. They also learn from previous experience. Because of these properties, ANNs are used in different applications. An artificial neural network is a network of simple processing units that are interconnected through weighted connections [39]. The interconnection topology between the units and the weights of the connection define the operation of the network. We are generally interested in feedforward networks where a set of units are designed as the input units through which input features are fed to the network.

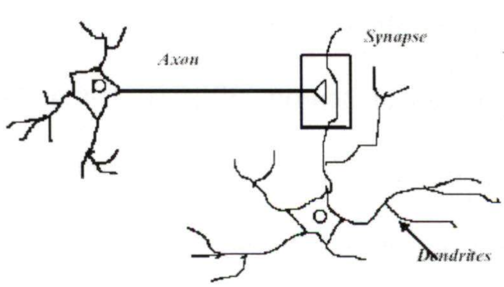


Figure 2.1 Biological Neuron

Biological Neural Model

Neural networks are made up of very simple and highly interconnected processors called neurons, which are analogous to the biological neural cells, or neurons, in the brain [57]. These neurons are connected by a large number of weighted links. Each neuron is capable of receiving, processing, and transmitting data. The biological neuron consists of mainly four parts, the cell body, synapses, dendrites and axon as shown in Figure 2.1. Inputs are received from other neurons by dendrites at connection points called synapses. On receiving these signals, the cell body sums the membrane potential provided by the dendrites. When the excitation in the cell body exceeds some threshold, the cell fires sending the signal down the axon to another neuron.

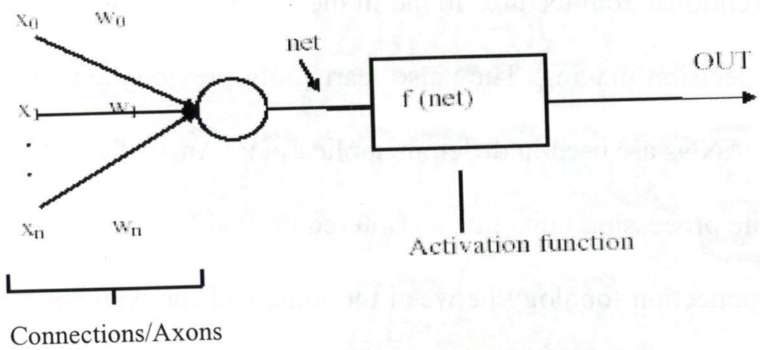


Figure 2.2 Artificial Neuron

The Artificial Neuron

In an artificial neuron, a set of inputs is applied, each representing an output of another neuron. Each input is multiplied by a corresponding weight, which is analogous to synaptic strengths in biological neuron [39][57]. The weighted inputs are summed to determine the net input of the neuron. This net input is processed further by using a squashing function, to produce the neuron's output signal. This function may be linear, nonlinear (step function) or sigmoidal (*S* shaped). In the case of a step function, the output value can either be *0* or *1*, depending on the sum of the weighted inputs, a value between *0* and *1* may be expected.

We may express this mathematically:

$$OUT = f\left(\sum_{i=0}^{n} w_i x_i\right) \tag{2.13}$$

OUT is neuron's output

f is the activation function

n is the number of inputs

x_i is the *ith* neuron's input

w_i is the weight of the *ith* neuron

The above simple model of the artificial neuron, given in Figure 2.2 ignores many characteristics of its biological counterpart, nevertheless the simple artificial neuron model is very useful in developing more complex models. The problem subsequently faced is choosing an ANN model best suited for a given task.

Artificial Neural Networks

The single artificial neuron has been defined, it is possible to describe an actual artificial neural network. An ANN is simply a set of interconnected neuron [92]. Each neuron can be treated as being a separate Central Processing Unit (CPU) running its own simple program. This simple program has the function of computing the weighted sum of the inputs from other neurons and giving the output as a single number. This output is then sent to other neurons, which are performing the same task. The idea of parallel processing also contributes to the robustness of the system. In a neural network, neurons are grouped into layers. There are three types of layers present in a neural network. These layers are input, hidden, and output layers. Each neuron in the input layer of the network receives a small piece of the input pattern. This input is fed to the neurons of the hidden layers. Each of the middle layer neurons thus receives the entire input pattern, but the pattern is modified by its passage through the weighted connections leading to the hidden layer. Since the weights on the connections are typically different for each neuron, each neuron sees a somewhat different version of the input pattern than its neighbors do. Normally all the hidden layer neurons transmit their output signals to all the neurons in the output layer. Thus, each of the output layer neurons receives the complete pattern of the output from the hidden layer neurons. This output layer presents the data back to the external environment. The number of layers present within a network can be counted in one of the two ways. Some researchers count all the layers, while others do not count the input layer. The reason why the input layer is not counted is because it plays no significant role in the computation of the result. Its role is only to feed the inputs into the network. The execution of a neural network can be simply explained. An input pattern about a given problem is fed into the input layer. The neurons in the input layer feed their output to the neurons in the next layer. The next layer's neurons (hidden layer), compute their outputs and feed them to the next layer. This interaction among the layers continues until some condition is satisfied, or when the neurons in the output layer get excited and produce an output to the external environment. Many neural network topologies and training algorithms exist.

Learning and Training of a Network

Neural networks learn to solve the problem, they are not programmed to do so. Thus learning and training are the fundamentals to all neural networks. Learning in neural networks consists of making systematic changes to their weights in order to improve their response to the acceptable levels. Training and learning are not the same. Training is the procedure by which the network learns, where as learning is the end result of that procedure. Training is a procedure external to the network, learning is the internal process or activity. There are mainly three types of training:

(a) Supervised Training

In this method, the network is provided with an input stimulus pattern along with the corresponding desired output pattern. It calculates the error between the desired and the actual output. This error is then used to modify the weights on the interconnections between the neurons.

(b) Graded Training

This training is also known as reinforcement training. It is similar to supervised training except that the desired output is not provided. Only the "grade" on how well the network is doing. There are some schemes for this kind of training and giving only the message "you succeeded" or "you failed".

(c) Unsupervised Training

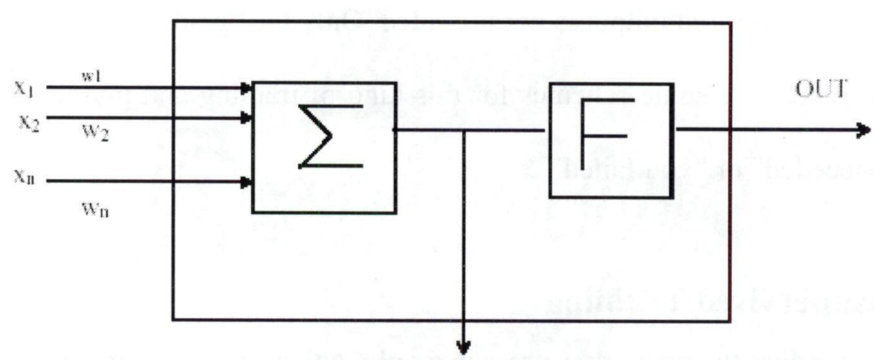
In this procedure the network is presented only with a series of input patterns and is given no information or feedback at all about its performance levels. This type of training procedure is mostly used by statistical modeling applications.

Backpropagation Neural Network

The Backpropagation algorithm can be used in conjunction with Artificial Neural Networks. The invention of the Backpropagation algorithm has played a large part in the resurgence of the interest in Artificial Neural Networks. Backpropagation is the systematic method for training multilayer artificial neural network. It has dramatically expanded the range of problems to which Artificial Neural Networks can be applied, and it has generated many successful demonstrations of its power.

Network Configurations

The neuron used as the fundamental building block for backpropagation networks is shown in Figure 2.3. A set of inputs is applied, either from outside or from the previous layer. Each of these is multiplied by a weight, and the products are summed. This summation of products is termed as net and must be calculated for each neuron in the network. After net is calculated, an activation function is applied to modify it thereby producing the signal OUT. A sigmoidal activation function is normally used.



$$Net = x_1 w_1 + x_2 w_2 + \ldots\ldots + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

$$OUT = f(Net)$$

Figure 2.3 Artificial Neuron with Backpropagation Algorithm

The Multilayer Network

The Figure 2.4 shows a multilayer network suitable for training with the backpropagation algorithm. The first set of neurons connecting to the inputs serves only as distribution

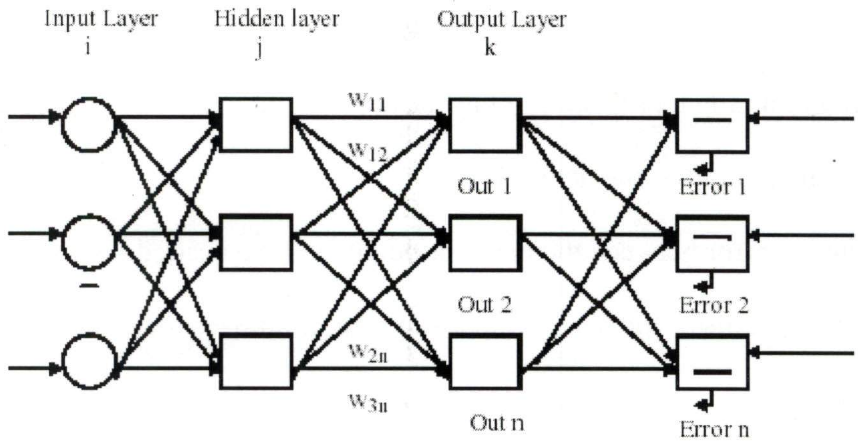


Figure 2.4 Two Layer Backpropagation Network

points, they perform no input summation. The input signal is simply passed through to the weights on their outputs. The network shown in the figure is considered as being two layers. Also a neuron is associated with the set of weights that connects to its input. Thus the weight in layer 1 terminate on the neurons of layer 1. The input or distribution layer is designated layer 0.

An Overview of Backpropagation Training

The objective of training the network is to adjust the weights so that the application of the set of inputs produces the desired set of outputs. For reasons of brevity, these inputs and outputs can be referred to as vectors. Training assumes that each input vector is paired with a target vector representing the desired output, together these are called a training pair. Before starting the training process, all the weights must be initialized to small random numbers. This ensures that the network is not saturated by large values for the weights, and prevents certain other training pathologies. For example, if the weights all start at equal values and the desired performance requires unequal values, the network will not learn.

Training the backpropagation network requires the steps that follow:

1. Select next training pair from training set, apply input vector to the network input.
2. Calculate the output of the network.
3. Calculate the error between the network output and the desired output.
4. Adjust the weights of the network in a way that minimizes the error.
5. Repeat steps 1 through 4 for each vector in the training set until the error for the entire set is acceptably low.

The operations required in the steps 1 and 2 above are similar to the way in which the trained network will be ultimately used, that is, an input vector is applied and the resulting output is calculated. Calculations are performed on a layer by layer basis. Referring to Figure 2.4, the first outputs of the neurons in layer *j* are calculated, these are then used as inputs to layer *k*, the layer *k* neuron's outputs are calculated and these constitute the network output vector.

In step 3, each of the network outputs labeled 'Out' in Figure 2.4 is subtracted from the corresponding component of the target vector to produce an error. This error is used in step 4 to adjust the weights of the network, where the polarity and the magnitude of the weight changes are determined by the training algorithm. After sufficient repetitions of the four steps, the error between the actual outputs and the target outputs should be reduced to an acceptable value, and the network is said to be trained. At this point, the network is used for recognition and the weights are not changed. The steps 1 and 2 constitute a "forward pass" in that the signal propagates from the network input to its output. Steps 3 and 4 constitute "reverse pass", here the calculated error signal propagates backward through the network where it is used to adjust the weights.

2.5.4 Fuzzy Theory

Fuzzy systems provide a mathematical framework for capturing uncertainty [9][57][104]. Image recognition is an area where fuzzy representation and fuzzy reasoning can be successfully applied, mainly for two reasons: (i) ambiguity in the images to be recognized, and (ii) the need for fast processing, that is, complicated formulas may not be applicable for a real-time recognition, in this case a fuzzy system may be more convenient.

Different approaches are possible depending on the image recognition tasks, two of them being (i) object recognition, that is recognizing shape, distance, and location of objects, and (ii) texture analysis, for example, an image X of size m x n pixels can be represented as a set of fuzzy sets and membership degrees to which pixels belong to the fuzzy concepts, such as "brightness", "darkness", "edgeness", "smoothness".

Fuzzy methods can be used at two levels of image recognition and image processing: (i) low-level image processing, tasks to be performed at this level being image segmentation, boundary detection, image enhancement, and clustering, and (ii) high-level image understanding, in which process ends up with a symbolic description of the image.

2.5.5 Neuro-Fuzzy Technique

As ANNs and fuzzy systems share common application areas, good examples are control systems and pattern recognition. Both fuzzy systems and neural networks are model-free systems. When they are combined we arrive at the area of fuzzy-neural (or Neuro-fuzzy systems). Hybridization of neural network and fuzzy logic are used for rule generation problem. The neuro-fuzzy computing is the fuzzy system augmented by neural network and is called Neuro-Fuzzy system.

The objective is to take advantage of the features of both. In implementing fuzzy systems, ANNs can play a role in:

- Computing membership functions
- Fuzzification of inputs
- Implementing membership functions
- Combining membership functions

2.6 Conclusions

In this chapter a review of retrieval of image databases is presented. The visual features, which are used as image features, are studied and described. The mathematical tools, which can be employed for improving the retrieval performance in terms of quality, retrieval speed, storage efficiency etc., have been reviewed and presented.

Chapter 3

Retrieval based on Texture Feature

3.1 Introduction

Texture has been one of the most important characteristics, which has been used to classify and recognize objects and scenes. In images texture has been recognized as an important aspect of human perception. Texture has been defined as the uniformity, density, coarseness, roughness, regularity, intensity and directionality of discrete tonal features and spatial relationships. It is an important characteristic for the analysis of various types of the images including natural scenes, remotely sensed data and biomedical modalities. Most texture features are based on structural, statistical, or spectral properties of the image. Any texture feature must include both the statistical and structural information present in the image to characterize it efficiently. Various techniques have been developed for texture segmentation, classification, and synthesis. Texture analysis makes a significant contribution in the area of retrieval of image databases. In literature various texture features including contrast, directionality and coarseness have been discussed in [1][13][38][71][72][97] and [102]. Contrast measures the vividness of the texture and is a function of gray-level distribution. Directionality measures the "peakedness" of the distribution of gradient directions in the image. Coarseness measures the scale of texture. These textual features of images contain main visual characteristics of images that can be utilized in image clustering and retrieval.

Some of the texture features have been depicted in Figure 3.1.

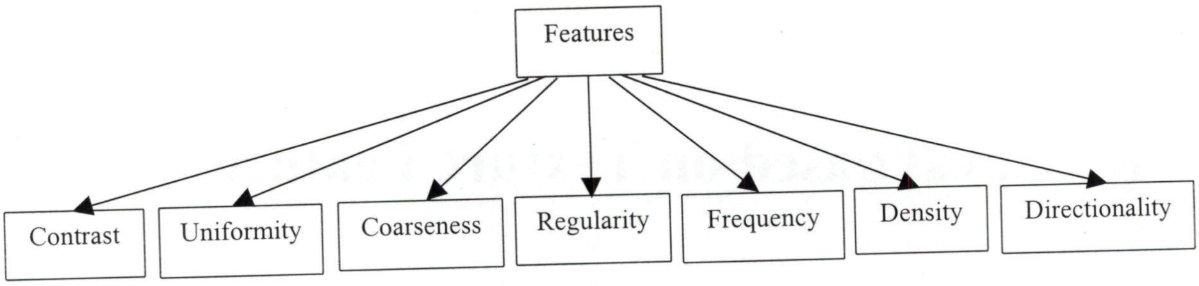


Figure 3.1 Texture Features

In QBIC [3][32] system features based on coarseness, contrast, and directionality were used. Gabor filter based multi-resolution representations has been used in [71][72] to extract texture information. In Photobook [85] project 2-D Wold based decomposition as texture descriptor has also been used. The main drawback of these approaches are that when large database of images are available, these techniques are either ineffective or require a lot of computation. More recent approaches as described in [13][70][107] employs a region based query systems, which involve image segmentation based on color and texture. Though there have been approaches that use post processing methods like relevance feedback to improve the retrieval accuracy, still these methods depend heavily on the feature representation methods. Thus it is crucial to improve the low-level feature extraction methods and algorithms to improve overall image retrieval. Attempts have been made to improve retrieval efficiency using easy to compute low level feature extraction algorithms. The texture features if done carefully, can provide significant information for scene interpretation and image retrieval and classification. Applying wavelet transform on the images can provide information about the contrast of images. Also, different sub-bands generated by wavelet transform have information about horizontal, vertical, and diagonal edges in images, which helps in extracting features related to directionality of images [115]. Neural Networks have also been used for extraction and classification of texture feature [92][125].

3.2 Proposed Schemes

Two schemes are proposed one based on Haar Wavelet transform, the other based on multilayer perceptron neural network approach. In the first approach Haar wavelet transform are used for feature extraction. Instead of storing a single feature vector for the whole image, each image is divided into sub-images of varying size, for each sub-image the signature i.e., the feature vector is computed and then on these feature vectors clustering is performed to group the sub-images with similar characteristics. The clustering can be performed using distance metric between pair of sub-images. Each cluster thus contains group of sub-images with similar characteristics. In the second approach texture features are extracted from the sub-images using a Neural Network with supervised learning, the network is trained, the input and the output texture image for the Neural Network is same. After the training is finished, the values of the hidden units are extracted and are considered as feature vector. Then these feature vectors are submitted as input to the multi-layer perceptron classifier, for classification of texture patterns into sub-classes. These schemes are described in detail in the subsequent sections.

3.2.1 Haar Wavelet Approach

Wavelets are mathematical tool used for hierarchically decomposing functions in ways that are both efficient and theoretically sound. A wavelet representation of a function consists of a coarse overall approximation together with detail coefficients that influence the function at various scales [49][115]. The wavelet transform has excellent energy compaction and de-correlation properties, which can be used to effectively generate compact representations that exploit the structure of data. By using wavelet subband decomposition, and storing only the most important subbands (that is, the top coefficients), we can compute fixed size low-dimensional feature vectors independent of resolution, image size and dithering effects. Also, wavelets are robust with respect to color intensity shifts, and can capture both texture and shape information efficiently. Furthermore, wavelet transforms can be computed in linear time, thus allowing for very fast algorithms.

Haar wavelets have been used to compute feature signatures because they are the fastest to compute and have been found to perform well in practice [44][54]. Haar wavelets enable us to speed up the wavelet computation phase for thousands of sub-images of varying size. They also facilitate the development of efficient incremental algorithms for computing wavelet transforms for larger images in terms of the ones for smaller sub-images.

One Dimensional Haar Wavelets

Suppose a one-dimensional pixel image I = [9, 7, 3, 5] is given. The Haar wavelet transform for the above image can be calculated as follows. We first average the values together pairwise to get a new lower resolution image [8, 4]. Clearly, some information has been lost in this averaging process. To be able to restore the original four values of the image, we need to store some detail coefficients that capture the missing information. In Haar wavelets, the difference of the (second of the) averaged values from the average itself constitutes the detail coefficients. Thus, for the first pair of averaged values, the detail coefficient is 1 since 8 - 7 = 1, while for the second we need to store -1 since 4 - 5 = -1. Note that it is possible to reconstruct the 4 pixels of the original image from two averages and the two detail coefficients. By recursively repeating the above process on the lower resolution image containing only the averages, we get the following full decomposition.

Resolution	Averages	Detail Coefficients
4	[9, 7, 3, 5]	
2	[8, 4]	[1, -1]
1	[6]	[2]

Table 3.1 Detail Coefficients

The wavelet transform of the original image is defined to be the single coefficient representing the overall average of the pixel values followed by the detail coefficients in the order of increasing resolution as shown in Table 3.1. Thus, the one-dimensional Haar wavelet transform for the original image is given by I' = [6, 2, 1, -1].

Each entry in I' is called a wavelet coefficient. Using the wavelet transform of an image, rather than the image itself has a number of advantages. One advantage is that a large number of detail coefficients tend to be very small values. Thus, truncating these small coefficients from the transform introduces only small errors in the reconstructed image, giving a form of "lossy" image compression. Intuitively, the wavelet coefficients in the above example carry different weights with respect to their importance for the reconstructed image. For example, the overall average of the whole data set is more important than any of the detail coefficients because it affects the whole range of reconstructed values. In order to equalize the importance of all coefficients, we need to normalize the final wavelet coefficients appropriately. A basis function $u(x)$ is normalized if $\langle u | u \rangle = 1$. The new normalized coefficients are obtained by dividing each old wavelet coefficients with superscript j by $\sqrt{2^j}$, where j denotes the index of the approximation level the coefficient appears in (where level 1 is the coarsest resolution level). Thus, the wavelet transform for the previous example becomes $I' = [6, 2, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$.

Two Dimensional Haar Wavelets

There are two ways in which wavelets can be used to transform the pixel values in a two-dimensional image. Each of these transforms is a two-dimensional generalization of the one-dimensional wavelet transform described above. In a standard decomposition, a one-dimensional transformation is applied first to the rows and then to the columns (or vice versa) in order to obtain a wavelet transformation of a two-dimensional matrix.

The second approach is called non-standard decomposition. In this method, we perform a single step of horizontal pairwise averaging and differencing on the pixel values in each row of the image. Next, we apply vertical pairwise averaging and differencing to each column of the result. This process is repeated recursively only on the quadrant containing averages in both directions. Algorithm 3.1 gives the pseudocode for computing the wavelet transform.

```
procedure NSdecomposition (c:array[1..2^j, 1..2^j] of real)
   c ← c/2^j
   g ← 2^j
    while g ≥ 2 do
      for row ← 1 to g do
         decompositionstep(c[row, 1..g])
      end for
      for each col ← 1 to g do
         decompositionstep(c[1..g, col])
      end for
      g ← g/2
    end while
end procedure
```

Algorithm 3.1 pseudocode for computing wavelet transform

We have performed two dimensional non-standard wavelet decomposition for the images.

Clustering of sub-images

The number of sub-images for an image could be large in number. Therefore, storing the feature vectors or signatures for these sub-images could be quite expensive in terms of storage space as well as processing. Solution to this could be group sub-images with similar characteristics into a single cluster [50][52][53]. As the number of sub-images for each image will vary in number. An efficient clustering algorithm in terms of time complexity should be used. The algorithm should be insensitive to outliers i.e., the noisy objects which are not contained in any of the cluster should be discarded. For clustering a constant number c of well scattered points in a cluster are chosen first. The chosen scattered points are next shrunk towards the centroid of the cluster by a fraction α. These scattered points after shrinking are used as representatives of the cluster. The clusters with the closest pair of representative points are merged at each step. This clustering algorithm is less sensitive to outliers since shrinking the scattered points toward the mean dampens the adverse effects of outliers as these are typically far way from the mean. Each cluster thus contains a group of sub-images with similar characteristics (texture feature). The number of clusters depends on the complexity of the image. It identifies a region of the image with related pixel values. The query image is thus decomposed into a number of regions. Given a set of feature vectors $v = \{v_i,\ 1 \leq i \leq N\}$ the goal of the algorithm is to detect similar regions and labels the vectors based on the regions they belong to. The idea here is to perform wavelet transform and efficiently represent high dimensional data, as well as connected component analysis on this representation. The regions of each image are indexed using their signatures. As similarity matching is performed at the region level the algorithm can handle the images containing similar regions but region in one image is a translation or scaling of the matching region in the other image. For indexing, a hash table has been used.

