CONTENT BASED IMAGE RETRIEVAL SYSTEM –USING COLOR, SHAPE & TEXTURE

By

SWATI JAIN

(05MCE023)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY OF SCIENCE & TECHNOLOGY

AHMEDABAD 382481

MAY 2007

CONTENT BASED IMAGE RETRIEVAL SYSTEM –USING COLOR, SHAPE & TEXTURE

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Engineering

By

Swati Jain
(05MCE023)

Guide
Dr. S.N. Pradhan

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY OF SCIENCE & TECHNOLOGY

Ahmedabad 382481

May 2007

This is to certify that Dissertation entitled

Content Based Image Retrieval System –using Color, Texture & Shape

Submitted by

Swati Jain

has been accepted toward fulfillment of the requirement

for the degree of

Master of Technology in Computer Science & Engineering

Dr. S. N. Pradhan
Professor

Prof. D. J. Patel
Head of The Department

Prof. A. B. Patel
Director, Institute of Technology

CERTIFICATE

This is to certify that the Major Project entitled "Content Based Image Retrieval Syatem–using Color, Texture & Shape submitted by Ms. Swati Jain (05MCE023), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. S. N. Pradhan
Guide,
Department of Computer Science & Engineering,
Institute of Technology,
Nirma University,
Ahmedabad.

ACKNOWLEDGEMENT

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr**. **S.N Pradhan**, Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout the Major project. I am heartily thankful to him for his precious time, suggestions and sorting out the difficulties of my topic that helped me a lot during this study. I am thankful to **Dr. Gopal Krishnan**, Scientist, ISRO, Ahmedabad whose exposure to real time problems in the field of image processing helped me in sorting the implementation complexities.

I also want to extend my thanks to **Prof. Jaladhi Joshi,** Asst. Professor and **Prof. Madhuri Bhavsar** Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad, who helped me when ever I got stuck and needed help.

I would like to give my special thanks to **Prof**. **D.J Patel**, Head, Department of Computer Engineering, Institute of Technology, Nirma University, Ahmedabad for his encouragement and motivation throughout the project. I am also thankful to **Prof. A. B. Patel**, Director, Institute of Technology for his kind support in all respect during my study.

I am thankful to all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work. I also took help of my classmates to give me the feedback on the outputs. Their valuable suggestions are also to be thanked.

Very important, the blessings of God and constant support of my family members made the way for completion of my project. I am very much grateful to them.

Swati Jain
Roll No. 05MCE023

ABSTRACT

Rapid increase in the usage of color image has led to the need of an image retrieval system for them. The world is dominated by visual information and a tremendous amount of such information is being added every day. It would be impossible to cope with this explosion of visual data, unless they are organized in such a manner that we can retrieve them efficiently and effectively. The main problem in organizing and managing such visual data is indexing, the assignment of a synthetic descriptor which facilitates its retrieval. There are various mathematical models that describe the image information and content. The Work presents the image retrieval based on color, shape and texture. Two different indexed data bases are maintained for the mentioned features. Global color dominance in the image is used as the index key in the color database, and local color information is retained by quantizing images in color and pixel space to optimize in memory need and performance. Texture information is obtained using co-occurrence matrix of the color images converted to gray. The properties like correlation, energy, homogeneity and contrast are good parameters to measure textural property; these are evaluated at proper offset and angle. Shape in the image is identified using the contours in the image. Only the geometrical figures like circle, ellipse, quadrilateral, triangle, pentagon etc. can be identified. While retrieving images based on features like texture and shape one thing missing is, color. Color is inevitable feature that dominates the human perception most. To a human observer, images with similar color scheme and reasonably similar in other features is voted higher in similarity list, than images measuring better in other feature descriptors. Making this as the base, the work presented here uses all the three features independently and also grouping color with texture and shape. Results have shown that they match better with the human perception. MATLAB 7 Is the tool used to implement image processing algorithms and postgre SQL 8.1 for windows is the DBMS used for managing the database of image features.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 GENERAL

In today's world digital images and videos have become more effective ways of expression / communication and information capturing. They are playing increasingly important role in telecommunication and every day's life in modern information society. Hence, the past few years witnessed a proliferation of content-based image retrieval techniques. Images are typically characterized by intrinsic attributes of image such as color, texture, and shape. Users should be able to explore images in a database by visual similarities. Salient structures of images are revealed through visualization models derived from features extracted from images. Visualizations are generated from three feature classes: color, texture, shape and its combination like texture + color or shape +color.