The algorithm for clustering is as given below:

Input : *N image feature vectors*
Output : *The set of all detected region of Clusters*

1. *Perform preprocessing of Image by decomposing the image into sub-images.*
2. *Apply wavelet transform to get wavelet based feature vectors.*
3. *Find the connected components (Clusters) for regions.*
4. *Enter the regions consisting clusters whose feature vectors lies within* $\in$ *distance into hash table.*

Experimental Results

We have chosen the clustering based scheme, since it is fast and it allows to group together the regions that may not be homogeneous in color but overall have similar wavelet features. As we have used texture feature this clustering method suits our need. For clustering the value of α is assumed as 0.3. It has been observed experimentally the value 0.3 for α generated right clusters and the effects of outliers were dampened. The query image is also decomposed into number of regions as mentioned above. For matching of regions in the query image, all the regions whose signatures are within $\in$ distance of a region of the query are found from the image database. It computes all the pairs of matching regions for the query image and the image in the database. However, there is a possibility that the same query region may match a number of different regions in the image database. A solution for this when image database grows in size, is by adopting a strict definition for similar region pair set and the relationship between regions of query image and database image can be obtained by restricting the relationship to one to one correspondence, which prohibits a region from appearing multiple times in the similar region pair set. For assuming the value of $\in$ its value was varied from 0.05 to 0.09 for a fixed image and the response time was measured, the sub-image size considered was 32 x 32. As $\in$ is increased the average number of matching regions for a query image also increases consequently the query response time also increase. The query response time depends on the number of matching regions retrieved for a query image and the number of images in the database, which contains the matching regions. There are two measures recall and precision for evaluation of an algorithm:

Recall / precision defined below are essentially the measures of the relative sizes of the relevant and irrelevant sets between human judgement and system provided results.

Precision is the ratio of the number of relevant images retrieved to the total number of images retrieved. The ideal situation corresponds to 100% precision, when all retrieved images are relevant.

$$precision = \frac{|relevant \cap retrieved|}{|retrieved|} \tag{3.1}$$

Recall is the ratio of the number of relevant images retrieved to the total number of relevant images. We can achieve ideal recall (100%) by retrieving all the images from the database, but the corresponding precision will be poor.

$$recall = \frac{|relevant \cap retrieved|}{|relevant|} \tag{3.2}$$

Experiments are performed varying the value of Epsilon and recall is calculated. Results show that the optimal performance is achieved when the number of regions lies between 20 to 40 on an average. The recall does not improve much if the number of regions increase more than 40. Correspondingly we see that the number of regions extracted which lies between 20 to 40 has the value of $\in$ as 0.05, so we have considered the value for $\in$ as 0.05. Manually we tagged the similar images for a query image in the Brodatz texture album containing 96 images. So we could calculate the recall for the images which are visually similar. Figure 3.2 shows the execution time for computing feature vectors of sub-images of varying sizes. It may be seen from the figure that the execution time increases as the sub-image size is increased. In order to evaluate the efficacy of the algorithm we used images from the Brodatz texture album. The size of the images which we considered are 128 x 128. The execution time represent the actual time required for the computation of Haar wavelet for the sub-image, the reading time for image from the disk is excluded. As the sub-image size is varied from 2 x 2 to 128 x 128, the numbers 2^k along the x-axis represent a sub-image size of 2^k x 2^k. Since the size of the image is taken as power of two, the x-axis is represented in log scale. Similar texture images retrieved for the given query image using the proposed method are shown in Figure 3.3.

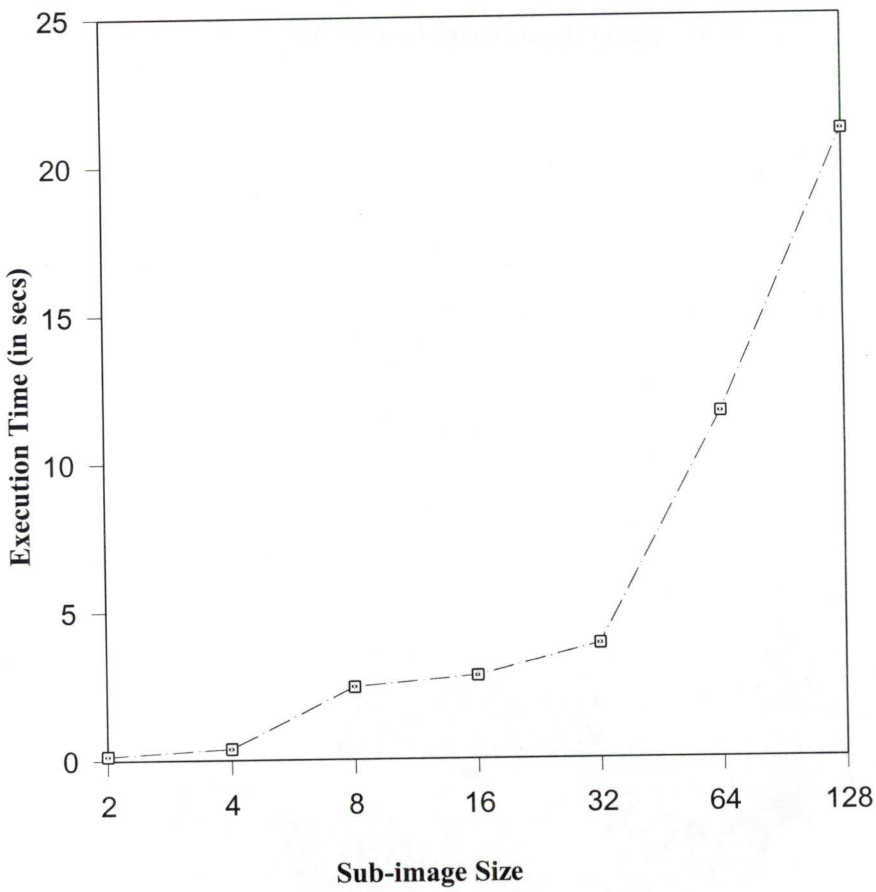
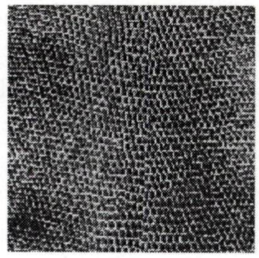


Figure 3.2 Execution time for computing feature vectors for sub-images of varying sizes



Query Image

Query Result

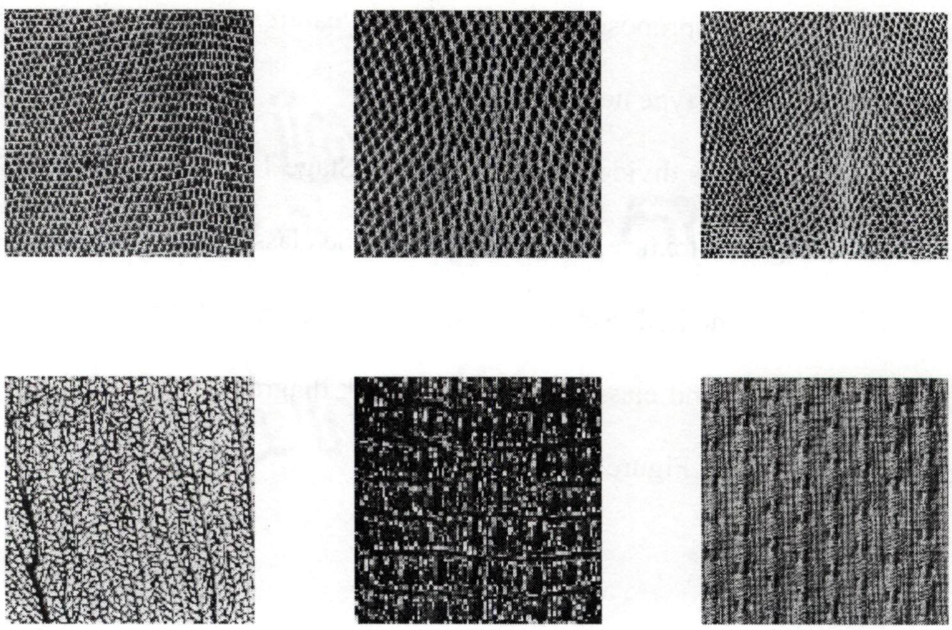


Figure 3.3 Query Image and the results obtained on applying proposed method

3.2.2 Neural Network Approach

A number of techniques have been developed for texture feature extraction, segmentation, classification and discussed in [39][55][59][60][92] and [125]. Artificial Neural Networks have been applied to many problems, and have demonstrated superiority over traditional methods when dealing with noisy or incomplete data. Research shows that the neural networks can be a good option for feature extraction and classification [51][65]. There are methods describing neural network based texture feature extraction and classification. Most of the methods are mainly based on Kohonen neural network. In the proposed method texture feature extraction is carried out using Multi-Layer perceptron type neural networks.

The proposed scheme is divided into two stages. Stage 1 performs the feature extraction from the sub images of texture. Stage 2 performs the classification of texture features into texture classes. The method uses a Multi-Layer Perceptron (MLP) based neural network for feature extraction and classification. The block diagrams for Stage 1 and Stage 2 are given in Figure 3.4 and Figure 3.5.

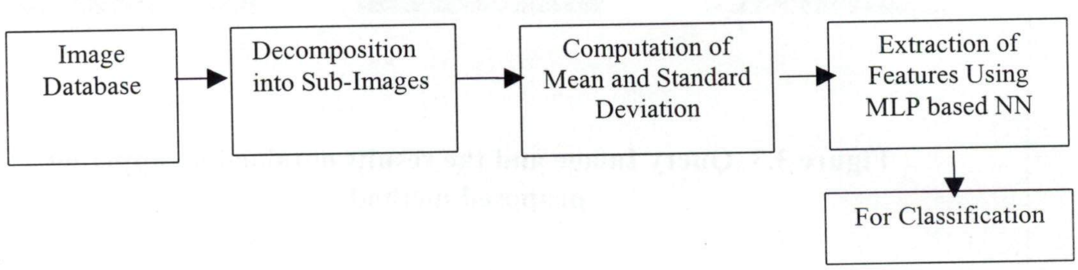


Figure 3.4 Stage 1: Feature Extraction using MLP based NN

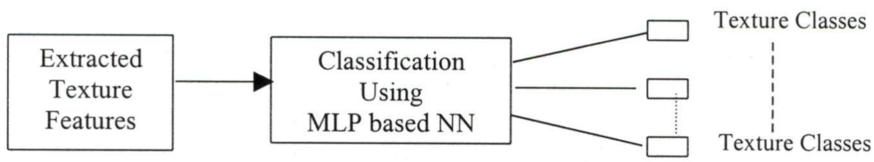


Figure 3.5 Stage 2: Texture Classification using MLP based NN

Feature Extraction using Multi-Layer Perceptron based Neural Network

The multi-layer perceptron based neural network for feature extraction is basically based on the concept of input units - hidden units - output units mapping, in which the input texture pattern and the output texture pattern are identical. The network learns the same patterns and its characteristics are provided through the hidden layer, which can be used as a feature vector. A single hidden layer feedforward neural network is used in the proposed scheme. It has m inputs, m outputs and p hidden units. The input and output texture patterns are same and the network is trained using a supervised learning approach. Figure 3.6 shows the feature extraction scheme using Multi Layer Perceptron (MLP) based Neural Network (NN) with single hidden layer. After the network is trained the hidden units values are considered as the feature vector. Noisy images are also considered and submitted to the network, the output for the noisy image is the same texture image without introduction of noise as shown in Figure 3.7. With this it can be possible to classify noisy image also. These feature vectors for the texture are submitted as input to the Multi-Layer Perceptron based Classifier.

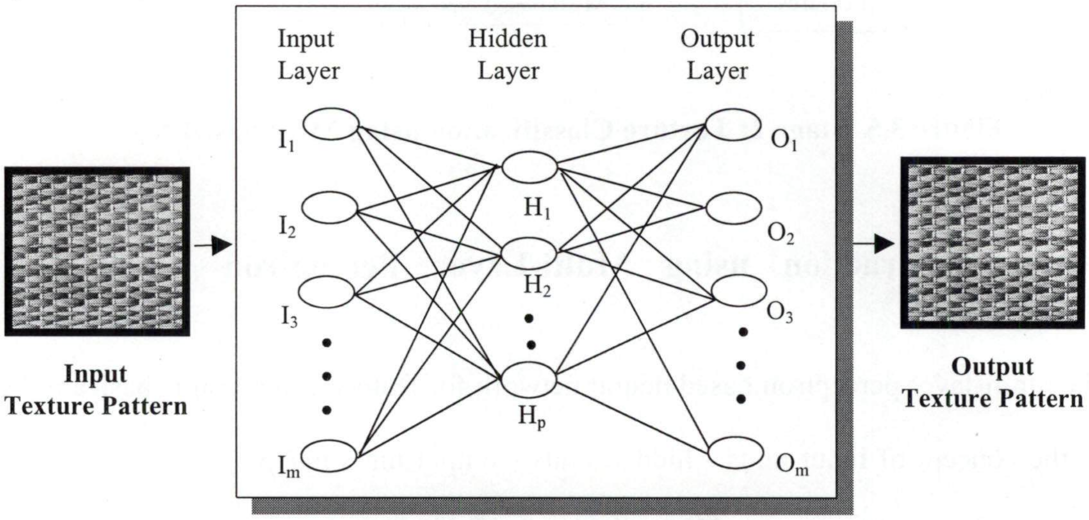


Figure 3.6 Feature Extraction Using MLP based NN

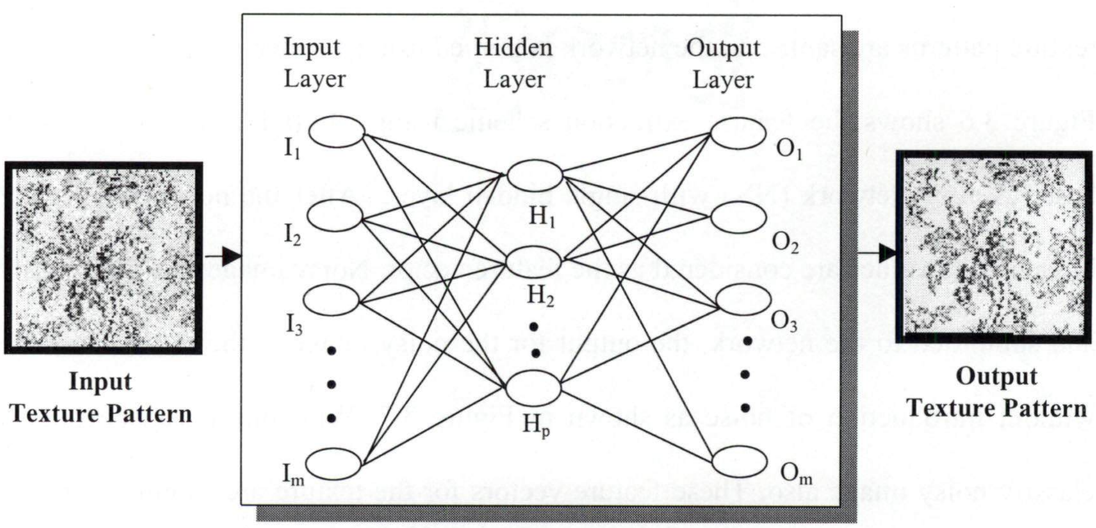


Figure 3.7 Feature Extraction Using MLP based NN with 25% Noise introduced to input texture pattern

Texture Classification using Multi-Layer Perceptron based Neural Network

Texture feature classification using an MLP based NN is shown in Figure 3.8. The numbers of inputs are m, same as the number of hidden units in the feature extraction neural network. The texture classes are classified into 32 classes. The number of outputs in the Neural Network is considered as 32.

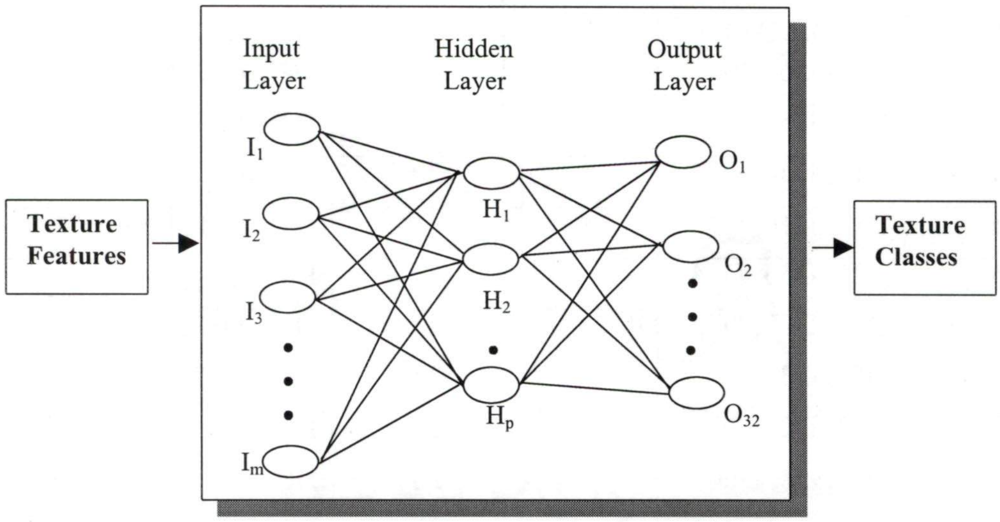


Figure 3.8 Texture feature classification using MLP based NN

- Each image $\boldsymbol{x}$ belongs to a set $\boldsymbol{I}$ of possible images.
- The set $\boldsymbol{I}$ is assumed to be partitioned into $\boldsymbol{K}$ classes, $\Omega = \{w_1, w_2, \ldots, w_k\}$, we assume image $\boldsymbol{x}$ from set $\boldsymbol{I}$ belongs to one and only one class.
- The classification problem can be stated as "Given a set of observed features $\boldsymbol{y}$, from an image $\boldsymbol{x}$, classify $\boldsymbol{x}$ into one of the $\boldsymbol{K}$ classes in Ω".

  where $\boldsymbol{y} = \{y_1, y_2, \ldots y_m\}$ denotes the set of $\boldsymbol{m}$ features based on which the classification procedure must operate.

Implementation of the Technique

The texture database from Brodatz texture album is used for evaluating the performance of the proposed technique. The Brodatz texture database contains 96 texture images of size 512 x 512. Total 96 texture images in the Brodatz texture database is grouped into 32 texture classes. The image patterns in a class are visually similar. The classification is performed manually. Table 3.2 shows the images in the similar texture classes.

Class	Texture Pattern	Class	Texture Pattern
1	D1, D6, D14, D20, D49	17	D69, D71, D72, D93
2	D8, D56, D64, D65	18	D4, D29, D57, D92
3	D34, D52	19	D39, D40, D41, D42
4	D18, D46, D47	20	D3, D10, D22, D35, D36, D87
5	D11, D16, D17	21	D48, D90, D91
6	D21, D55, D84	22	D43, D44, D45
7	D53, D77, D78, D79	23	D19, D82, D83, D85
8	D5, D33	24	D66, D67, D74, D75
9	D23, D27, D28, D30, D54	25	D2
10	D7, D58, D60	26	D86
11	D59, D61, D63	27	D37, D38
12	D62, D88, D89	28	D9
13	D24, D80, D81	29	D12, D13
14	D50, D51, D68, D70, D76	30	D15
15	D25, D26, D96	31	D31
16	D94, D95	32	D32

Table 3.2 Texture patterns used for classification into different classes

In order to limit the number of inputs to the ANN the image is partitioned into small images. To create small images, which belong to same class, each image of 512 x 512 is partitioned into 128 x 128 sub-images, thus 16 sub-images are formed. The mean and standard deviation is taken for each row of 128 pixels to reduce the size of input vector fed to the neural network. The mean μ and standard deviation σ for number of pixels $n = 128$ are calculated using the following equations:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.3}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{3.4}$$

For training set 12 sub-images have been used and the rest 4 sub-images are used for testing. The experiments are carried out in two stages, in stage 1 the training of the network is done for texture feature extraction, and in stage 2 the training of the network is done for classification. The design of the network architecture determines the number of input nodes, the number of output nodes and the number of hidden nodes. As images are partitioned into sub-images of 128 x 128. So the number of inputs and outputs are 128 x 2 = 256. The Brodatz texture database contains 96 texture images of size 512 x 512. The design of the network architecture determines the number of input nodes, the number of output nodes and the number of hidden nodes. A number of methods have been used for determining the number of hidden nodes in the feedforward networks, for example trial and error, evolutionary algorithms, and simulated annealing. Since the main focus here is investigating the strength and limitations of the basic approach with the backpropagation algorithm rather than testing a consecutive or stochastic search algorithm, we used the trial-and-error method for determining the number of hidden nodes. The number of hidden nodes needs to be determined empirically determined during network training and testing. It is difficult to give a formula to determine the number of the hidden nodes, however we used the following guideline: *"Use as few as* *possible"*. There are two reasons for this. First is the computation time. The more hidden nodes in the network, the longer the time taken in training the Neural Network. Second, possibility of over fitting that is using too many hidden nodes results in high accuracy on training set, but a high error rate on the test set. The network was trained with different number of hidden units and number of iterations to improve the feature extraction. Table 3.3 and Table 3.5 gives some of the results obtained for texture pattern after training the network for feature extraction and classification respectively, where as Table 3.4 and Table 3.6 gives some of the results obtained for texture pattern with 25% noise added to input texture pattern after training the network for feature extraction and classification respectively. The number of hidden units and number of iterations were varied to achieve better results. Figure 3.9 gives the learning profile for feature extraction, for the different values of momentum coefficients 0.3, 0.5 and 0.7 and learning coefficients 0.6, 0.7 and 0.8 respectively, the number of hidden nodes considered is 10. The minimum error 0.05796 is achieved with momentum coefficient = 0.7 and learning coefficient = 0.8. Similarly, the values for momentum coefficient and learning rate were varied, with different number of hidden nodes. The learning profiles (for feature extraction) with number of hidden nodes 15 and 22 are given in Figure 3.10 and Figure 3.11, it is clear from these figures the best results are obtained with momentum coefficient as 0.7 and the learning rate as 0.8. The learning profile for a noisy image for texture feature extraction is shown in Figure 3.12. The number of pairs for testing and training phase is 384 and 1152 respectively. In the proposed scheme texture feature extraction improves with different number of iterations and number of hidden units. We have performed experiments with noisy images also. We used noisy images with 25%, 50% and 75% noise added to them. The input for feature extraction is a noisy image of a texture pattern and the output is the same texture pattern without noise. Then the network is trained, it has been observed that the training time required is more in case of noisy images. Texture images with 25% noise can be classified correctly, but if we use images with 50% and 75% noise, the classification is not proper. Figure 3.13 and Figure 3.14 shows some of the texture classes and texture classes with noise added to them from Brodatz texture album.

Number of Hidden Units	Number of Iterations	RMS Error
10	1000	0.005796
10	10000	0.003165
15	10000	0.002190
18	10000	0.001787
22	10000	0.001481

Table 3.3 Results of Feature Extraction with MLP based NN

Number of Hidden Units	Number of Iterations	RMS Error
10	10000	0.006361
12	10000	0.004214
14	10000	0.004041
16	10000	0.003712
18	10000	0.002326
20	10000	0.002118

Table 3.4 Results of Feature Extraction with MLP based NN with 25% Noise added to input image

Number of Hidden Units	Number of Iterations	RMS Error	Classification of Images	Classification Rate
32	125000	0.008332	337/384	87.76%
32	100000	0.008614	332/384	86.45%
32	50000	0.008810	329/384	85.67%
30	50000	0.009112	278/384	72.39%
28	50000	0.009336	268/384	69.79%

Table 3.5 Results of Feature classification with MLP based NN

Number of Hidden Units	Number of Iterations	RMS Error	Classification of Images	Classification Rate
32	100000	0.008723	313/384	81.51%
30	100000	0.008789	302/384	78.64%
28	100000	0.008941	289/384	75.26%
26	100000	0.009346	267/384	69.53%
24	100000	0.009532	258/384	67.18%

Table 3.6 Results of Feature classification with MLP based NN with 25% Noise added to input image

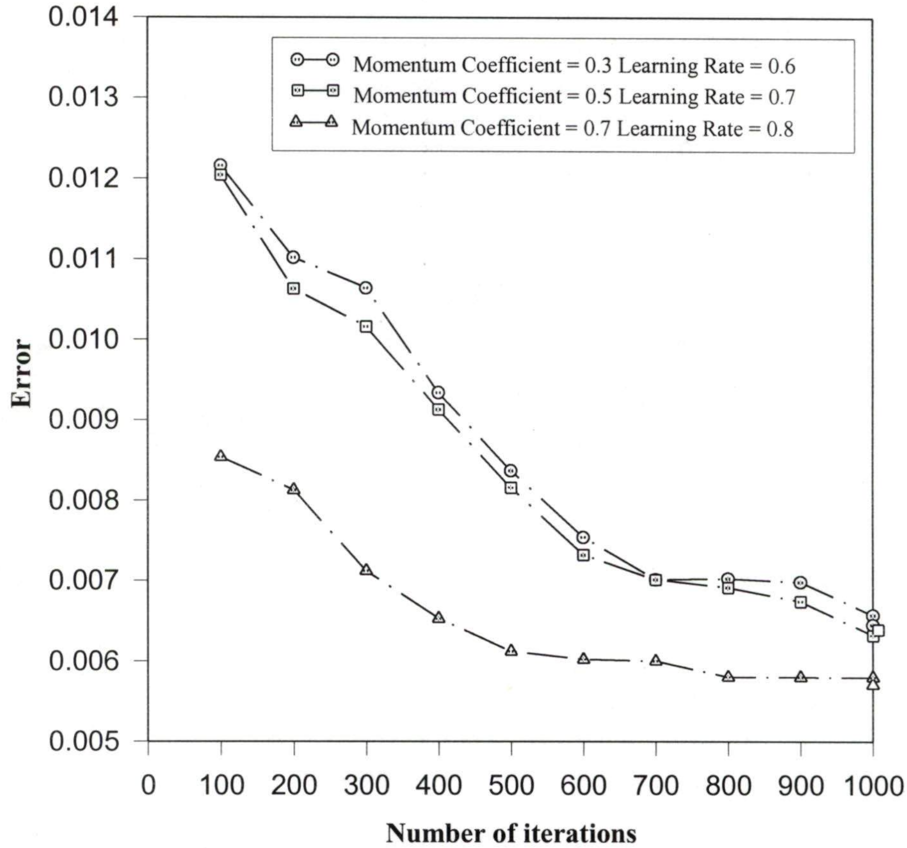


Figure 3.9 Learning Profile with hidden nodes = 10

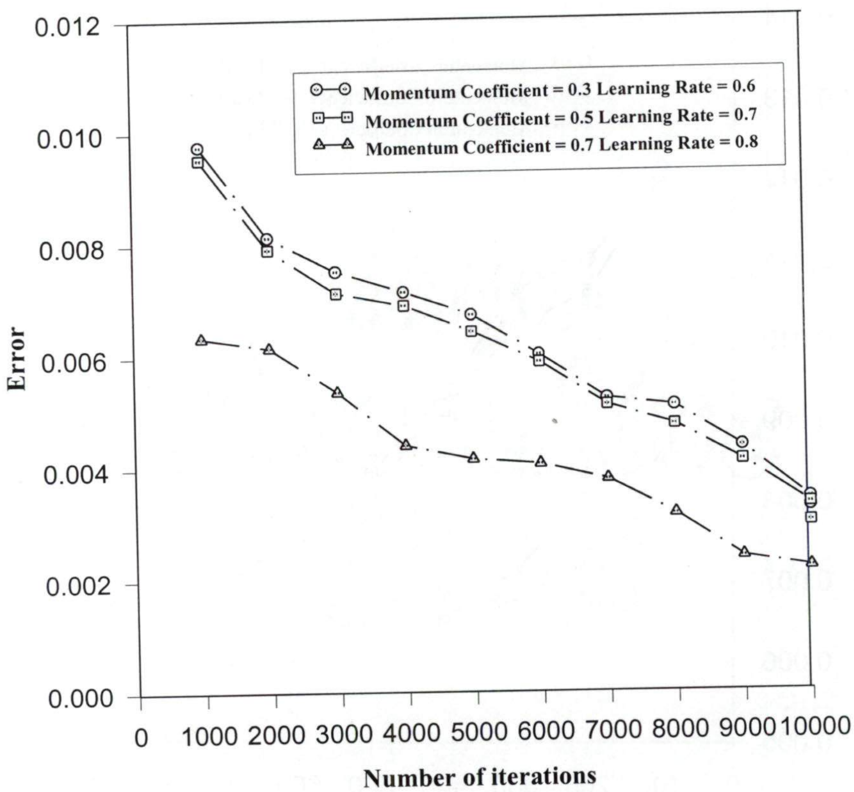


Figure 3.10 Learning Profile with hidden nodes = 15

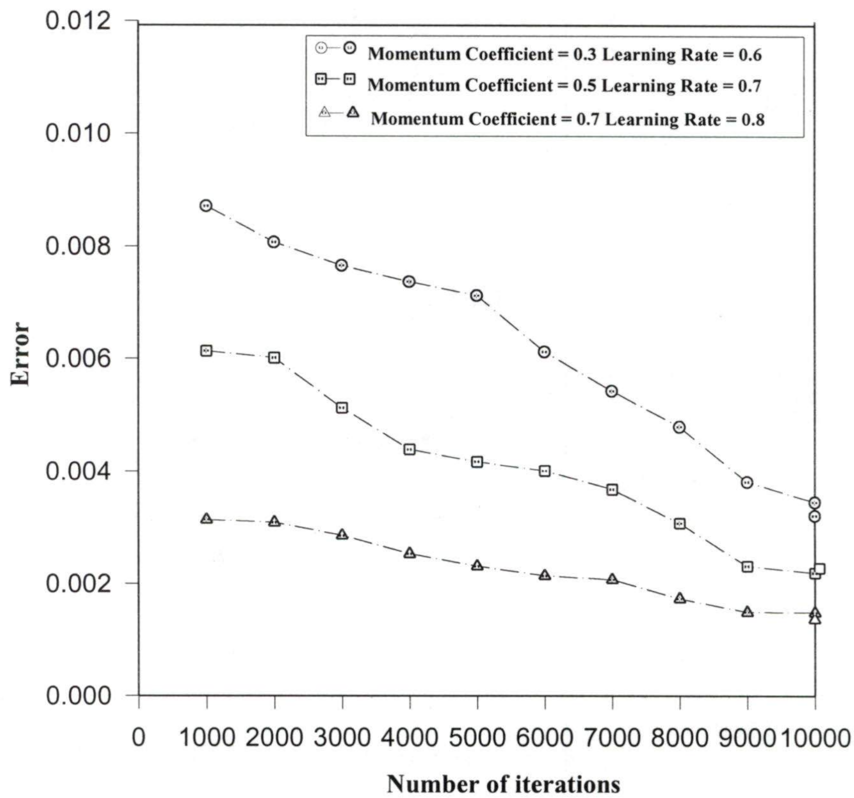


Figure 3.11 Learning Profile with hidden nodes = 22

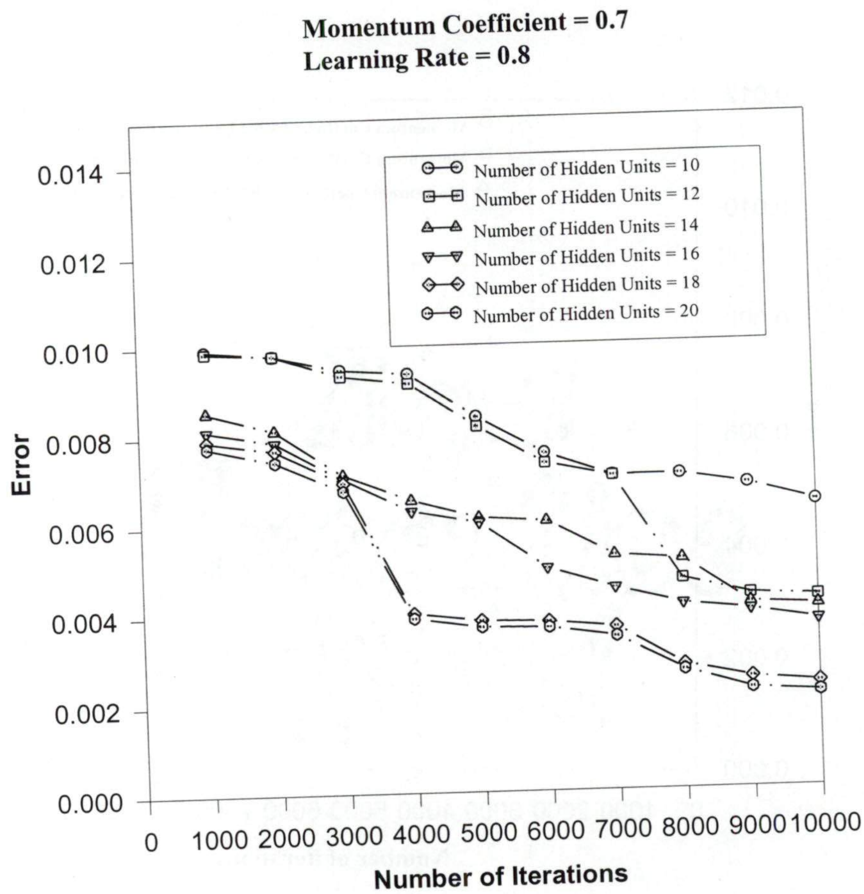


Figure 3.12 Learning Profile for a noisy image with 25% noise

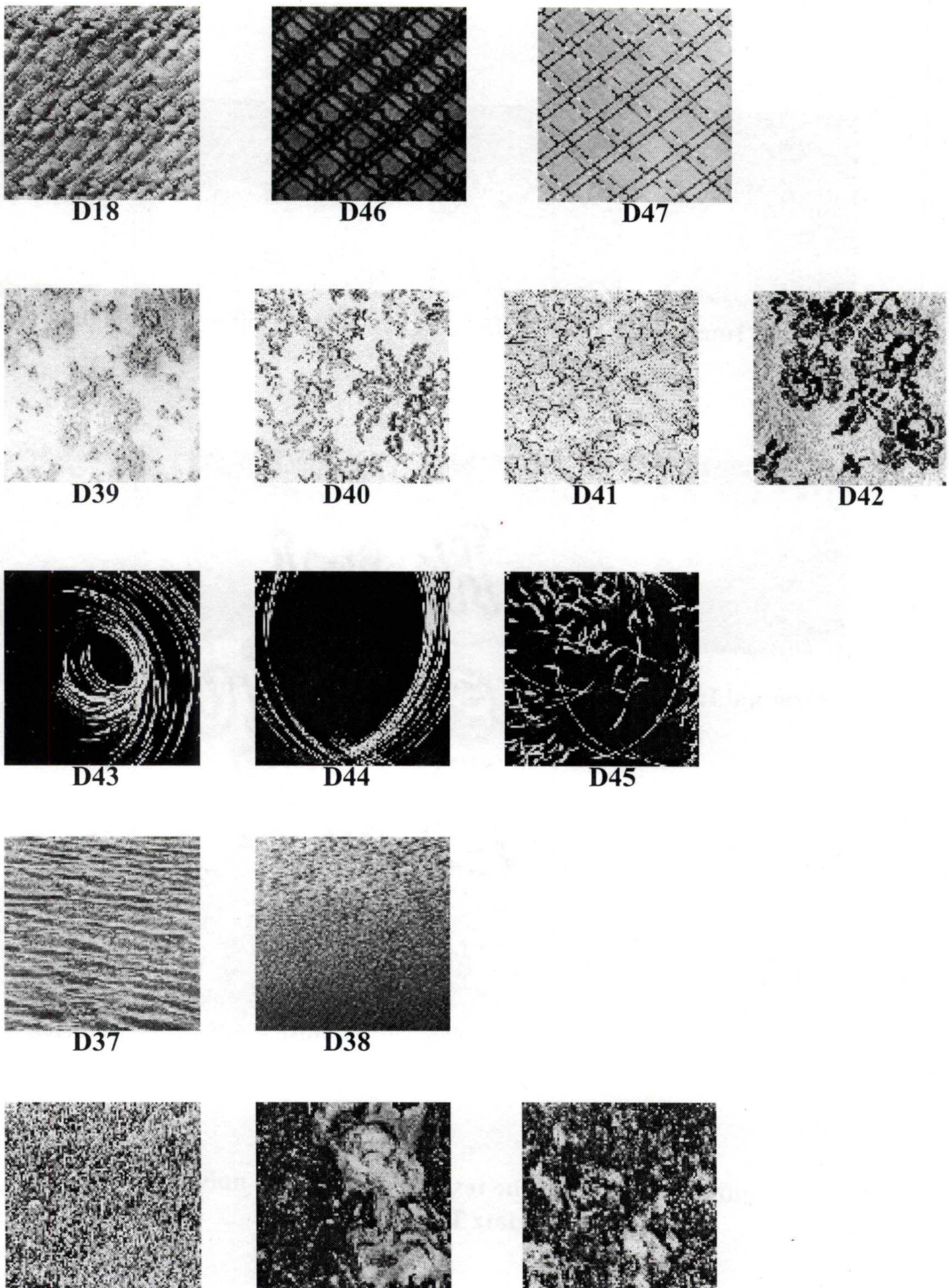


Figure 3.13 Some of the texture classes from Brodatz Album

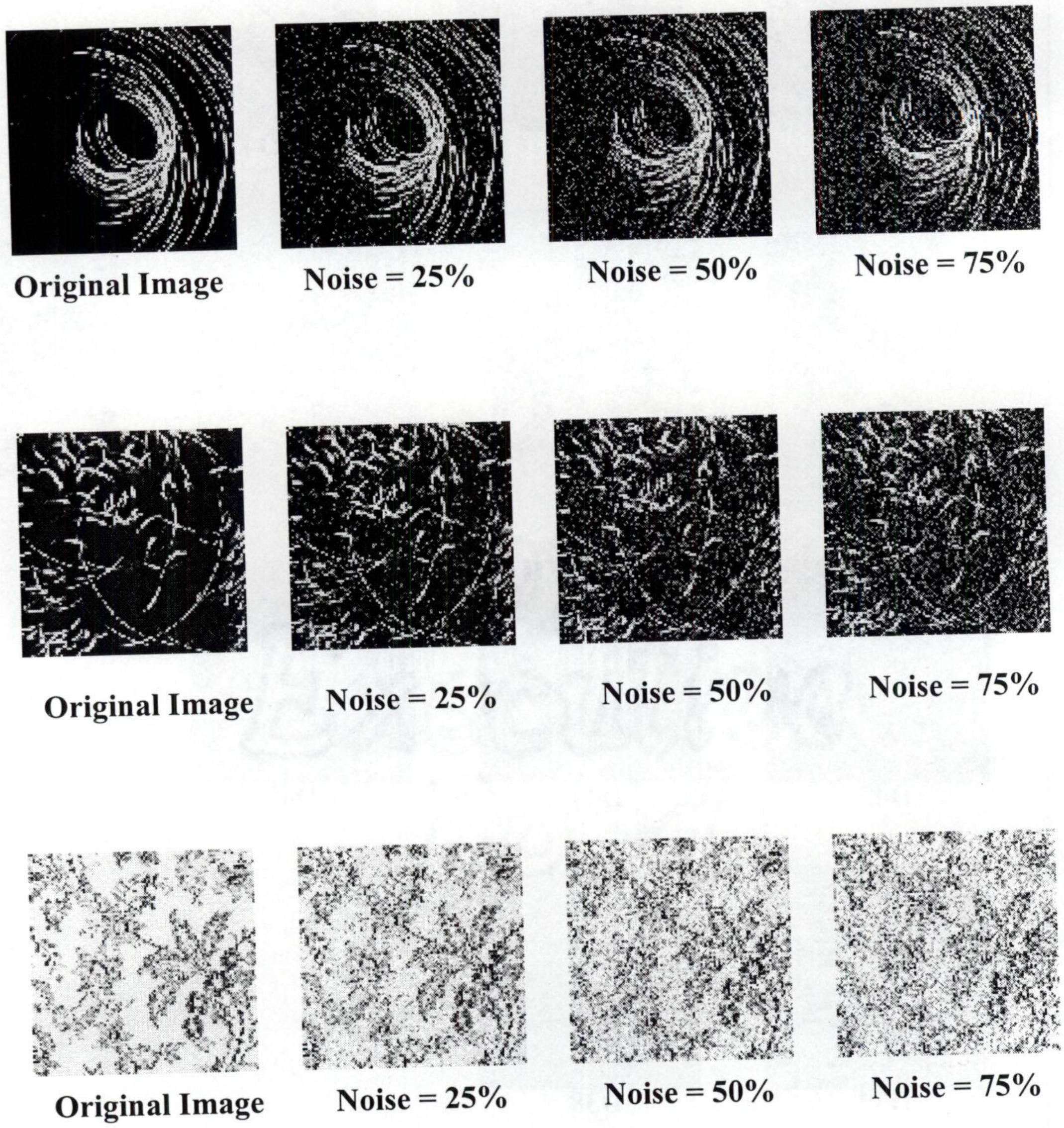


Figure 3.14 Some of the texture images with noise added to them from Brodatz Texture Album

3.3 Discussion of Results and Conclusions

In the first method, some aspects of decomposing an image using non standard wavelet decomposition and sub-images are used in order to search the objects of similar nature available in different regions of an image have been explored. The clustering algorithm uses space that is linear in the input size n and has a worst case time complexity of $O(n^2 \log n)$. For lower dimensions (e.g., two), the complexity can further reduce to $O(n^2)$. The factors, which effect the performance of algorithm, are image size and the size and number of sub-images generated, also the size of the feature vector or signature. A good choice of sub-image size and threshold value can greatly help in searching the database to get more accurate results. More reasonable ways of increasing the threshold dynamically may help in obtaining more accurate quality measurements.

In the second method based on neural network, experimentally it is shown that the texture features can be extracted using neural network and texture classes can be learnt using the low level image features. The results obtained have been compared with the results of the method proposed by Ma and Manjunath in [71]. It is observed that the results are comparable. Also with the proposed scheme the classification of noisy images is possible, the technique is able to classify images up to 25% noise correctly.

The experiments have been performed on Brodatz texture album, as it has 96 image patterns and testing the algorithm on these limited numbers of texture patterns is possible and manually it is possible to test the visually similar patterns.

Chapter 4

Retrieval based on Color Feature

4.1 Introduction

The color feature is the most dominant and distinguished in most of the applications. Traditionally, color images require large storage and high computational requirements. But with advances in technology, the computing and storage costs are rapidly decreasing. As a result, color images are increasingly being used in many applications. The foremost problem with most of the content based image retrieval systems is providing the user's needs to the databases for retrieval [3][11][14]. Histogram based color retrieval techniques suffers from important spatial knowledge [11][20][40][43][48]. The user can not provide the value for feature as these are either represented in the complex form such as matrices or huge numbers. As a result of it example based querying was introduced [23][32][105], which supports a query by example model for retrieval. It has been found that the user can only recognize a small set of feature values in general. Thus the proposed method uses color feature values from approximately (256x256x256) 16 millions to twenty seven colors which are within human perceptual range as described in [73]. If a user has to retrieve images based on color information, the user refers to a set of colors that fall within the human perceptual range [25][34][42][56][63][73][75]. We have considered the color feature values using reference color table method proposed by Mehtre et. al. in [73]. The proposed method takes care of "imprecision". The user always provides partial information while posing queries. However, it is not easy for the user to provide preferences in terms of numeric values. In the proposed method efforts have been made to model the imprecision using fuzzy interpretation [9][57]. The scheme employs a neural network based technique for image retrieval, which uses the color information stored in the features to obtain an overall impression of colors in the image. The neural network is trained using supervised learning approach.

4.2 Proposed Technique

This section explains the technique of the proposed Image Retrieval System. First the Image Archival phase is performed i.e., extraction of color feature from images and these features are stored in the Image database. Images from the Kodak album have been downloaded from World Wide Web (WWW) and the color feature extracted for these images are stored in the database. Let F_{ID} denote the set of all features used for representation of image. For example F_{ID} = {Color, Texture, Shape}. In the proposed technique we have used only color feature, i.e. F_{ID} = {Color}. In the next phase the query is submitted to retrieve the similar images in color from the database. The proposed method allows users to provide their preferences in linguistic terms. The query can be submitted in terms of natural language such as "mostly", "many" and "few" content.

Fuzzy logic has been employed to define "imprecision" in the query image. Figure 4.1 shows the block diagram of the proposed image archival and retrieval scheme. Reference color table for color feature extraction, has been used and only the colors that fall within the human range of perception are used as input to the neural network and the output is type of content. The output obtained after the training of neural network is used to compute the similarity factor for each image in the image database for a specific query.

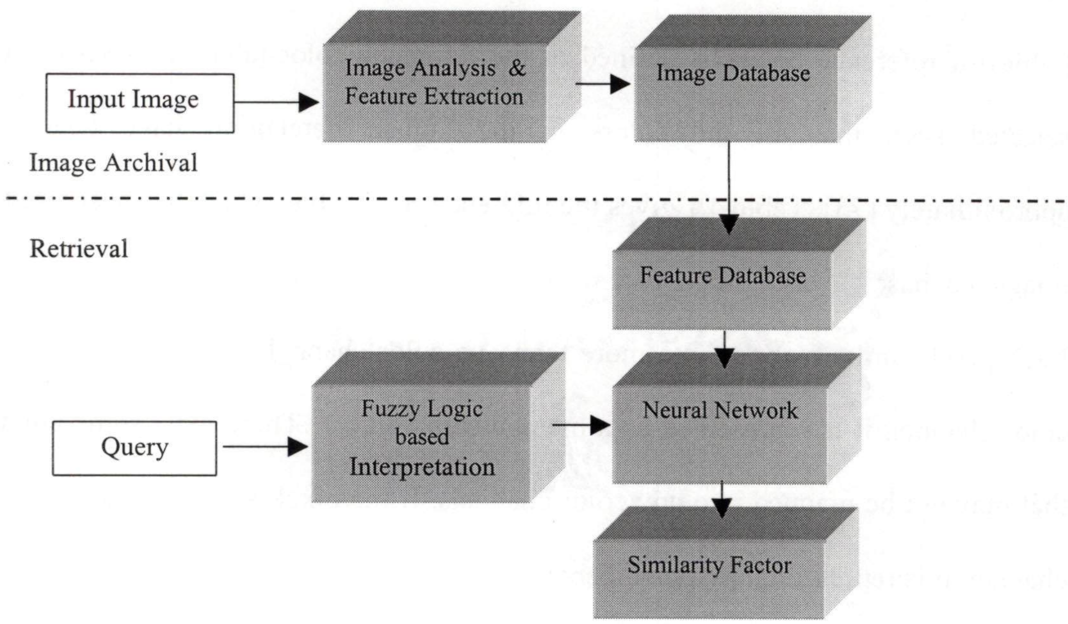


Figure 4.1 Block diagram of proposed scheme for color based image retrieval

4.3 Color Feature Extraction

Each image has a set of pixels. Let there be N_p numbers of pixels in each image. First we define a relationship between the pixel and F_u under the interpretation domain U as a mapping F_p: $N_p \rightarrow map(F_u, U)$ with $F_p(p,f) = \mu_p(f)$, where $p \in N_p$, $f \in F_u$, and $\mu_p(f)$ represent the membership function of p in the fuzzy set f. Each pixel in the image is mapped into RGB color space. The RGB color space is then mapped into HSV color space. The RGB and HSV values of each pixel is then passed to a filter, which finally are mapped into the following color channels: (Black, Darkblue, Blue, Darkgreen, Turquoise, Skyblue, Green, SpringGreen, Cyan, Brown, Violet, MarineBlue, OliveDrab, Grey, SlateBlue, LawnGreen, PaleGreen, LightCyan, Red, Maroon, Magenta, Orange, Pink, LightMagenta, Yellow, LightYellow, White). In this way, each pixel in the color image is mapped into the twenty seven color channels. For mapping each pixel into various colors the reference color table method has been used to store the color feature vector. A set (table) of reference colors is defined in the reference color table. This set of colors is selected such that all the colors in the human perception have been covered approximately [73]. Table 4.1 gives the reference color table, which has been used for the image database.

Each pixel can be mapped into more than one color channel. If the pixel maps into the color channel, it is represented as a membership value 1. There exist some color pixels that may not be mapped into any color channels. If the pixel does not map into the color channel, it is represented as a membership value 0.

Thus a relationship is obtained between the pixel and the feature representation set. Pixels that do not belong to a particular color channel, i.e., the membership value 0 does not have any significance.

In the proposed technique, the color information is extracted from the image and stored in the database using the summary representation system. The summary mapping is defined as follows:

$$g : I \rightarrow map(F_u, U) \tag{4.1}$$

in which F_u is a set of twenty seven colors, and I be the set of images in the databases.

Color	R	G	B
Black	0	0	0
DarkBlue	0	0	128
Blue	0	0	255
DarkGreen	0	128	0
Turquoise	0	128	128
SkyBlue	0	128	255
Green	0	255	0
SpringGreen	0	255	128
Cyan	0	255	255
Brown	128	0	0
Violet	128	0	128
MarineBlue	128	0	255
OliveDrab	128	128	0
Grey	128	128	128
SlateBlue	128	128	255
LawnGreen	128	255	0
PaleGreen	128	255	128
LightCyan	128	255	255
Red	255	0	0
Maroon	255	0	128
Magenta	255	0	255
Orange	255	128	0
Pink	255	128	128
LightMagenta	255	128	255
Yellow	255	255	0
LightYellow	255	255	128
White	255	255	255

Table 4.1 Reference Color Table

The summary information of a colored image shown in Figure 4.2 is given below:

S(i) = {(Black,0.02), (Darkblue,0.00), (Blue,0.00), (Darkgreen,0.14), (Turquoise,0.00), (Skyblue,0.00), (Green, 0.02), (SpringGreen,0.00), (Cyan,0.00), (Brown,0.00), (Violet,0.00), (MarineBlue,0.00), (OliveDrab,0.00), (Grey,0.00), (SlateBlue,0.00), (LawnGreen, 0.00), (PaleGreen,0.00), (LightCyan,0.00), (Red,0.81), (Maroon,0.00), (Magenta,0.00), (Orange,0.02), (Pink,0.23), (LightMagenta,0.00), (Yellow,0.00), (LightYellow,0.00), (White,0.00) }

Figure 4.2 Example Colored Image

This summary information can be interpreted as the image I belongs to a fuzzy set red with the membership value 0.81. The membership value 0.81 represents the ratio of total number of pixels that fall in the red channel to the number of pixels in the image. Generally user prefer to provide the queries in terms of natural language such as mostly, many and few. So we have assumed the content for each image as to be "mostly", "many" and "few". In the proposed model the interpretation domain is a fuzzy set [0,1]. The ranges of the values used are [0.9,1] for "mostly", [0.4, 0.5] for "many" and [0.1, 0.2] for "few". The color information for the image feature has been stored in the image database.

The algorithm is as follows:

Step 1: Given an image extract the color features for the image and store it in the image database.

Step 2: Given a query image q, extract its features $\{f_1^q, f_2^q, \ldots, f_r^q\}$.

Step 3: Train the neural network for Image features.

Step 4: Compare each query feature vector f_i^q *with the features of the images in the image database.*

Step 5: Return the images with the image numbers in the database and its corresponding similarity factor.

4.4 Discussion of Results and Conclusions

For testing the effectiveness of the algorithm collection of images from Kodak album downloaded from the World Wide Web has been used as Image Database. A three layer neural network has been used employing backpropagation algorithm. The training file prepared is related to queries in terms of mostly [0.9, 1], many [0.4, 0.5] and few [0.1, 0.2] and is trained on it. The training pairs generated in our experiments are 297. The values of the parameter, which provided optimum results, are as follows. Number of iterations and the number of hidden units were varied to achieve better results. Considering the Learning rate (η) as 0.3 and momentum coefficient (α) as 0.7 and number of hidden units as 10, optimum results are obtained. The output obtained after training is compared with each image in the database for computation of similarity factor. The results are obtained as the image number in the image database and its similarity factor. The more similar the image is in color, more is the value of similarity factor. The algorithm has been tested for color based queries. The complexity for feature computation for color is $O(M^2)$. The advantage of the approach is that the queries can be submitted in terms of natural language. Even if new images are added to the database, the network is already trained for the query image, and it will be able to compute the similarity factor for the new images added to the image database. The results obtained for color based queries are found to be quite satisfactory visually. Figure 4.3 gives the retrieval results based on color for natural scenes with the corresponding similarity factor of images. The retrieval results based for sunrise-sunset scenes with the associated similarity factor of images are shown in Figure 4.4.

Query Image

(i)

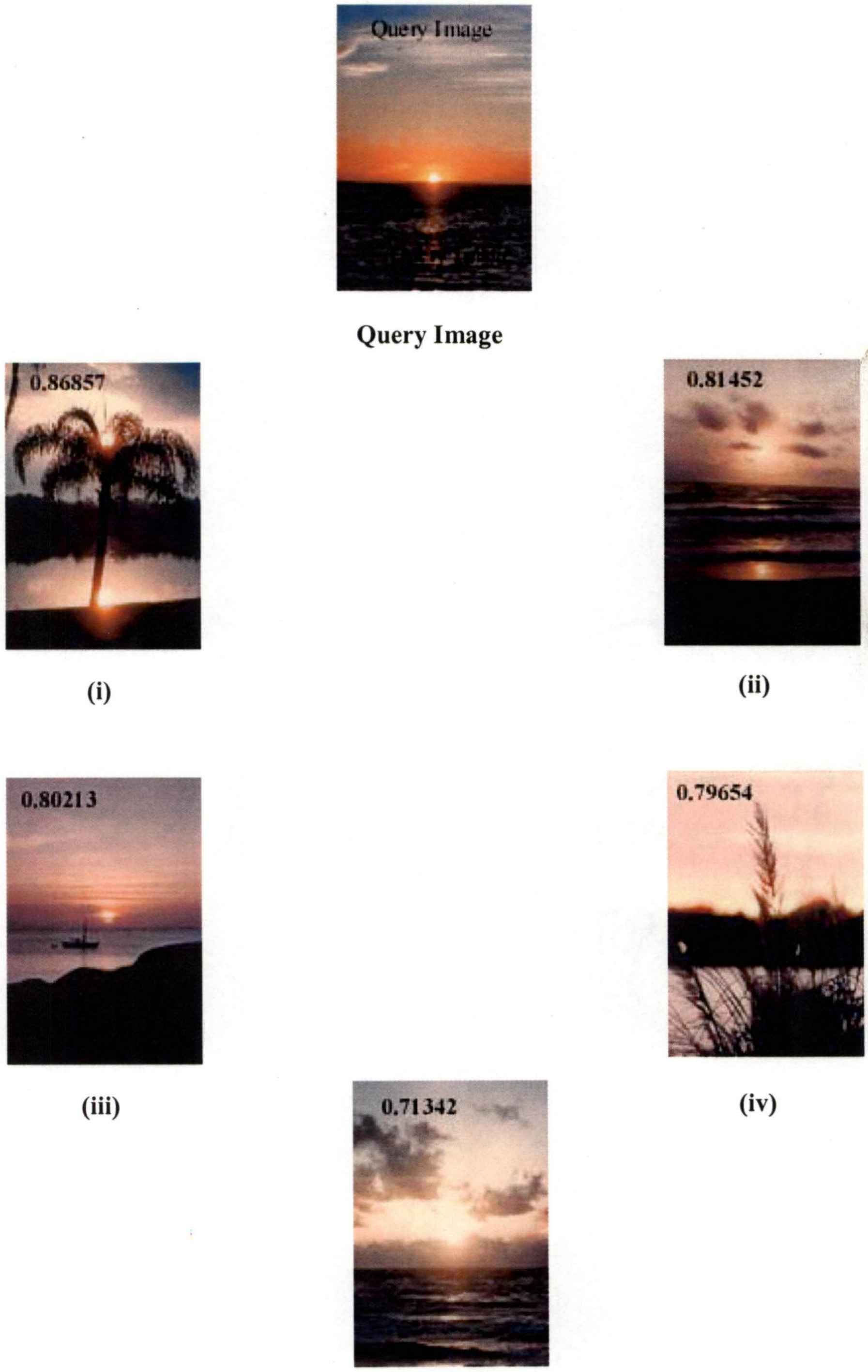
(ii)

(iii)

(iv)

(v)

Figure 4.3 First five similar images in color for natural scenes



Query Image

(i) (ii)

(iii) (iv)

(v)

Figure 4.4 First five similar images in color for sunrise-sunset scenes

Chapter 5

Retrieval based on Integrated Image Features

5.1 Introduction

A combination of image features for image retrieval is always needed because there is no single best feature that gives accurate description in any general setting. In image databases, each image is stored with its corresponding features. The features are chosen in the hope of capturing salient semantical information about the image. Many applications in the areas of computer graphics require to store and access large image databases as discussed in earlier Chapters. Color, texture and shape are important cues used for image matching and retrieval [122]. Serge and Malik have combined intensity with texture representation in [8]. Usually gray scale images are used for extraction of texture features. In most of the approaches the contribution of color to the texture is ignored. The similarity measure should conform to human perception i.e., perceptually similar images should have high value of similarity measure. For shape feature, the shape representation should be such that it should not lose the important shape information. Shape description or representation is an important issue both in image analysis for object recognition and classification and in image synthesis for graphics applications. Many techniques, including chain code, polygonal approximations, curvature, Fourier descriptors and moment descriptors, have been proposed and used in various applications [35]. Color and shape are the main features for human beings as well as computers to recognize the image. For all color based retrieval methods, there are some common issues such as, selection of a proper color space and the use of a proper color quantization scheme to reduce the color resolution. Wang et. al. [124] reduce the color resolution by hierarchical clustering, Color Naming System (CNS) merging, and an equalize quantization method. Swain and Ballard [118] used histogram intersection as color indexing. Wan and Kuo [123] use hierarchical color clustering method based on the pruned octree data structure. Most of the methods today employed for image retrieval suffer with the problem of high dimensionality. This leads to more computational time, and low performance. To overcome these problems dominant color region approach and color clustering have been proposed in [27]. As object shape is one of the important features of images, a number of shape representations have been used in content based image retrieval systems. In QBIC [3][32], moment invariants and other simple features such as area are used for shape representation and similarity measure. Moment invariants for trademark matching have also been used. But it is found that similar moment invariants do not guarantee perceptually similar shapes. A trademark shape description based on chain-coding and string-matching technique has also been used. The chain codes are not normalized and string matching is not invariant to shape scale. Mehrotra and Gary used coordinates of significant points on the boundary as shape representation [74]. The representation is not compact and similarity measure is computationally expensive, as these coordinates must be rotated to achieve rotation normalization. A region based approach that is translation and scale invariant is proposed by Lu and Sajjanhar in [69], using grid based approach for binary images, in which the edge and curvature details of object may be missed. A grid based approach for colored images is presented in [88]. They have compared it to Fourier Descriptor model and found their method to be better.

We have made an attempt to integrate the texture with the color feature, in which the color information is encoded into texture feature. We have also proposed an approach for retrieving similar images on the basis of dominant colors and local shape feature turning angle representation.

5.2 Proposed Schemes

Most of the existing Image Retrieval Systems are essentially limited by the way they function. So an attempt has been made to integrate the image features to obtain better retrieval of images. Two schemes are proposed, in the first scheme a method based on feature extraction, which integrates texture and color features is presented. A combined feature vector for texture and color is obtained which is more meaningful than the texture feature representation alone and significantly improves the performance. Fuzzy logic is used to represent the query as described in Chapter 4. The output obtained after training the neural network is used to calculate the similarity factor for the image features stored in the database. The second scheme combines the local shape feature representation based on turning angle integrated with color. In general, shape representations can be divided into two categories, boundary based, and region based. The most successful representations for these two categories are Fourier Descriptor and Moment Invariant. The main idea of Fourier Descriptor is to use the Fourier transformed boundary as the shape feature. Moment Invariant uses region based moments, which are shape feature invariant to transformation. A new scheme based on combination of methods to solve the image retrieval problems especially in shape representation and matching is proposed. This approach integrates shape representation with color and similar images are retrieved using the similarity measure. The retrieval performance is compared with that of the more established method based on Fourier descriptor (FD), and grid based method. The shape representation is invariant to rotation, translation and scale. The measure conforms to the similarity of human perception. An automatic indexing scheme of image database according to color quantization method, which could filter the image efficiently, is proposed. These two schemes are described in detail in the following sections.

5.2.1 Implementation of the Integrated Texture and Color Feature Vector Technique

The technique implemented is carried out in two phases. Phase 1 is the feature extraction and archival, Phase 2 is the retrieval of images.

Phase 1: Feature Extraction

For extraction of feature the concept of feature vectors which integrates color and texture feature is used for describing the details of pixels in an image. The regions extracted for an image are stored in the image database. The purpose of integrated feature vector is to combine the vector form of texture feature representation, derived from the responses to the image from a set of filters, with the color information. Feature vector combines the texture information with color and is derived by using a set of texture and color descriptors described below. A Gabor function and a Gaussian function as a color descriptor function has been considered. An integrated feature vector for a pixel *(x, y)* in the image *I* is then obtained by taking a vector of both color and texture descriptors for that particular pixel *(x, y)*.

Texture Feature Descriptors

A Gabor function is used to derive a set of filter banks for texture description. Gabor functions are Gaussian functions modulated by complex sinusoids. In two dimensions, the Gabor functions are as follows:

$$g(x, y) = \frac{e^{\frac{-1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi j\omega}}{2\pi\sigma_x\sigma_y} \tag{5.1}$$

where $j = \sqrt{-1}$, ω is the frequency of sinusoid and σ s are standard deviations (parameters of the Gabor function). A class of self similar Gabor wavelets by appropriate dilations and rotations of $g(x, y)$ through the generating function can be obtained.

$$
\begin{aligned}
& g_{mn}(x,y) = a^{-m} g(x',y'), \\
& a > 1, \quad m,n = \text{integer}, \\
& x' = a^{-m}(x\cos\theta + y\sin\theta), \\
& y' = a^{-m}(-x\sin\theta + y\cos\theta),
\end{aligned}
\tag{5.2}
$$

where $\theta = n\pi/K$, K is the number of orientations and $n = 0,1,\ldots, S-1$, S is the number of scales. Let U_l and U_h denotes the lower and upper center frequencies of interest. Then the following filter design ensures that the half peak magnitude support of the filter responses in the frequency spectrum touch each other.

$$
\begin{aligned}
& \sigma_u = 1/2\pi\sigma_x, \sigma_v = 1/2\pi\sigma_y, \\
& a = \left(U_h/U_l\right)^{-1/(S-1)}, \omega = U_h, \\
& \sigma_u = ((a-1)U_h)/(a+1)\sqrt{2\ln 2} \\
& \sigma_v = \tan\left(\pi/2K\right)\left[U_h - 2\ln 2\left(\sigma_u^2/U_h\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-1/2}
\end{aligned}
\tag{5.3}
$$

As considered in [72][73], for comparison we have also assumed for our experimental results, *S=4 (scales)* and *K= 6(orientations)*, and a filter size of *61x61* have been used. Given an Image *I(x,y)*, the transform coefficients are computed.

$$
T_{mn} = \iint I(x_1,y_1)g_{mn}^*(x-x_1, y-y_1)dx_1dy_1 \tag{5.4}
$$

where * indicates the complex conjugate. The texture descriptor for the given image *I* is then the vector of matrices,

$$
u_{texture} = [T_1, T_2, \ldots, T_N] \tag{5.5}
$$

where $N = S*K = 24$.

Color Feature Descriptors

A normalized Gaussian model to extract color descriptors of the images is used. The Gaussian model in n-dimension is represented as:

$$G_n(x) = K_c \exp\left(-\frac{1}{2}\left(\frac{\sqrt{\sum_{i=1}^{n}(x_i - \mu_i)^2}}{\sigma} \right)^2 \right) \tag{5.6}$$

In an image the total number of colors present is very large. There are over 16 millions colors in an image with 24 bit color. It is extremely difficult to model such a large number of colors. To limit the number of colors without loosing much information contained in an image from the point of user's perception, only limited number of colors are chosen following the experimental results in [70][77][83]. We have selected the Reference Color Table method described in Chapter 4 to quantize the color according to the sensation of human beings. Thus in order to represent color descriptors, twenty seven different Gaussian functions are generated by changing the value of μ_i. In the RGB color space, the 3-dimensional *(n=3)* Gaussian functions are formulated. μ represents the RGB values for a particular color. Suppose C_c is the response of the image I to a color c, the color descriptor for I is then given by the vector of matrices,

$$u_{color} = [C_1, C_2, \ldots, C_{27}] \tag{5.7}$$

By varying the value of σ the response of the Gaussian function can be tuned. For the experiments the values for $K_c = 1$, and $\sigma = 0.5$ are assumed.

Integrated Feature Vector

Integrated feature vector is obtained from texture and color descriptors. It represents the color descriptors alongside the texture descriptors. In the technique used, the length of an integrated feature vector is 51 (24 texture descriptors and 27 color descriptors). For a pixel *I* at *(x, y)* in an image, the feature vector is derived as follows :

$$u_{pixel}(x,y) = \left[u_{texture}(x,y), u_{color}(x,y)\right] \tag{5.8}$$

where $u_{texture}(x,y)$ is the 24 dimensional vector of $T(x,y)$ and similarly for color. Then the integrated feature vectors are normalized as follows.

$$\hat{u}_{pixel}(x,y) = \frac{u_{pixel}(x,y)}{\left\|u_{pixel}(x,y)\right\|_2} \tag{5.9}$$

The subscript 2 stands for Euclidean distance. The above mentioned normalized feature vector is used to generate feature representation.

Mean μ and the standard deviation σ of the energy distributions of the transform coefficient, calculated as follows represents a texture region [70].

$$\mu = \iint \left|u_{texture}(x,y)\right| dxdy \tag{5.10}$$

$$\sigma = \sqrt{\left|\left(u_{texture}(x,y) - \mu\right)^2\right| dxdy} \tag{5.11}$$

Then the feature representation for texture descriptor alone is given as follows as in [70].

$$r_{11} = \left[\mu_0, \sigma_0, \ldots, \mu_{23}, \sigma_{23}\right]^T \tag{5.12}$$

Similarly the feature representation from integrated feature vectors are extracted as follows.

$$\mu = \iint \left| u_{pixel}(x, y) \right| dxdy \tag{5.13}$$

$$\sigma = \sqrt{\iint \left| \left(u_{pixel}(x, y) - \mu \right)^2 \right| dxdy} \tag{5.14}$$

Thus, the feature representation for texture descriptor combined with color descriptors is represented as follows.

$$r_{12} = \left[\mu_0, \sigma_0, \ldots, \mu_{23}, \sigma_{23}, \ldots, \mu_{50}, \sigma_{50} \right] \tag{5.15}$$

For our experiments we have extracted regions using the integrated feature vector. These are then clustered using the clustering technique based on Euclidean distance between the feature vectors. The similarity between the clusters for merging was judged on the basis of threshold value for normalized Euclidean distance between their corresponding mean feature vector, which was assumed as 0.3. It is assumed that there are at most ten regions for an image. Using the above described method regions are extracted from an image and stored in the database as files.

Phase 2: Retrieval

In phase 2 the query is submitted to retrieve images having similar texture and color from the database. Natural linguistic terms are used for submitting the query as in Chapter 4. Fuzzy logic is used to define the query. The content for each image are assumed as to be "mostly", "many" and "few". In the proposed model the interpretation domain is a fuzzy set [0,1]. The ranges of the values taken are [0.9, 1] for "mostly", [0.4, 0.5] for "many" and [0.1, 0.2] for "few". A three layer neural network is used. Training set file is prepared, using the region file images extracted from the previous phase. The input to the neural network is query image (also a region file image) and output is the content type, the neural network is trained using backpropagation algorithm. The output obtained after the training of the neural network is used to compute the percentage of similarity for each image in the image database for a specific query.

Experimental Results

The algorithm has been tested on NASA images, the database contains around 200 images. The learning rate and momentum coefficient are assumed as 0.3 and 0.7 respectively. The number of hidden nodes assumed is 15. Some of the images with the significant regions extracted from the images on the basis of texture and color are shown in Figure 5.1. Experiments show that the system is able to retrieve image information of a very high satisfaction as can be seen in the figure. The performance of the proposed method is compared with that of the traditional Gabor method for texture extraction and the results are comparable. Figure 5.2 illustrates the retrieval performance for color and texture. It may be seen in the figure that the percentage of retrieving images for texture or color is 60% or above which is quite reasonable. Figure 5.3 gives the comparison of retrieval performance of the proposed method and that of texture feature extraction using Gabor method.

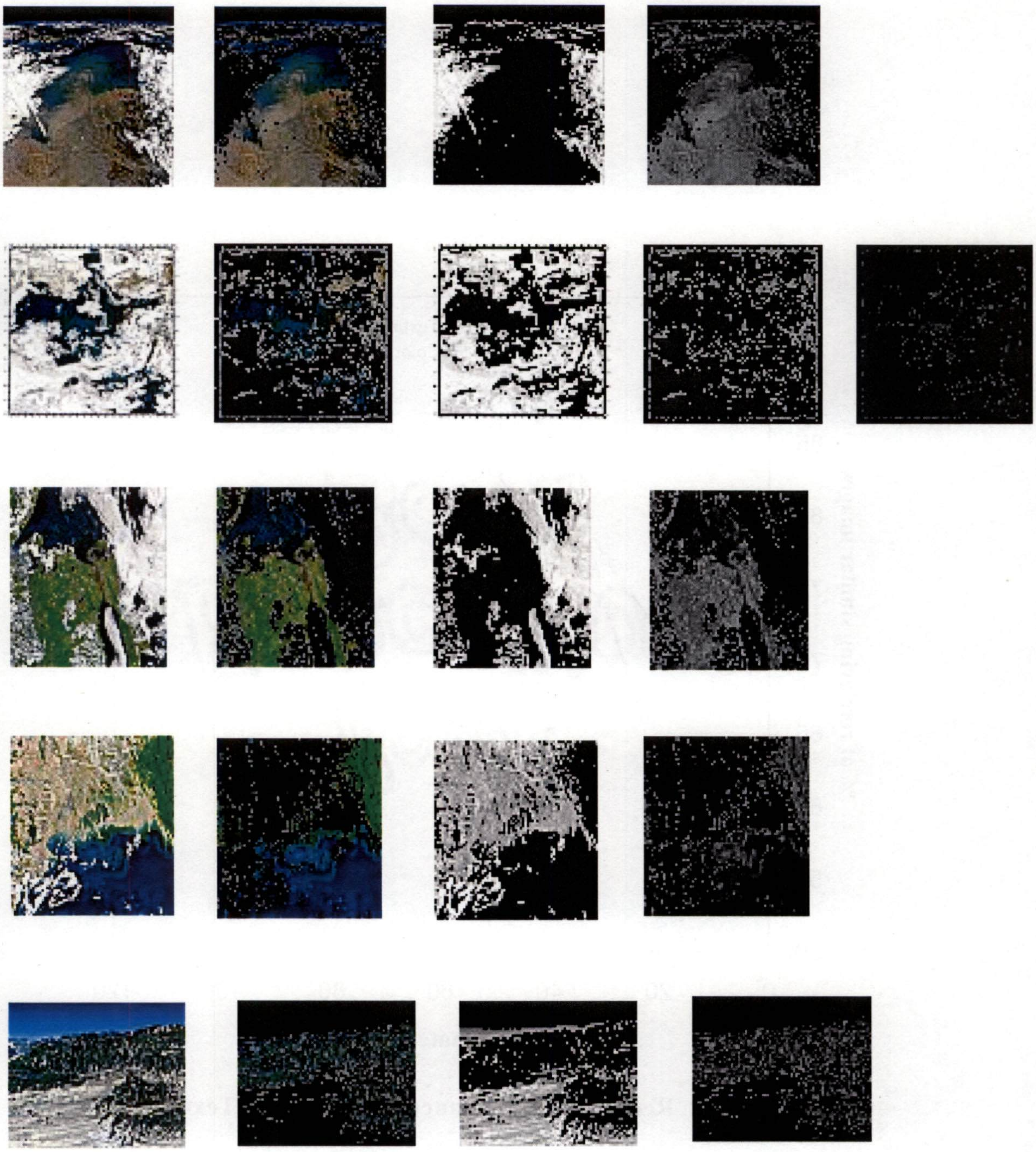


Figure 5.1 Some of the regions extracted from NASA images on the basis of Color and Texture

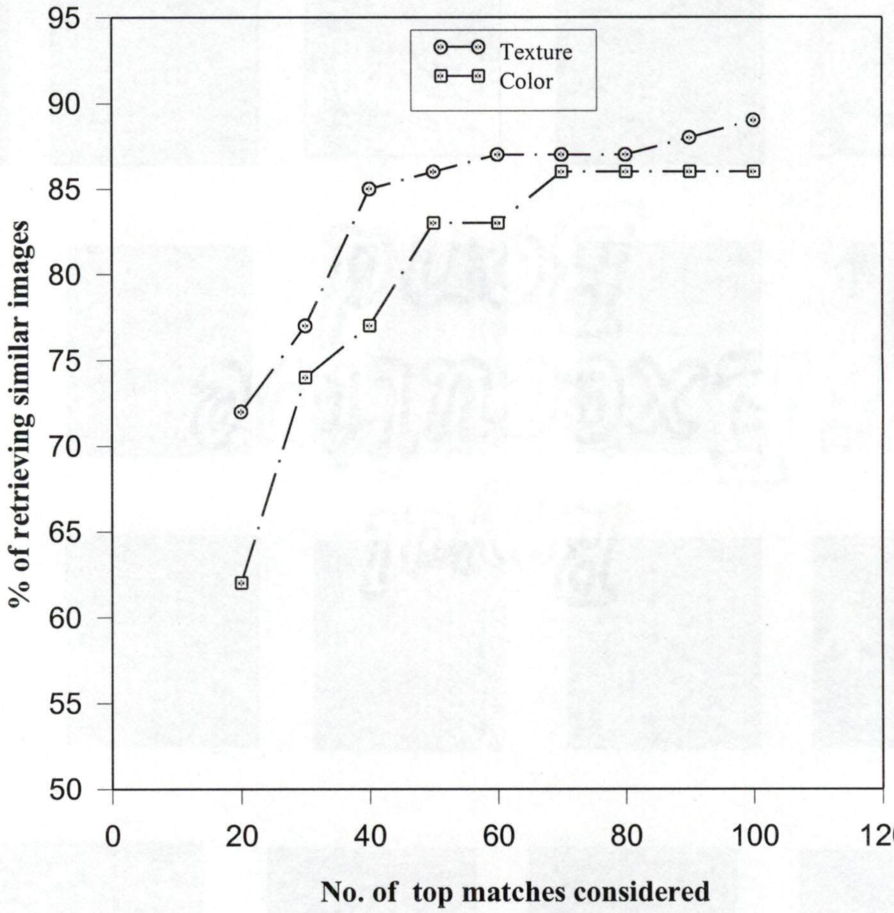


Figure 5.2 Retrieval Performance for Color and Texture

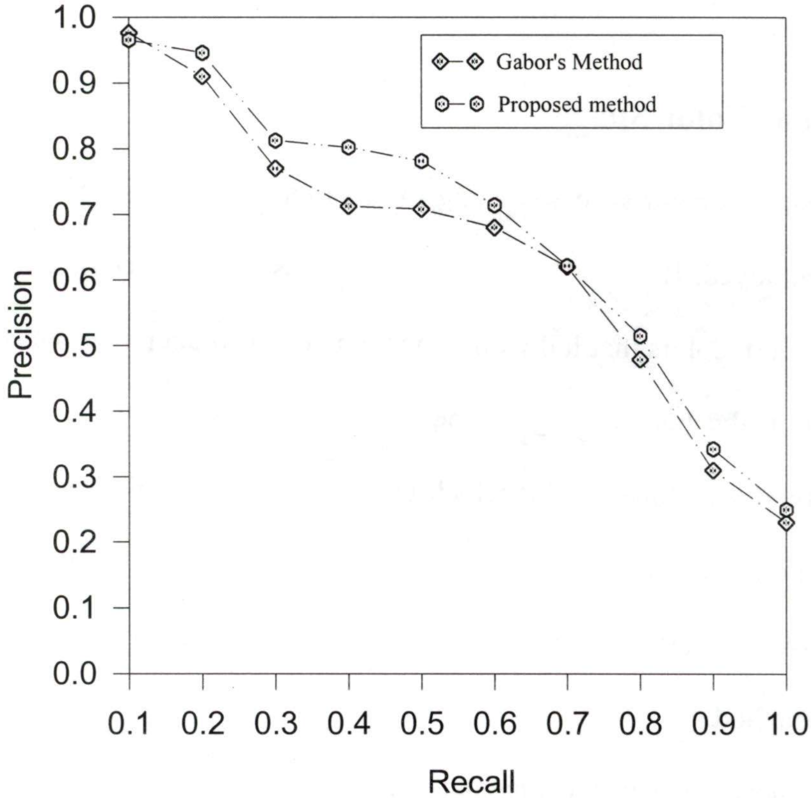


Figure 5.3 Comparison of Precision Vs Recall for Gabor method and the proposed method

5.2.2 Implementation of the Integrated Shape and Color Feature Technique

Color Quantization and Indexing of Image Database

The section describes the selection of color space, and the normalization of image. Besides, the indexing scheme and filter mechanism according to color quantization scheme to speedup the retrieval process is also described.

Selection of Color Space

There are several color spaces existing for a variety of reasons. In our study, RGB color space is employed. If a user has to retrieve images based on color information, the user refers to a set of colors that fall within the human perceptual range. Thus the color feature values within the human range of perception have been considered. Reference Color Table Method presented in [73] which uses twenty seven color covering approximately all colors perceptually has been used and is given in Table 4.1 of Chapter 4. The user always provides partial information while posing queries, this is taken care of in the proposed method.

Similarity means that two perceptually similar colors will be in the same or neighbor quantized color bin and two non-similar colors will not be in the same quantized color bin. Thus, the similarity of two colors can be determined according to the distance in the color space.

Color Quantization and Normalization

In the procedure of color quantization, firstly, we equally quantize the RGB color space to change color levels from 16 millions to 27 colors according to the reference color table. Finally, all the images are normalized with a resolution of 400*300 pixels.

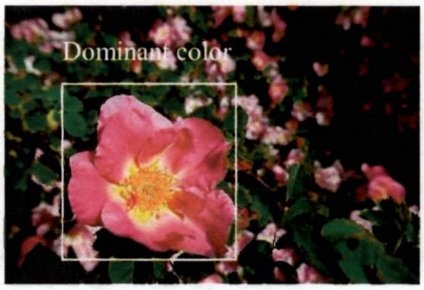


Figure 5.4 Image showing dominant color region

Indexing and Filtering Scheme

After the quantization and normalization, indexing of the images according to the dominant colors of the images is carried out. First, system reads the RGB values for each pixel in the image. Then a mapping is performed to the color mentioned in reference color table, which is near to the image pixel color. Figure 5.4 shows an image with its dominant color region. Histogram calculation on the basis of dominant colors (after mapping into 27 colors) of the image is performed. The color histogram is an array that is computed by differentiating the colors within the images and counting the number of pixels of each color. From the color histogram, we could choose the dominant colors whose numbers of pixels exceeds the threshold.

After getting the dominant colors, system will save the unique image ID to each corresponding color bin. According to this indexing scheme, system loads only the candidate images that have same dominant colors and eliminate irrelevant images immediately before the more complex and expensive similarity measure is performed. For a small image database, sequential searching of the image database during the retrieval process will be fast and will provide acceptable response time. However, it is not feasible for large image database. Therefore, a filtering mechanism to eliminate irrelevant images before the more complex and expensive similarity measure is proposed. First, system will load the image ID arrays according to the dominant colors of query image. Next, system will conjunct and rank the image ID arrays according to the number of appearance. After this step, system could filter out the irrelevant images effectively.

Feature Extraction

The section presents the proposed algorithms for extracting the features of image. After the feature extraction, the system will save its feature information and original image into the database.

Color Feature Extraction

The extraction of color feature includes two main procedures. First, system reads the RGB value of pixels from image file and maps it to the colors mentioned in the reference color table. A simple city block distance is used to compute the nearest color in the reference table. Then the histogram of the pixels with the newly assigned colors is computed. As the twenty seven color chosen covers mostly the colors which lies with in the sensation of human beings, the new image after assigning the nearest color from the table will perceptually be the same as the original image. So, the color feature chosen is the reduced color histogram based on the colors of the reference table. Therefore

$\overline{f} = (\lambda_1, \lambda_2, ..., \lambda_n)$ where λ_i is the relative pixel frequency (with respect to the total number of pixels) for the *ith* reference table color in the image. The size of the reference color table is n. This feature is computed for all the images in the database.

Shape Feature Extraction

There are three main steps of shape extraction: First, shape features are extracted using the block matching algorithm. Then, we normalize the shape and convert those shapes from region to contour by edge detection before the shape similarity measurement. In the first step, the color, location, height, width and area information of the objects are identified and recorded for the similarity measurement. The normalized contour generated by the second and the third steps is used in the measurement of shape similarity only.

A. Shape Extraction

The procedure of block matching algorithm for extracting the region feature is described below:

1. *Quantization and Normalization of the image.*
2. *Subdivide the image into a number of boxes on the chessboard. The size of each box is considered as 4*4 pixels and the representative color of a box is calculated by the average color of all pixels.*
3. *Starting from the upper-left corner, a box is chosen as an origin with the next box four units away both in the vertical and horizontal directions.*
4. *Extract the objects. Starting from an origin box, the program looks at the left, right, up, and down directions. And, the algorithm tries to combine as many boxes in a region as possible, if the color similarity between the origin box and the neighbor box is within a threshold.*
5. *Refine the image. Each region contains some boxes. Scattered small regions are removed, because in general it decreases the performance.*

Algorithm 5.1 Shape Extraction Algorithm

B. Edge Detection

Edge detection and edge tracing are very important tasks in segmentation application of an image processing system. There are two famous signal edge detectors: the Canny operator and the Shen-Castan Infinite Symmetric Exponential Filter (ISEF) method. The Canny algorithm [12] convolves the image with the derivative of a Gaussian function, and then performs non-maximum suppression and hysteresis threshold, the Shen-Castan algorithm convolves the image with the Infinite Symmetric Exponential Filter (ISEF), computes the binary Laplacian image, suppresses false zero crossings, performs adaptive gradient threshold, and also applies the hysteresis threshold. Experimentally, the Canny's edge detector claim to be the best that can be done under most certain specified circumstances. We have used Canny's algorithm from KUIM software for our experiments. The non-maximum suppression is a meant to ensure that the edge line is thinned and cost only one pixel wide.

C. Shape Representation and Normalization

Technically, edge detection is the process of locating the edge pixels, and edge tracing is the process of following the edges, usually collecting the edge pixels into a list. This is done in a consistent direction, either clockwise or counterclockwise around the objects. Chain coding is one of the methods of edge tracing. The result is non-raster representations of the objects, which can be used to compute shape measurement or otherwise identify or classify the object. But, rotation and scale of the object influence this method. We have used the turning angle representation to describe the object.

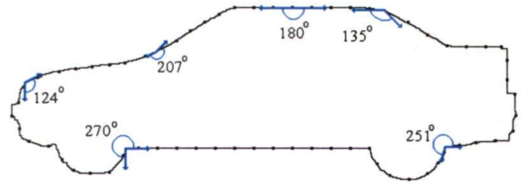


Figure 5.5 Turning angle representation of an object

Shape representation is a set of turning angles $\theta = \{\theta(1), \theta(2), \theta(3), \ldots \theta(N)\}$ as shown in Figure 5.5. This method is invariant to translation and scale of the object. In addition, it is invariant for rotation after the normalization. However, it requires the object to have one closed boundary. The turning angle of the object can express the edge's subtle difference including the curvature and distance. According to the turning angle variation, we classify feature tokens based on their curvature properties.

In the proposed method, the edge is segmented to be a fixed number of parts at first. And, the turning points are selected from the edge by computing the local maximum curvature points in each segment. So, the number of turning points in the image is fixed. This method is better than selecting the point at the segment point that may lose the important turning angle.

Similarity Measure of Features

The similarity measure of the images is obtained by integrating two components: color similarity measure named *CS* and shape similarity measure named *SS*. The overall similarity measure *S* is defined as following:

$$S = Wc * CS + Ws * SS \tag{5.16}$$

where *Wc*, *Ws* are the weights of color and shape, as it is difficult to assign weights precisely in terms of numeric values, we have used a method in which weight can be indicated by user in terms of "Very high", "High", "Medium", "Low" and "Very low" with a sample query image. So, the user can choose natural language interpretation as "Very high", "High", "Medium", "Low" and "Very low". The ranges assumed are taken as values within 0 to 1. Such as 1.0 as "Very high", 0.8 as "High", 0.6 as "Medium", 0.4 as "Low" and 0.2 as "Very low". The system converts the weight depending on natural language query submission.

Color Similarity Measure

We present the similarity formula of two colors first. Then, we define the similarity of two images according to the similarity of their colors.

A. Similarity of Two Colors

The perceptual color of a pixel can be specified by the corresponding color of the reference color table into which it maps. A color is selected from the 27 colors, which is very near to the image pixel value color and is stored as the new color of the image pixel. The color distance C_d can be calculated using city block distance formula. Assuming p the image pixel value and C the corresponding color entry in the reference color table, then color distance is calculated as follows:

$$C_d = \min_{i=1}^{27} \left| (p_r - C_{iR}) + (p_g - C_{iG}) + (p_b - C_{iB}) \right| \tag{5.17}$$

The similarity between two colors i and j is given by :

$$C(i,j) = C_d(i) - C_d(j) \tag{5.18}$$

B. Color Similarity of Two Images

The color similarity between two images is the difference of pixel numbers in the completely same color and the perceptual similar colors. In the first field, the similarity measure between query image Q and database image D for a color i can be determined as:

$$CSc(Q,D,i) = \min(Qi,Di) \tag{5.19}$$

And, the similarity measure between query image Q and database image D can be determined as following formula for all colors:

$$CSc(Q,D) = \sum_{i=1}^{27} CSc(Q,D,i)/400*300 \tag{5.20}$$

where 27 is the total number of colors in our color space.

In the second field, the similarity measure between query image Q and database image D for a color i can be determined as:

$$CSp(Q,D,i) = Max\left(\min(Qi,Dj) \times C(i,j)\right) \quad \forall j \in Cp \tag{5.21}$$

where Cp is the set of colors that are perceptually similar to color i. Then, the similarity formula between query image Q and database image D for all colors is:

$$CSp(Q,D) = \sum_{i \in CD} CSp(Q,D,i)/Qi \tag{5.22}$$

Finally, we define the similarity formula of the images according to the color feature.

$$CS(Q,D) = Max(CSc(Q,D), CSp(Q,D)) \tag{5.23}$$

Shape Similarity Measure

Given two shapes represented by their turning angle vectors $\theta 1$ and $\theta 2$. For the best match and partial similarity, we record the minimum distance of the two shapes by rotating one to match another. Then, the distance of similarity between these two shapes is calculated:

$$SS_D(\theta_1, \theta_2) = \min(\sum_{i=1}^{N} |\theta_1(i) - \theta_{2r}(i)|) \tag{5.24}$$

where N is the fixed number of turning points in the image and $\theta 2r's$ are in the shape set of rotating angle $\theta 2$. In this formula, if the segment is a straight line, we will set the turning angle to 180 degree for the segment.

Let the normalized similarity measurement degree from 1 (complete matching) to 0 (most dissimilar matching). The measurement between the requested image and archive images will be:

$$SS(\theta_1, \theta_2) = 1 - \frac{SS_D(\theta_1, \theta_2)}{N * 360} \tag{5.25}$$

where N *360 is the maximum distance measure of the requested image and archive images, SS_D is the minimum distance measure of the requested image and archive images. If the archive images include the requested image, SS_D is equal to 0, and SS is equal to 1. In addition, a threshold is set for disregarding dissimilar images. If SS_D is greater than the threshold, it means that the requested image is dissimilar to archive image. Then, we set SS to 0.

Image Querying and Retrieval

In this section, image querying and retrieval procedure is described. The same steps are applied for query image for feature extraction for color and shape as for other images.

In the query phase, user can retrieve the images by giving the sample image. The following is the procedure of image query:

Step1: User submits a query image with the options "Very high", "High", "Medium", "Low" and "Very low" are converted to weights.

Step2: Query image features are extracted.

Step3: After the features are extracted, feature information and the weights of query image will be sent to similarity measure mechanism and filtering mechanism.

Step4: Filter mechanism will get the feature record of relevant images from image database for detail similarity measure.

Step5: Similarity measure mechanism will measure the similarity of features information between the query image and database images and returns the results to users.

Step6: The most similar database images will have a high value of similarity factor. Along with the image name.

Algorithm 5.2 Image Querying Algorithm

Experimental Results

We have used flag database consisting of about 200 images. Each image is indexed first on the basis of dominant color. We have compared the result of the proposed color shape technique with the existing methods FD and Grid based method. It has been shown in [69][88] that the Grid based method has a better performance when smaller grid cell size is used and allows the generation of more accurate binary numbers. But the smaller grid cell sizes increases the storage requirements, and computation cost of similarity measure. The best grid cell size to achieve a good balance between performance, the storage requirements and the computation costs was found to be 12x12 pixels per grid cell. Therefore for our experiments also we employed a grid cell size of 12x12 pixels and the length of the standardized major axis was fixed at 192 pixels. The major axis was used as reference for scaling and rotation normalization. For Fourier descriptor method, the retrieval performance does not change significantly when using 8, 16, 32 and 64 coefficients to index a shape. This is due to the fact that the high frequency coefficients contain information about the fine details of the shape and not the global features of the shape. In our experiments, we used radius based signature with 64 uniformly sampled boundary points and thus 64 ordered radii are used as shape signature. In addition, in order to reduce storage requirements and computation costs only 8 low frequency coefficients for indexing and similarity measure of shapes was used in our experiments. The retrieval performance is measured using recall and precision for FD method, Grid based method and the proposed method. Recall is defined as the ratio between the number of relevant or perceptually similar images retrieved and the total relevant images in the database. Precision measures the retrieval accuracy and is defined as the ratio between the number of relevant or perceptually similar images and the total number of images retrieved. Some of the query results obtained on the basis of color and shape from the flag database are shown in Figure 5.6. The comparative retrieval performance for our method and that of FD and Grid based method is shown in Figure 5.7. The results shows that our method is effective.

Figure 5.6 Query Result from the flag database using the proposed method

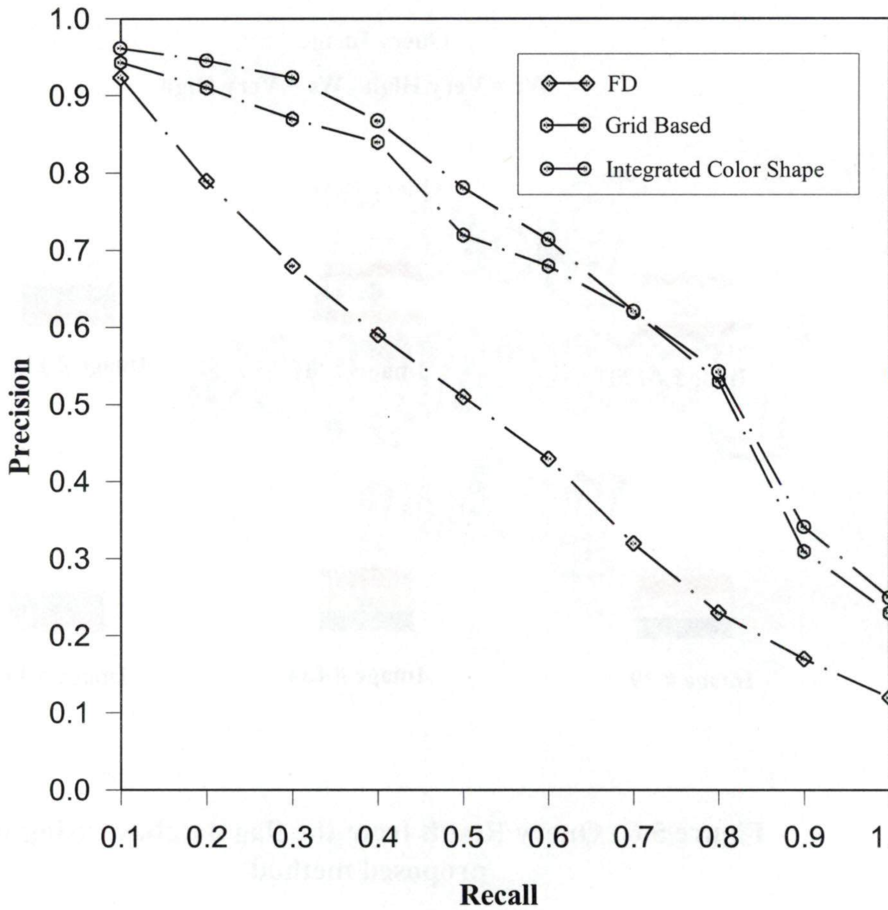


Figure 5.7 Retrieval Performance averaged over ten queries from flag database

5.3 Discussion of Results and Conclusions

An approach for feature extraction for pixels of image, which include color and texture representation is presented. A neuro-fuzzy approach using linguistic terms based on fuzzy interpretation has been used to classify each pixel of the image using the representation of integrated feature vector. The features used characterize different aspects of the texture in a small neighborhood of a pixel in the images. It has been found that the feature representation using the proposed method can capture the user information needs better than the texture feature only. The results obtained from the proposed technique are compared with the conventional Gabor filter method. The result of the approach appears to be acceptable.

The color, and shape features are used to retrieve images efficiently while searching the image databases. The color quantization method which quantize the colors into limited numbers which lies within human perceptual range are used and it indexes and filter the images while searching on the basis of dominant color. It also improves the database hierarchy for faster retrieval of images. The shape information takes care of the curves and details of the total shape of the image. The performance and accuracy are reasonable as shown in Figure 5.6 and Figure 5.7 for flag database.

Chapter 6

Frame Sliced Signature File based Indexing

6.1 Introduction

One of the ways to have fast retrieval of images is to have better indexing technique. Image databases are traditionally large which lead to resurrection of research efforts for efficient and effective retrieval of images from large image collections. An efficient storage mechanism ensures an efficient retrieval. Typically, such image retrieval systems extracts and stores visual features from a given query image, which are then used to compare the features of other images stored in the database. The similarity function is thus based on the abstracted image content rather than image itself.

The color feature is mostly used to compute the abstracted image content [42]. It exhibits the features, which are desired such as complexity of extraction, invariance to scaling and rotation, and partial occlusion. The Global Color Histogram (GCH) has been used widely for representing the distribution of colors in an image. GCH stores the feature vectors containing the percentage of colors in the image, the technique presented in [19] uses binary signatures to represent ranges of color percentages in the image. The binary signatures are more space efficient, and also have higher retrieval effectiveness [89]. So such signatures have been used in the image metadata.

In spite of how effective an image abstraction technique is, a linear searching for comparing all images in the image database to the query image is not practical. A Signature tree (S-tree) method is presented in [76][121] to use it more efficiently for indexing and searching color images. The signature methods presented in [29][121] also seem to be well suited to the concept of parallelism introduced with the development of massively parallel machine architectures. In these methods, queries are answered as follows, the query signature is compared with the set of stored signatures in the image database in a bit matching operation which identifies the required query images. Large image databases applications would benefit significantly by parallelism, especially in the bit-matching step. The results given in [26] are based on the assumption that the main memory is sufficient enough to hold all the signatures. In fact the algorithm does not extend well to the case of very large image databases that do not fit in the main memory.

Our motivation is to investigate the applicability of signature methods in parallel environment for very large image database applications, in which the signature file exceeds the main memory capacity of the computer system. The main contribution of this proposed technique is the presentation of a parallel signature method that fits better to a large image database applications. The method is based on the Frame Sliced Signature File (FSSF). When the signature size exceeds the main memory available in the computer system, the use of partial fetch slice swapping is proposed. The idea is to examine the frame slices already in the main memory and to fetch as many from the disk as necessary. A performance evaluation has been made with the existing techniques.

6.2 Binary Signatures for Colored Image

For an n-colored model of an image, a GCH is a n-dimensional feature vector $(h_1, h_2, \ldots, h_n)$, where h_j represents the percentage of pixels in an image corresponding to each color element c_j. The retrieval of similar images is based on the similarity between their respective GCHs. Euclidean distance (though other distance calculation method could also be used) is normally used to compute the difference between the abstracted feature vectors that represent the two images. While GCH provides a good abstraction of images, storing an n-dimensional feature vector for each image may require significant storage space. There are many other image abstraction techniques available [11][42]. The storage overhead of GCHs has been addressed using the binary signatures as an alternative to represent an image. Traditionally binary signatures are used to index text documents with the signatures representing individual words or collection of words in the document [47]. Similar signatures are presented for indexing color images in [19]. Each of these image signatures is a composition of binary sub-signatures representing the image colors. Typically in Image retrieval domain, the color space is quantized in reasonably small number of colors within the range of human perception. This eliminates the effects of small color variations in the image and avoids large files for high-resolution representations. An image signature is composed of n sub-signatures, one for each quantized color. Each sub signature is t bits long, each bit representing a range (in percentage) of colors in the image. These ranges may be of variable or equal size and the bit set in the color image signifies that the dominance of color c lies in a specific range. Through experimental results in [19], it has been shown that the so-called Variable Bin Allocation (VBA) scheme is superior as compared to GCH. Hence the VBA representation has been used in our proposed parallel frame slice signature file method. Some images and their respective binary signature representation are shown in Figure 6.1.

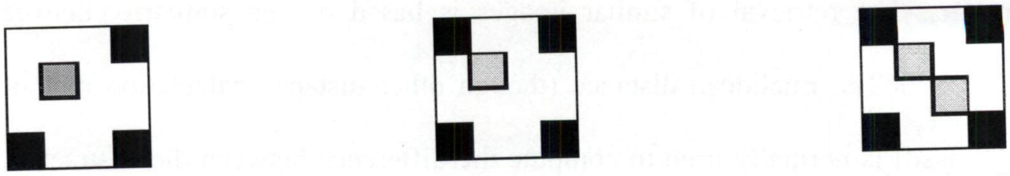


IMAGE A
VBA : 0000100000 0100000000 0000000001
GCH : (18%, 6%, 76%)

IMAGE B
VBA : 0000010000 0100000000 0000000001
GCH : (24%, 6%, 70%)

IMAGE C
VBA : 0000010000 0001000000 0000000001
GCH : (24%,12%, 64%)

Figure 6.1 Images and their abstracted binary signatures

6.3 Frame Sliced Signature Files

The idea behind the method is to explore a mechanism in a way in which the signature file can be partitioned effectively [128], so that efficient retrieval of image databases can be performed. For large image databases the storage medium is magnetic disk. The factors which affect the time of one disk access are the seek time (including the latency time), and the data transfer time of one block. Normally the disk access time is dominated by the seek time. Therefore, reducing the random access is more meaningful than reducing the amount of data to be transferred. With this requirement a vertical partition of the signature file can be obtained. In case of the representation used for the image with Variable Bin Allocation (VBA), $k=3$ frames can be assumed for each color R, G and B, each frame can store $s = 10$ bits. To access the block i a hash function can be chosen. With this assumption only one random access is required. Insertion will be much faster since access to only k frames is needed [129]. The method is referred as Frame Sliced

Signature File (FSSF), and the structure of a frame sliced signature file has been illustrated in Figure 6.2.

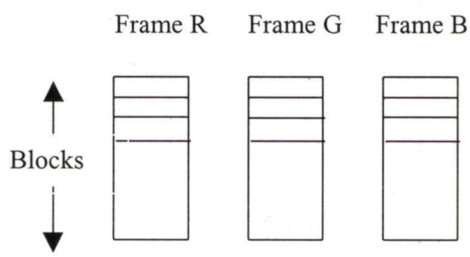


Figure 6.2 Frame - Sliced Signature File

6.4 Proposed Parallel Frame Sliced Signature File Method

This section describes the application of Frame Sliced Signature File method in a parallel environment, where multiple processors operate together to reduce the total time required for a computational task. Figure 6.3 gives the block diagram of the proposed Frame Sliced Signature File method. The main problem in designing a signature file organization is distribution of blocks in the frames among the specific number of processing units [127]. In case of image as only three color frames are sufficient to represent all colors, and the sub-signatures for R, G and B can be stored in the frames for R, G and B respectively. The optimization criterion is the query response time, which should be minimized with respect to all possible queries. However, it gives better results than other approaches proposed for indexing an image database. The insertion algorithm for the Parallel Frame-Sliced Signature File method is more efficiently executed in batches. It simply constructs the new binary image signatures and then appends the frames for the RGB color representation of an image using VBA method. For insertion of a binary signature in a frame a hash function can be used to determine the *i-th* block where the signature is to be inserted. After the address is calculated the VBA vector for R, G and B can be inserted in the respective frames. In case of VBA only one bit is set and a bit is appended to the partitioned signature which tells the position of set bit position. A pointer file can be maintained which contains the starting address of the block for the *i-th* set bit position. The similarity metric used for finding the similarity between the query image signature and the image signature in the database is as follows:

$$d(Q,I) = \sum_{j=1}^{n} \left[pos(B_Q^j) - pos(B_I^j)\right]^2 \tag{6.1}$$

where Q is the query image signature, I is another image signature, and $pos(B_R^m)$ gives the position of the bit set in the bin for color m of image R. The reasoning behind the squaring is to further accentuate the distance between corresponding sets of bins. Finally, using the obtained similarity distances, the image set can be re-ordered with respect to their ascending distances relative to the query image, the top ranked are presented as the query's answer.

In the proposed approach, the insertion and accessing is easy as only the block address is to be calculated the frames are fixed that is one frame for each color R, G and B respectively. For matching a query image, a binary signature for query image (using the VBA method as used for image signatures) is generated and is also hashed to access the block address. The set bit position for the query image VBA is compared with the set bit position in the frame, if the distance lies with in the threshold value the image can be added to similar image list, else the query image is compared with the images for the next set bit position. The starting address of the block (for next set bit position) can directly be obtained from the pointer file. Similarity between the query image signature and the image signature is calculated using Equation 6.1 given above, for each color. As the frames can be allocated in the memory associated with the parallel processors the similarity distance can be calculated in a parallel fashion and the sum can be computed.

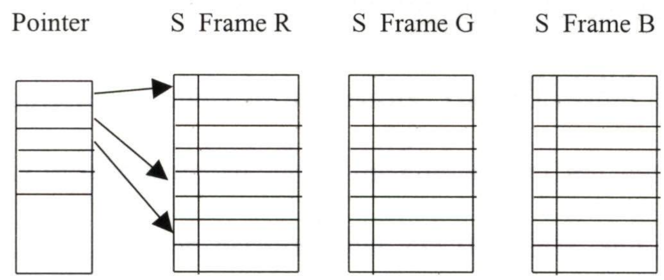


Figure 6.3 Block Diagram of Proposed Frame Sliced Signature File Method

The retrieval algorithm is presented in Algorithm 6.1. Given an image query, it produces a list of signatures that match the query image signature. It proceeds by matching it against the signatures in the database, finally false drops are eliminated. False drop probability is given as.

$$F_d = \textit{false drops} / (N - \textit{actual drops}) \tag{6.2}$$

which is the probability that a non qualified signature accidentally becomes qualified. A decision is to be taken how many blocks to fetch in the main memory. A page frame in dynamic memory allocation can be an optimum choice. The algorithm for retrieval of images using parallel frame sliced signature file is as follows.

1. *Initialize parallel memory with frames R, G and B.*
2. *Generate the binary signature using VBA for query image.*
3. *Determine the block address using the hashing function.*
4. *Compare the set bit position for the block with the set bit position in the Query image signature for colors R, G and B.*
5. *If it is with in the threshold difference, add it to the similar image list.*
6. *Get the address of the block for next set bit position from the pointer file.*
7. *Compare the next block. Go to step 4.*

Algorithm 6.1 Algorithm for Parallel Frame Sliced Signature File Retrieval

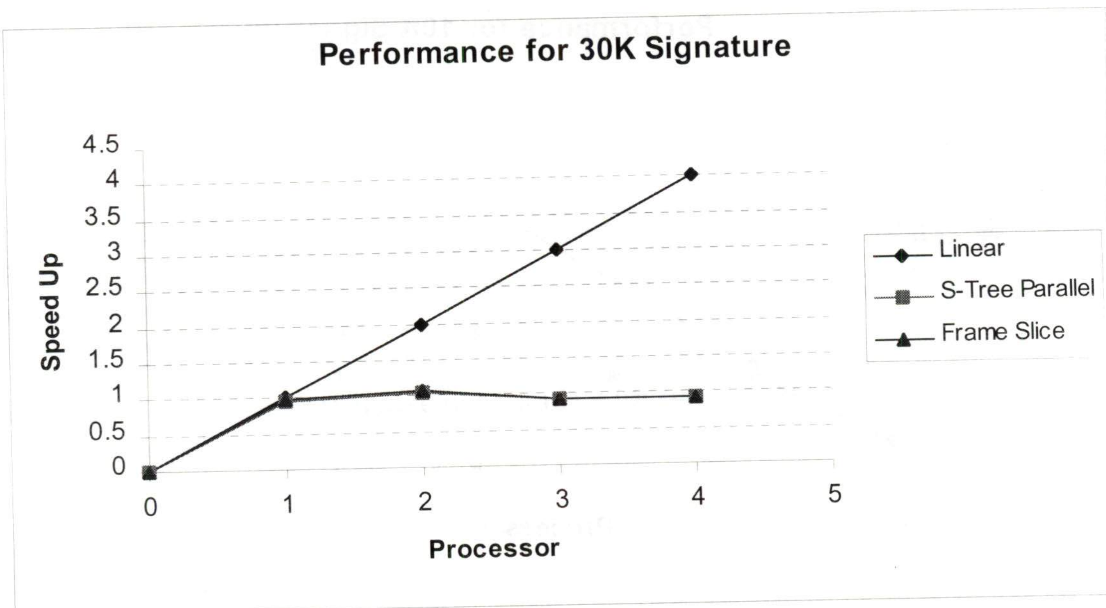


Figure 6.5 Speedup Performance for 30K signature

frames can be allocated in the memory associated with the parallel processors the similarity distance can be calculated in a parallel fashion and the sum can be computed.

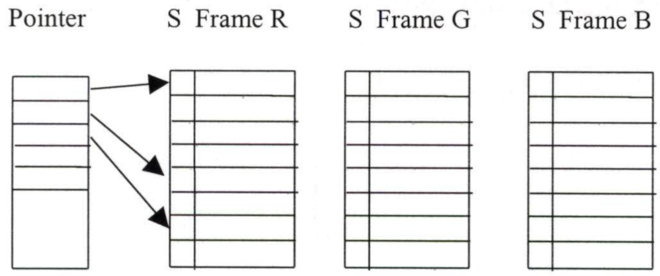


Figure 6.3 Block Diagram of Proposed Frame Sliced Signature File Method

The retrieval algorithm is presented in Algorithm 6.1. Given an image query, it produces a list of signatures that match the query image signature. It proceeds by matching it against the signatures in the database, finally false drops are eliminated. False drop probability is given as.

$$F_d = false\ drops\ /\ (N - actual\ drops) \tag{6.2}$$

which is the probability that a non qualified signature accidentally becomes qualified. A decision is to be taken how many blocks to fetch in the main memory. A page frame in dynamic memory allocation can be an optimum choice. The algorithm for retrieval of images using parallel frame sliced signature file is as follows.

1. *Initialize parallel memory with frames R, G and B.*
2. *Generate the binary signature using VBA for query image.*
3. *Determine the block address using the hashing function.*
4. *Compare the set bit position for the block with the set bit position in the Query image signature for colors R, G and B.*
5. *If it is with in the threshold difference, add it to the similar image list.*
6. *Get the address of the block for next set bit position from the pointer file.*
7. *Compare the next block. Go to step 4.*

Algorithm 6.1 Algorithm for Parallel Frame Sliced Signature File Retrieval

***False Drop Probability*:**

Fortunately, with VBA the false drop probability is very low, in case of image signatures, as with VBA only one bit is set.

***Response Time*:**

Considered to be the time to find the first qualified signature in the worst case, with in a given threshold distance. With the sequential assumption the response time is:

$$T_r = \textit{Time to read the block} + \textit{Time to compute similarity metric} \tag{6.3}$$

In worst case it will be N.

***Insertion of frame signatures*:**

Without any buffering

$$T_i = k * T_{seek} \tag{6.4}$$

A solution can be write the newly inserted signatures in main memory and write them to disks when a certain number of signatures are accumulated.

6.5 Discussion of Results and Conclusions

To evaluate the performance of our algorithm simulation is carried out assuming three processors. Our simulation is done in C/C++. Experiments are conducted on PARAM 10000 parallel computing system with Ultra Sparc II 64-bit RISC CPUs, 2GB RAM with SUN SOLARIS 5.6 Operating System. For testing the effectiveness four signature collections are used, each has 10K, 20K, 30K, and 50K signatures. The dataset has been downloaded from the website of University of Alberta. Table 6.1 to 6.3 shows the speedup performance of the S-Tree parallel traversal method and the proposed parallel frame sliced signature file method relative to sequential processing for a query on each of the datasets using different number of processors. The processing time does not depend on the algorithm but instead is dependent on the computers CPU and main memory speed (which may vary). The speedup performances for 10K, 30K and 50K signatures have been presented in Figure 6.4 to Figure 6.6. Speedup is a measure of how well a parallel algorithm scales and is calculated by dividing the sequential time by the parallel time. We have shown that the proposed approach for searching an image database using a frame sliced signature file method in a parallel environment is significantly better than sequential search. The proposed method has the advantage over the S-Tree parallel traversal method, as in case of large S-Tree the internal leaves are more, and processing them adds more relative overhead than for smaller tree. This proposed method could be useful for very large image databases up to the size of available hard disk. The only overhead in case of proposed approach is to append the set bit position to each frame. But this makes faster the computation of similarity metric, as each bit is not to be checked.

	10k Signature	20k Signature	30k Signature	50k Signature
S-Tree Parallel	0.828	0.9091	0.9552	1.9016
Parallel Frame Sliced Signature File	0.837	0.9312	0.9759	1.9190

Table 6.1 Speedup performance in seconds with one processor relative to sequential processing

	10k Signature	20k Signature	30k Signature	50k Signature
S-Tree Parallel	0.8134	0.9481	1.0441	0.8815
Parallel Frame Sliced Signature File	0.8277	0.9611	1.0574	0.8992

Table 6.2 Speedup performance in seconds with two processors relative to sequential processing

	10k Signature	20k Signature	30k Signature	50k Signature
S-Tree Parallel	1.3911	0.9068	0.9078	0.5573
Parallel Frame Sliced Signature File	1.4063	0.9126	0.9133	0.5716

Table 6.3 Speedup performance in seconds with three processors relative to sequential processing

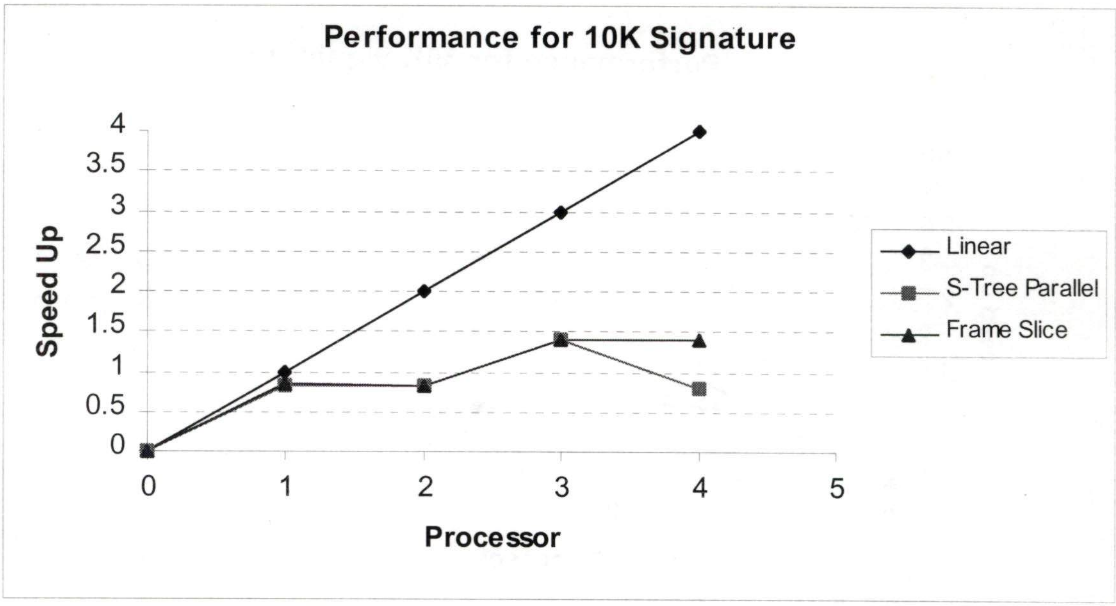


Figure 6.4 Speedup Performance for 10K signature

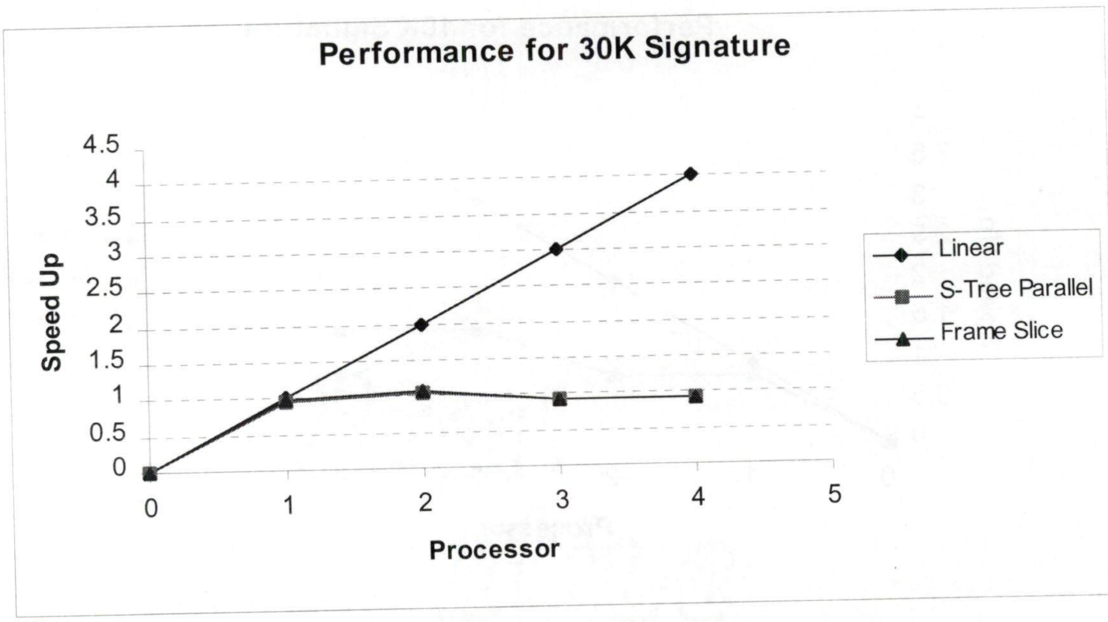


Figure 6.5 Speedup Performance for 30K signature

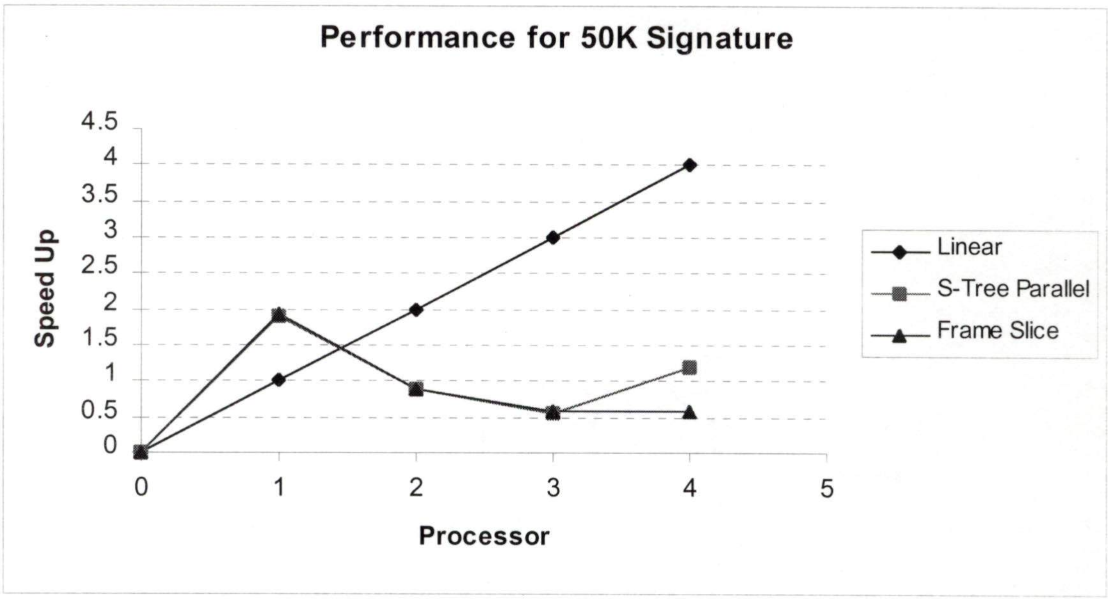


Figure 6.6 Speedup Performance for 50K signature

Chapter 7

Case Study - Biomedical Image Database

7.1 Introduction

The increasing reliance of modern medicine on diagnostics techniques such as radiology, histopathology, and computerized tomography has resulted in an explosion in the number and importance of medical images now stored by most hospitals. While the prime requirement for medical imaging systems is to be able to display images relating to a named patient, there is increasing interest in the use of Image Retrieval techniques to aid diagnosis by identifying similar past cases. One area where computers have scored great success in biomedicine has been in medical imaging. Probably the greatest medical advancement in the late twentieth century was the development of Computerized Tomography (CT) scanning techniques, which in many instances removed the need for exploratory surgery [86][91]. The same CT techniques that make image reconstruction possible using X rays have subsequently been applied to magnetic resonance imaging, a more sensitive technique for analysis of soft tissue and for metabolic studies. The recent development of digital radiography is replacing traditional methods of storing X-ray film, with direct computer storage providing the ability to transfer images from the office to the physician's home or to remote locations.

In medical imaging it has been observed that the use of visual texture conveys useful diagnostic information. However image processing modes based on scan sections or radiographic views do not completely provide diagnostic information in advance, when it would be easier to control a disease, make a therapeutic decision, or perform surgery [68]. This is due to the fact that gray level differences in tissues are small compared to the accuracy with which the measurements may be carried out for a reasonable patient – dose of X-rays. As there are many texture analysis methods available, it is possible to derive numerous texture parameters from a region of interest (ROI) in an image. Image retrieval is critically important in patient digital libraries, clinical diagnosis, clinical trials and pathology slides. Most of the existing image retrieval systems [31][85][109][110] are designed for general purpose picture libraries such as photos and graphs. Regardless of the imaging technology, all digitized images use the same general format. For example, in an image that is 512x512 has 512 pixels in each row with 512 rows, thus the image contains over 250,000 pixels. It is very important to extract the maximum possible information from any image obtained in terms of low-level image features color, texture, structure and shape. Texture analysis is also useful in medical applications. Doctors can classify lymphoproliferative disorders from the class of healthy leukocytes. Texture, shape and color are taken as key feature descriptors for the microscope images. Texture and color feature is used in almost all of these systems, and is generally accepted that these are the key features for image retrieval systems. Many texture representations have been presented and used in Image retrieval systems [30][71][109]. The texture features are obtained after converting the color image into gray scale, the contribution of color in texture pattern is generally ignored. Based on the insight gained, we have used the assumption of an integrated feature vector presented in Chapter 5, which encompass color information into texture features for medical image diagnostics. The feature vector derived is more meaningful to human perception than the texture feature representation alone and significantly improves the retrieval performance. The obtained feature vector is used for training the neural network for classification of biomedical images into different classes on the basis of fuzzy interpretation [39]. A fuzzy interpretation of natural language such as mostly, many and few is developed as described in Chapter 4. The neural network is designed to learn these. The proposed method investigates the use of texture extraction filters to include the color information. The soft tissues such as gray matter, white matter and cerebrospinal fluid in the brain can be differentiated based on the color and texture present in the MRI images. The proposed technique can assist the medical community in diagnosing the disease.

7.2 Case Study and Experimental Results

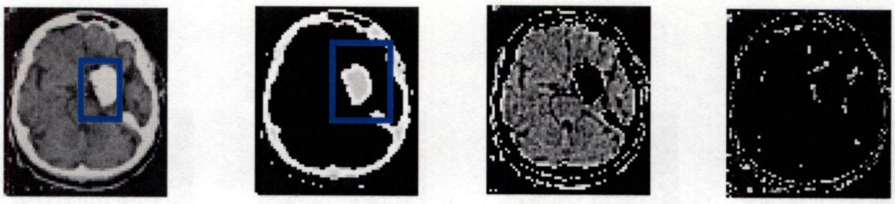
For example an Optical Character Recognition method may be good for graphs or charts found in biomedical educational area while a region based approach is much better for pathology and radiology images. A method based on classification of images using texture and color reported in Chapter 5 is applied for searching biomedical images. These methods are used to categorize biomedical images to a category. This method can be useful in diagnostic of diseases if the possibility of a color and texture is known and the images can be classified into category of diseases. The lesions based on selected texture features can be detected. For testing the effectiveness of the algorithm, collection of biomedical images from National Technical Information Services Springfield U.S.A., real MRI images obtained from PGI, Chandigarh have been used as Image Database. Also some medical images downloaded from the Internet have been used as dataset. Cancerous tumor may form in practically any organ. Malignant tumors are classified according to their site of origin as Carcinoma in epithelia, Sarcoma in connective tissues and muscles, osteo-sarcoma in bones, lymphoma or leukomia in haemopoitic glands, and melanoma that can originate in skin. Diagnostics of these tumors based on color and texture has been analyzed in the technique.

Some of the results obtained based on the experiments performed on biomedical image database and the regions identified on the basis of color and texture are presented in Figure 7.1 to Figure 7.4. The regions are extracted and stored in the medical image database using the algorithm. The query image having abnormalities can be submitted to the Neural Network (network architecture and parameters are assumed same as in Chapter 5) and training can be performed. The trained output can be compared with the images in the database for similarity factor. The similar images are said to have the same abnormality based on texture and color as described in Chapter 5. The results obtained on applying our algorithms have been consulted with Radiologist from PGI, Chandigarh and the diagnostics matches the medical reports.

As such the algorithm has been tested and implemented on about 100 medical images. However, only some of the results are being reported.

Calcified tumor occur due to calcification of bone. Carcinoid tumor may occur in any organ, the images given in Figure 7.1 show the affected regions and the successive images show the region classified on the basis of different texture and color. The marked boxes shows the affected regions extracted using our algorithm.

Calcified tumor



Carcinoid tumor

Figure 7.1 Classification of tumor based on texture and color

The diagnostics given by the doctor for Figure 7.2 is as follows, Brain MRI show a cavernous malformation located in the pons. Also it is apparent from the Figure the lesion has a different texture and color intensity as marked in the images in Figure 7.2, the regions are obtained on applying the proposed algorithm.

In Figure 7.3 the doctor diagnosis cavernous malformation from intraoperative photography from the colored image showing pontine haematoma. As can be seen in the image the proposed algorithm is able to extract the region having the haematoma (box marked) in the images shown in the figure.

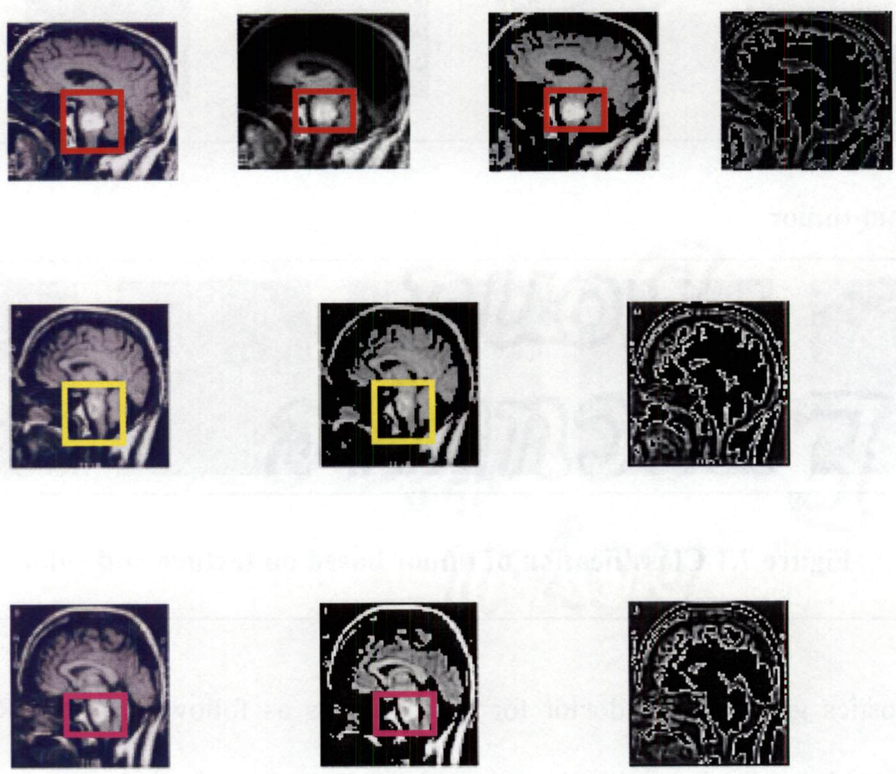


Figure 7.2 Regions Extracted from the Sagittal MRI images of brain

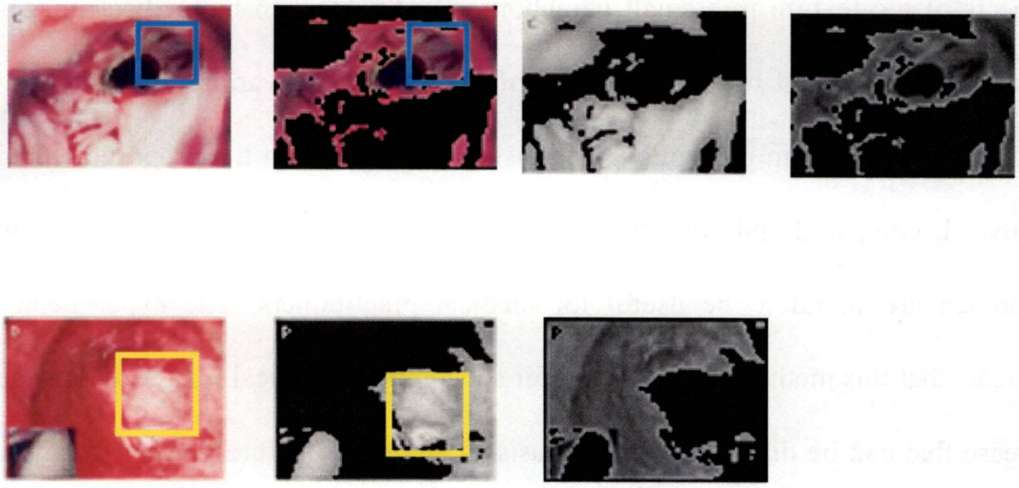


Figure 7.3 Regions extracted from Colored Medical Images

In Figure 7.4 the clinical finding diagnosis tumor from the axial MRI image showing ill defined hypodense lesion. The proposed algorithm is able to extract the region as shown in the images.

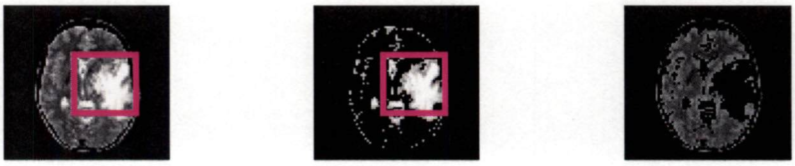


Figure 7.4 Regions extracted from axial MRI Images of brain

7.3 Discussion of Results and Conclusions

The study has shown some promise in use of texture and color for extraction of diagnostic information from medical images. The features used characterize different aspects of the texture in a small neighborhood of a pixel in biomedical images. These image features could be used to discriminate among the various tissue types that are inaccessible to human perception. The results obtained from the proposed technique are analyzed, compared and consulted with the Radiologist. The preliminary results of the approach are found to be useful for medical practitioners. The experimental results indicate that this method can be useful for screening biomedical images for any suspected disease that can be diagnosed on the basis of color and texture present in any region of the biomedical image. One major advantage of using neural networks in medical decision support system is that a huge effort of knowledge engineering into the domain knowledge can be saved, provided that sufficient amount of training cases are available. This will also help the medical community to predict new diseases in the various parts of the body. So far only two features of images such as texture and color are only worked upon, as shape also plays a vital role in biomedical image processing efforts are being made to include the shape feature also.

Chapter 8

Conclusions and Scope for Future Work

8.1 Conclusions

This thesis addresses some issues pertaining to Retrieval of Image Databases. Special cases of this nature arise in many important applications spanning the domain of agriculture, geology, computational biology and bio-informatics, among others. Efficient schemes based on low level image feature extraction are proposed for improving the performance of retrieval of image databases.

The major findings of our work can be summarized as follows:

1. A technique for texture feature extraction using haar wavelet approach has been implemented. For matching and retrieval of similar images clustering is employed for finding the similar regions. Another technique which employs Neural networks for extraction of texture feature and classification of texture into different sub classes is also proposed. Experimentally it is shown that the texture features can be extracted using neural network and texture classes can be learnt using the low level image

features. Brodatz texture images are used to test the algorithm. The results obtained of the proposed technique matches the results presented by Manjunath et. al. [72].

2. Color feature extraction using fuzzy interpretation of queries which helps the user to submit queries in linguistic terms is proposed, as in case of image query it is very difficult to submit query precisely in terms of color weightage.
3. Single image feature may lack in representing the image information. Therefore, we have attempted to combine image features such as texture and color for better image retrieval. The outcome is the method is able to extract regions on the basis of color and texture features. It has been found that the feature representation using the proposed method capture the user information needs better than the texture based feature only. The results obtained from the proposed technique are analyzed and compared with the techniques presented in [71]. The results of the approach appears to be comparable.
4. A new combined index which indexes the image database on the basis of dominant color region, and then the local shape feature turning angle is used to represent shape features. It filters the images based on dominant color while searching. Besides, it improves the database hierarchy for faster retrieval of images. The performance of the proposed method is compared with that of Fourier descriptor method and grid based method, it is found that the precision Vs recall of the proposed method is better.
5. A frame sliced signature file based parallel approach for fast image retrieval is proposed. We have shown that the variable bin allocation, which represents the color information of an image with the help of a compact binary signature, can be a better choice for parallel approach. The false drop probability with this approach is zero as

in case of Variable Bin Allocation only one bit is set. Speedup performance is also improved as compared to S-Tree parallel traversal approach relative to sequential processing. The results reported in [26] are based on the assumption that the main memory is sufficient enough to hold all the signatures whereas the proposed method does not have any such constraints.

6. The technique proposed for combined texture and color feature presented in Chapter 5, has been applied to Real MRI images obtained from PGI Chandigarh. Experimental results have been performed on colored and gray scale medical images. The results obtained are comparable with the diagnostics recommended by the medical practitioners.

8.2 Scope for Future Work

This study opens up number of avenues for future work. There are number of research issues which need to be addressed.

1. In our study, we have used two dimensional images, the work can be extended for 3D images.
2. We have used Neural Network and Fuzzy Logic approach for retrieval of images. The Soft Computing approach, which includes neural network, Fuzzy logic and genetic algorithms, can also be explored for further improving the retrieval performance.
3. Efforts should be made to implement the parallel frame slice signature file approach for distributed environment. So that fast retrieval of image databases may be possible on the Web.
4. Though we have made an initial attempt using the two image features color and texture for medical image diagnostics, efforts are required to integrate the shape

feature also for extracting abnormalities in medical images, so that more accurate diagnostics may be achieved. An important area where this technique can be helpful is in Telemedicine.

5. Studies should be made to integrate all image features such as color, texture, shape and spatial relations together for better retrieval results.
6. Efforts should be made for classification of images using combined strategies for better results. Bagging and boosting techniques can be explored.
7. Semantic based image retrieval is an area where more research efforts are needed. Extraction of semantic low level features is still an open problem.

References

[1] Aksoy.S. and Haralick.R.M., "Using Texture in Image Similarity and Retrieval," World Scientific, March 2000.

[2] Ankerst.M. and Seidl.T., "A Multistep Approach for Shape Similarity Search in Image Databases," IEEE Transactions on Knowledge and Data Engineering, vol.10, no. 6, November-December 1998.

[3] Ashley.J., Flickner.M., Hafner.J., Lee.D. and Niblack.W., "The Query By Image Content (QBIC) System," Proceedings of ACM SIGMOD International Conference 1995, CA, USA.

[4] Aslandogan.Y.P. and Yu.C.T., "Techniques and Systems for Image and Video Retrieval," IEEE Transactions on Knowledge and Data Engineering, vol.11, no. 1, January 1999.

[5] Bach.J.R., Paul.S. and Jain.R., "A Visual Information Management System for the Interactive Retrieval of Faces," IEEE Transactions on Knowledge and Data Engineering, vol.5, no. 4, August 1993.

[6] Beckmann.N. and Kreigel.H.P, "The R*-Tree : An Efficient and Robust Access Method for Points and Rectangles," Proceedings of ACM SIGMOD 1990.

[7] Berchtold.S., Keim.D.A. and Kriegel.H.P., "The X-Tree : An Indexing Structure for High Dimensional Data," Proceedings of 22^{nd} VLDB Conference, pp.28-39, Bombay, India 1996.

[8] Belongie.S. and Malik.J., "Finding Boundaries in Natural Images : A New Method using Point Descriptors and Area Completion," Fifth European Conference on Computer Vision, Freiburg, Germany, 1998.

[9] Bharadwaj.K.K. and Neerja, "Fuzzy Hierarchical Censored Production Rules System," International Journal of Intelligent Systems, John Wiley and Sons, NewYork, vol.11.no.1, pp.1-26, 1996.

[10] Bhatnagar.R., Horvitz.R., "A Hybrid System for Target Classification," Pattern Recognition Letters, 18(1997), pp. 1399-1403.

[11] Brunelli.R. and Mich.O., "Histogram Analysis for Image Retrieval," ITC-IRST Tech. Report-9812-03.

[12] Canny. J., "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8 (6), pp.679-698, 1986.

[13] Carson.C., Belongie.S., Greenspan.H. and Malik.J., "Region Based Image Querying," Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries, 1997.

[14] Cascia.M., Sethi.S. and Sclaroff.S., "Combining Textual and Visual Cues for Content Based Image Retrieval on the World Wide Web," Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries, June 1998.

[15] Chakrabarti.K. and Mehrotra.S., "The Hybrid Tree : An Index Structure for High Dimensional Feature Spaces," Fifteenth International Conference on Data Engineering, pp. 440-447, Australia, 1999.

[16] Chang.S.K., Yan.C.W., Dimitroff.C. and Arndt.T., "An Intelligent Image Database System," IEEE Transactions on Software Engineering, vol.14, pp. 681-688, May 1988.

[17] Chang.S.K., Shi.Q.Y. and Yan.C.W., "Iconic Indexing by 2-D Strings," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-9, no.3, pp. 413-428, May 1987.

[18] Chen.J.Y., Taskiran.C., Delp.E.J. and Bouman.C.A., "ViBE : A New Paradigm for Video Database Browsing and Search," Proceedings of Workshop on Content-Based Access of Image and Video Libraries (in conjunction with CVPR'98) Santa Barbara, CA, pp. 96-100, June 1998.

[19] Chitkara.V., "Color Based Image Retrieval Using Compact Binary Signatures," Technical Report, Department of Computer Science, University of Alberta, 2001.

[20] ChuChin.W.W., Cardensa.A.F. and Taira.R.K., "Knowledge Based Image Retrieval with Spatial and Temporal Constraints," IEEE Transactions on Knowledge and Data Engineering, vol.10, no. 6, November 1998.

[21] Ciaccia.P., Patella.M. and Zezula.P., "M-Tree : An Efficient Access Method for Similarity Search in Metric Spaces," Proceedings of the 23rd VLDB Conference, pp.426-435, Athens, Greece, 1997.

[22] Ciocca.G., Gagliardi.I. and Schettini.R., "Retrieving Color Images by Content," Proceedings of Image and Video Content Based Retrieval, 1998.

[23] Corridoni.J.M., Bimbo.A.D. and Vicario.E., "Painting Retrieval Based on Color Semantics," Image Databases and Multimedia Search, World Scientific Series on Software Engineering and Knowledge Engineering, vol.8, 1997.

[24] Cosman.P.C., Oehler.K.L., Riskin.E.A. and Gray.R.M., "Using Vector Quantization for Image Processing," Proceedings of IEEE, vol.81, pp. 1326-1341, September 1993.

[25] Das.M. and Riseman.E.M., "Feature Selection for Robust Color Image Retrieval," Multimedia Indexing and Retrieval Group, University of Massachusettes. MA.

[26] Davidson.A., Avnik.J. and Nascimento.M., "Parallel Traversal of Signature Trees for Fast CBIR," Department of Computer Science, University of Alberta, 2000.

[27] Deng.Y., Manjunath.B.S. and Kenney.C., "An Efficient Color Representation for Image Retrieval," IEEE Transactions on Image Processing, vol.10, no.1, January 2001.

[28] Deogun.J.S., V.V.Raghvan, and Sever.H., "On Feature Selection and Effective Classifiers," Journal of American Society for Information Science, vol. 49(4), pp. 423-434, April 1998.

[29] Deppisch.U., "S-tree: A Dynamic Balanced Signature Index for Office Retrieval," Proceedings of the 9th ACM SIGIR International Conference, pp. 77-87, 1986.

[30] Duda.R.O., Hart.P.E., and Stork, Pattern Classification, NewYork, Wiley 2000.

[31] Faloustsos.C., Barber.R., Flickner.M., Hafner.J., Niblack.W., Petkovic.D. and Equitz.W., "Efficient and Effective Querying by Image Content," Journal of Intelligent Information Systems, vol. 3, pp.231-262, 1994.

[32] Flickner.M., Sawhney.H., Niblack.W., Ashley.J., Huang.Q. and Dom.B., "Querying by Image and Video Content," IEEE Computer, vol.28, no.9, pp.23-32, September 1995.

[33] Furht.B., "Multimedia System : An Overview," IEEE Multimedia, vol.1, no.1, Spring 1994.

[34] Gevers.T. and Smeulders.A.W.M., "A Comparative Study of Several Color Models for Color Image Invariant Retrieval".

[35] Gonzalez.R and Woods.R.E, Digital Image Processing, Addison Wesley, 1993.

[36] Gupta.A. and Jain.R., "Visual Information Retrieval," Communications of the ACM, vol.40, pp.70-79, May 1997.

[37] Guttman.A., "R-trees : A Dynamic Index Structure for Spatial Searching," Proceedings of ACM SIGMOD Conference of Management of Data, pp. 47-57, 1984.

[38] Haley.G.M. and Manjunath.B.S., "Rotation Invariant Texture Classification using Modified Gabor Filters," Proceedings of IEEE International Conference on Image Processing, ICIP'95.

[39] Haykin.S., Neural Networks, Prentice Hall, 1999.

[40] Hsu.W., Chua.T.S. and Pung.H.K., "An Integrated Color-Spatial Approach to Content-based Image Retrieval," Proceedings of ACM Multimedia 95.

[41] Hsu.C.C, Chu.W.W. and Taira.R.K., "A Knowledge Based Approach for Retrieving Images by Content," IEEE Transactions on Knowledge and Data Engineering, 8(4): pp. 533-539, August 1996.

[42] Huang.J, Kumar.S, Mitra.M, Zhu.W.J and Zabih.R., “Image Indexing using Color Correlograms,” Proceedings of IEEE CVPR’97, San Juan, Puetro Rico 1997.

[43] Huang.J. and Zabih.R., “Combining Color and Spatial Information for Content Based Image Retrieval,” Proceedings of Second European Conference on Research and Advance Technology for Digital Libraries, 1998.

[44] Jacobs.C.E, Finkelstein.A. and Salesin.D.H., “Fast Multiresolution Image Querying,” Proceedings of ACM 1996.

[45] Jaimes.A and Chang.S.F, “Automatic Selection of Visual Features and Classifiers,” Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases VIII, vol. 3972, SanJose,CA, January 2000.

[46] Jain.A.K., Vailaya.A. and Xiong.W., “Query by Video Clip,” Multimedia Systems: Special Issue on Video Libraries, vol.7, pp.369-384, September 1999.

[47] Ray.S and Jiang.H., “Sequential and Parallel Algorithms for Partitioning Tree TaskGraphs on Shared Memory Architecture,” Proceedings of IEEE International Conference on Parallel Processing (ICPP), pp. 266-270, August 1994.

[48] Jose.J.M., Furner.J. and Harper.D.J., “Spatial Querying for Image Retrieval : A User Oriented Evaluation”.

[49] Joshi.R.C. and Tapaswi.S., “Efficient Image Retrieval Technique : An Indexing Approach,” Proceedings of International Conference on Multimedia Processing and Systems ICMPS-2000, I.I.T Chennai, pp.199-202, August 13-15, 2000.

[50] Joshi.R.C and Tapaswi.S., “Efficient Image Retrieval through Clustering and Neural Network Approach,” Proceedings of SCI-2001, pp. 304-308, Fifth World

Multiconference on Systemics, Cybernetics and Informatics, Orlando Florida USA, July 22-25,2001.

[51] Joshi.R.C. and Tapaswi.S., "Image Classification using Supervised Learning Approach," Proceedings of All India Seminar on Recent Trends in Communication Networks, Department of Electronics & Computer Engineering, I.I.T Roorkee, November, 7-8, 2001.

[52] Joshi.R.C. and Tapaswi.S., "Clustering of Large Image Databases : A New Approach," Proceedings of WDC-2001, Workshop on Distributed Computing, IEEE Computer Society, India & IEEE Gujarat Section, Department of Computer Science, Gujarat University, Ahmedabad, December 20-21,2001.

[53] Joshi.R.C. and Tapaswi.S., "Retrieval of Image Databases using Wavelet Based Decomposition for Semantic Label Clustering," Proceedings of ICWA-2002, International Conference on Wavelets and their Applications, School of Mathematics, Anna University, Chennai, January 6-8, 2002.

[54] Joshi.R.C. and Tapaswi.S., "Query Implementation Technique for Large Image Databases," Special Issue on Image Processing, Defence Science Journal, July 2002.

[55] Joshi.R.C. and Tapaswi.S., "Retrieval of Image Databases : A Neural Network Based Semantic Classifier Approach," Proceedings of International Conference on Computer Applications in Electrical Engineering CERA-2001, Department of Electrical Engineering, I.I.T. Roorkee, pp. 306-309, February 2002.

[56] Joshi.R.C. and Tapaswi.S., "Retrieval of Image Databases using Supervised Learning Approach," Proceedings of First European Conference on Color in Graphics, Imaging and Vision CGIV-2002, Poitiers, France, April 2-5,2002.

[57] Kasabov.N., Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering, M.I.T. Press London.

[58] Katayama.N. and Satoh.S., "The SR-Tree : An Index Structure for High Dimensional Nearest Neighbor Queries," Proceedings of the ACM SIGMOD Conference, Tucson, Arizona, pp.369-380, May 1997.

[59] Kittler.J., Hatef.M. and Duin.R.P.W., "Combining Classifiers," Proceedings of IEEE International Conference ICPR'96.

[60] Kohonen.T., Kangas.J., Laaksonen.J. and Torkkola.K., "LVQ_PAK: A Program Package for the Correct Application of Learning Vector Quantization Algorithms," Proceedings of International Joint Conference on Neural Networks, Baltimore, pp. 725-730, June 1992.

[61] Kumar.V. and Mullins.J., "An Integrated Data Structure for Database Systems," Journal of Computer Information Systems.

[62] Kurniawati.R., Jin.J. and Shepherd.J., "The SS+-Tree : An Improved Index Structure for Similarity Searches in a High Dimensional Feature Space," Proceedings of the SPIE Storage and Retrieval of Image and Video Databases, pp.110-120, SanJose, CA, February 1997.

[63] Lai.T.S. and Tait.J., "Using Global Color Features for General Photographic Image Indexing and Retrieval," Proceedings of SIGIR-98, Melbourne, Australia.

[64] Lee.T., Sheng.L. and Bozkaya.T., "Querying Multimedia Presentation Based on Content," IEEE Transactions on Knowledge and Data Engineering, vol.11, no. 8, May 1999.

[65] Lerner.B., Guterman.H., Aladjem.M. and Dinstein.H., "A Comparative Study of Neural Network based Feature Extraction Paradigms," Pattern Recognition Letters, 20 (1999), pp.7-14.

[66] Li.W.S., Candan.K.S, Hirata.K. and Hara.Y, "Facilitating Multimedia Database Exploration through Visual Interfaces and Perceptual Query Reformulations," Proceedings of the 23rd VLDB Conference, pp. 538-547, Athens, Greece 1997.

[67] Lin.K., Jagdish.H. and Faloutsos.C., "The TV-tree : An Index Structure for High Dimensional Data," VLDB Journal, pp. 517-542, 1994.

[68] Lorenz.A., Blum.M., Ermet.H. and T.Senge., "Comparison of Different Neuro-Fuzzy Classification Systems for the Detection of Prostate Cancer in Ultrasonic Images".

[69] Lu.G. and Sajjanhar.A., "Region Based Representation and Similarity Measure Suitable for Content Based Image Retrieval," Multimedia Systems, pp. 165-174, 1999.

[70] Ma.W.Y. and Manjunath.B.S., "NETRA : A Toolbox for Navigating Large Image Databases," Proceedings of IEEE International Conference on Image Processing, vol.1, pp. 568-571, Santa Barbara, CA, October 1997.

[71] Ma.W.Y. and Manjunath.B.S., "Texture Features and Learning Similarity," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 425-430, 1996.

[72] Manjunath.B.S. and Ma.W.Y., "Texture Features for Browsing and Retrieval of Image Data," IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8): pp.837-842, August 1996.

[73] Mehtre.B.M., Kakanhalli.M.S., Narasimhalu.A.D. and Man.G.C., "Color Matching for Image Retrieval," Pattern Recognition Letters, 16(1995), pp. 325-331.

[74] Mehrotra.S., Rui.Y., Ortega.M. and Huang.T., "Supporting Content Based Queries over Images in MARS".

[75] Miyahara.M., "Mathematical Transform of (r,g,b) Color Data to Munsell (h,s,v) Color Data," Proceedings of SPIE Visual Communications and Image Processing, vol. 1001, 1988.

[76] Nascimento.M. and Chitkara.V., "Color Based Image Retrieval using Signature Trees," Tech Report 02-01, Department of Computer Science, University of Alberta, 2000.

[77] Ogle.V.E. and Stonebraker.M., "Chabot : Retrieval from a Relational Database of Images," IEEE Computer, vol.28, no.9, pp.40-48, September 1995.

[78] Oomoto.E. and Tanaka.K., "OVID : Design and Implementaion of a Video-Object Database System," IEEE Transactions on Knowledge and Data Engineering, vol.5, no. 4, August 1993.

[79] Ortega.M., Rui.Y., Chakrabarti.K., Porkaew.K., Mehrotra.S. and Huang.T.S., "Supporting Ranked Boolean Similarity Queries in MARS," IEEE Transactions on Knowledge and Data engineering, vol.10, no. 6, November 1998.

[80] Papadias.D., Mamoulis.N. and Meretakis.D., "Image Similarity Retrieval by Spatial Constraints," Department of Computer Science, Hongkong.

[81] Pass.G., and Zabih.R. "Histogram Refinement for Color Based Image Retrieval," ACM Journal of Multimedia Systems 7(3) : pp. 234-240, May 1999.

[82] Pass.G., Zabih.R. and Miller.J. "Comparing Images Using Color Coherence Vectors," Proceedings of ACM Multimedia pp. 65-73, Boston, Massachusetts, USA 1996.

[83] Pekalska.E. and Duin.R.P.W., "Classifiers for Dissimilarity Based Pattern Recognition," Proceedings of International Conference on Pattern Recognition, ICPR'2000.

[84] Pei. Soo-Chang and Hui-Lirng Shiue, "Indexing and Retrieval of Color Image Database," Proceedings of Conference on Computer Vision, Graphics and Image Processing, Taipei, pp.196-199, 1998.

[85] Pentland.A., Picard.R.W. and Sclaroff.S., "Photobook : Content Based Manipulation of Image Databases," Proceedings of SPIE Storage and Retrieval for Image and Video Databases II, vol.2185, pp. 34-47, February 1994.

[86] Peterson.M.E. and Pelikan.E., "Detection of Bone Tumors in Radiographic Images Using Neural Networks," Pattern Analysis and Applications,1999.

[87] Poynton.C, "Frequently Asked Questions about Color," 1997. URL http://Home.InfoRamp.Net/~poynton/ColorFAQ.html.

[88] Prasad.B.G. and Gupta.S.K., "Color and Shape Index for Region Based Image Retrieval, " Springer Verlag Berlin, Heidelberg, pp.716-725, 2001.

[89] Rabitti.F. and Zezula.P., "A Dynamic Signature Technique for Multimedia Databases," ACM Transactions 1990.

[90] Ramakrishnan.R., Database Management Systems, McGrawHill, 1997.

[91] Reinus.W.R., Wilson.A.J., Kalman.B. and Kwasny.S., "Diagnosis of Focal Bone Lesions using Neural Networks," Investigative Radiology 1994; 29(6): pp. 606-611.

[92] Ripley.B.D., "Pattern Recognition via Neural Networks".

[93] Robinson.J.T., "The KDB-Tree : A Search Structure for Large Multidimensional Dynamic Indexes," Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 10-18, April 1981.

[94] Roussopoulos.N., Faloutsos.C. and Sellis.T., "An Efficient Pictorial Database System for PSQL," IEEE Transactions on Software Engineering, vol.14, pp.681-688, May 1988.

[95] Roussopoulos.N. and Kelly.S., "Nearest Neighbor Queries," Proceedings of ACM SIGMOD, 1995.

[96] Rowley.H.A., Baluja.S. and Kanade.T., "Neural Network Based Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.20, pp. 23-38, January 1998.

[97] Rubner.Y. and Tomasi.C., "Texture Based Image Retrieval without Segmentation".

[98] Rui.Y., Huang.T.S. and Chang.S.F., "Image Retrieval : Current Techniques, Promising Directions, and Open Issues," Journal of Visual Communication and Image Representation, vol. 10, pp. 39-62, 1999.

[99] Rui.Y., Huang.T.S. and Mehrotra.S., “Content Based Image Retrieval with Relevance Feedback in MARS,” Proceedings of IEEE International Conference on Image Processing, 1997.

[100] Rui.Y., Huang.T.S. and Mehrotra.S., “Relevance Feedback : A Power Tool for Interactive Content-Based Image Retrieval,” IEEE Transactions on Circuit and Video Technology, 1997.

[101] Rui.Y., She.A.C. and Huang.T.S., “Modified Fourier Descriptors for Shape Representation – A Practical Approach,” Proceedings of First International Workshop on Image Databases and Multimedia Search, Amsterdam, Netherlands, 1996.

[102] Rushing.J.A., Ranganath.H.S., Hinke.T.H. and Graves.S.J., “Using Association Rules as Texture Features,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no.8, August 2001.

[103] Salton.G. and McGill.M.J., Introduction to Modern Information Retrieval, McGraw-Hill Book Company, 1983.

[104] Santini.S. and Jain.R., “Similarity Queries in Image Databases,” Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition CVPR’96, 1996.

[105] Santini.S., “The Semantic Foundations of Image Databases,” Praja, Inc.

[106] Sellis.T., Roussopoulus.N. and Faloutsos.C., “The R+-Tree : A Dynamic Index for Multi-Dimensional Objects,” Proceedings of the ACM SIGMOD, pp. 154-165, Seattle, Washington, USA, 1998.

[107] Servetto.S., Rui.Y., Ramchandran.K. and Huang.T.S., "A Region Based Representation of Images in MARS," Kluwer Academic Publishers.

[108] Shaft.U. and Ramakrishnan.R., "Data Modeling and Querying in the PIQ Image Database Management System," Bulletin of IEEE Computer Society Technical Committee on Data Engineering, pp. 28-36, 1996.

[109] Smith.J. and Chang.S.F., "Local Color and Texture Extraction and Spatial Query," Proceedings of IEEE International Conference on Image Processing, 1996.

[110] Smith.J.R. and Chang.S.F, "Single Color Extraction and Image Query," Proceedings of International Conference on Image Processing, Washington D.C., 1995.

[111] Smith.J.R. and Chang.S.F, "Tools and Techniques for Color Image Retrieval," Proceedings of SPIE Storage and Retrieval for Image and Video Databases IV, vol.2670, pp. 426 -437, 1996.

[112] Smith.J.R. and Chang.S.F., "VisualSEEK : A Fully Automated Content Based Image Query System," Proceedings of ACM Multimedia, Boston, MA, pp. 87-98, November 1996.

[113] Smith.J.R. and Chang.S.F., "Automated Binary Texture Feature Sets for Image Retrieval," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, May 1996.

[114] Srihari.R.K, "Automatic Indexing and Content Based Retrieval of Captioned Images," IEEE Computer, vol.28, no.9, pp.49-56, September 1995.

[115] Stollnitz.E.J, DeRose.T.D. and Salesin.D.H., "Wavelets for Computer Graphics : Theory and Applications," Morgan Kaufman Publishers, 1996.

[116] Stricker.M. and Dimai.A., "Color Indexing with Weak Spatial Constraints," Proceedings of SPIE Storage and Retrieval for Large Image and Video Databases, pp.29-40, 1996.

[117] Subrahmanian.V.S., "Principles of Multimedia Database Systems," Morgan Kaufman Publishers, Inc. San Francisco, California, 1998.

[118] Swain.M. and Ballard.D., "Color Indexing," International Journal of Computer Vision, 7(1) : pp. 11-32, 1991.

[119] Swets.D.L. and Weng.J.J., "Hierarchical Discriminant Analysis for Image Retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, May 1999.

[120] Taycher.L., Casco.M. and Sclaroff.S., "Image Digestion and Relevance Feedback in the Image Rover WWW Search Engine," BU-CS TR97-014, SanDiego, December 1997.

[121] Tousiduo.E., Nanopoulos.A. and Manolopoulos.Y., "Improved methods for Signature-Tree Construction," The Computer Journal, 43(4): pp. 301-314, 2000.

[122] Vertan.C. and Boujemaa.N., "Color Texture Classification by Normalized Color Space Representation".

[123] Wan, Xia and C.C Jay Kuo, "Color Distribution Analysis and Quantization for Image Retrieval," Proceedings of SPIE storage and Retrieval for Image and Video Database IV, vol. SPIE 2670, pp.9-16, February 1996.

[124] Wang.J., Yang. Wen-Jann and Acharya Raj, "Color Clustering Techniques for Color-Content-Based Image Retrieval from Image Databases," Proceeding of IEEE International Conference on Multimedia Computing and Systems (ICMCS), pp. 442-449, June 3-6, 1997.

[125] Wasserman.P., Neural Computing Theory and Practice, Van Nostrand Reinhold, NewYork, NY, USA,1989.

[126] Yang.Q. and Xubin.H., " Characterizing the Home Pages," Proceedings of Second International Conference on Internet Computing, pp. 976-982, June 2001.

[127] Yoshitaka.A. and Ichikawa.T., "A Survey on Content Based Retrieval for Multimedia Databases," IEEE Transactions on Knowledge and Data Engineering, vol.11, no.1. January 1999.

[128] Zezula.P. and Rabitti.F., "Dynamic Partitioning of Signature Files," ACM Transactions on Information Systems, vol.9., no.4, October 1991.

[129] Zheng.Lin, Faloutsos.C., "Frame Slice Signature Files," IEEE Transactions on Knowledge and Data Engineering, vol.4., no.3., June 1992.

More Details of some of the References

(As available from net)

[1] Aksoy.S. and Haralick.R.M., "Using Texture in Image Similarity and Retrieval," World Scientific, March 2000. http://citeseer/nj.nec.com/aksoy00using.html.

[25] Das.M. and Riseman.E.M., "Feature Selection for Robust Color Image Retrieval," Multimedia Indexing and Retrieval Group, University of Massachusettes. MA. http://vis-www.cs.umass.edu/vislib/Papers/riseman/files.html.

[34] Gevers.T. and Smeulders.A.W.M., "A Comparative Study of Several Color Models for Color Image Invariant Retrieval," Proceedings of First International Workshop IDB-MMS-96, Amsterdam Netherlands, pp.17-26, August 1996.

[48] Jose.J.M., Furner.J. and Harper.D.J., "Spatial Querying for Image Retrieval : A User Oriented Evaluation," Proceedings of ACM SIGIR Conference of Image Retrieval, SIGIR-98.

[80] Papadias.D., Mamoulis.N. and Meretakis.D., "Image Similarity Retrieval by Spatial Constraints," Proceedings of Seventh Conference on Information and Knowledge Management , Bethesda Maryland, U.S.A., 1998.

[92] Ripley.B.D., "Pattern Recognition via Neural Networks," Cambridge University Press, January 1996.

[97] Rubner.Y. and Tomasi.C., "Texture Based Image Retrieval without Segmentation," Proceedings of ICCV-99, pp.1018-1024, 1999.

[105] Santini.S., "The Semantic Foundations of Image Databases," Praja, Inc. http://citeseer/nj.nec.com/correct/445276.

[122] Vertan.C. and Boujemaa.N., "Color Texture Classification by Normalized Color Space Representation," Proceedings of ICPR 2000, vol.III, pp.584-587. 2000.