Content-based image retrieval has been a highly potential area for research in the computer vision community. A number of image retrieval systems have been developed over the last few years, notably, IBM's QBIC [1], PhotoBook [2], ImageRover [3], and Webseek [4]. It is important to understand what does retrieving relevant images entail. If users are provided with a spatial user interface in which content similarity between images can be intuitively conveyed by their spatial proximity, then such interfaces may help users to benefit more from a given image database. There has been a steady increase in the interest in this type of layout and visualization techniques, which tend to place similar objects near to each other and separate dissimilar objects far apart in the visualization space. Structuring and visualizing digital images, based on their content similarities, however, is not as mature as its text-based counterpart. Currently, many content based image retrieval techniques have been developed to incorporate higher-level feature extraction capabilities, but a lot of work remains to be done. Ultimately, feature-extraction techniques, combined with other techniques, are expected to narrow down the gap between relatively primitive features extracted from images and high-level, semantically-rich perceptions by humans so that users will be able to find the right images more easily and intuitively. Visualization often needs to find an optimal way to arrange various displays so that design patterns and trends will become apparent. Ideally, images of similar layouts, spatial properties, or overall shapes should be closely grouped together. Users should be able to explore and compare images within such structures.

1.1.1 ISSUES IN CONTENT-BASED IMAGE RETRIEVAL (CBIR)

The key issue in CBIR is how to match two images according to computationally extracted features. Typically, the content of an image can be characterized by a variety of visual properties known as features. It is common to compare images by color, texture, and shape, although these entail different levels of computational complexity. Color histograms are much easier to compute than a shape-oriented feature extraction. Most content-based image retrieval techniques fall into two categories:

- Manual approach
- Computational approach

In manual approaches**,** a human expert may identify and annotate the essence of an image for storage and retrieval. For example, radiologists often work on medical images marked and filed manually with a high degree of accuracy and reliability. In manual approach, the image contents are modeled as a set of features also known as meta data, these are extracted manually or semi-automatically and managed within the framework of conventional database management system. The short comings of such techniques are, purely textual descriptions of visual documents are not feasible because

- They cannot be automated as the problem of image understandingis complex and difficult.
- They require tremendous efforts of uniform manual work.
- There are things which cannot be described by text but can be described by an image.

Computational approaches, on the other hand, typically rely on feature-extraction and pattern-recognition algorithms to match two images. Feature-extraction algorithms commonly match images according to the attributes, also known as features.

- Color
- Texture
- Shape

A robust CBIR technique should support a combination of these features Ideally, users should be able to use high-level and semantically-rich image query classes, such as human facial expressions, in their image retrieval. However, the reliability of today's feature-extraction techniques has yet to reach such a level of satisfaction. This is partially why simpler, and relatively low-level feature-extraction techniques are still being widely used and continuously developed.

An image retrieval system should be automatic and efficient to give an acceptable response time. The use of low-level features in the feature-based image retrieval systems makes the approach automatic but not necessarily efficient. The use of "real" distance in the retrieval process can be computationally very expensive. For example, in the case of retrieval by color, accounting for the effect of color correlations is time consuming. For this, most of the processing that can be done *a priori,* should be done off-line and as little computation as possible should be done on-line. The CBIR system should also be generic enough to be transparent when different low-level features are used with almost no or very little changes.

1.2 OBJECTIVE OF STUDY

The main motivating factors for selection of this topic for dissertation are;

- Content-based image retrieval has been a highly potential area of research in the computer vision community.
- No search engine in internet has content based image retrieval. Searching for the images in most commonly used search engine like *Google,* the search is text based which retrieves images based on name of the image.

As all the images with name car is retrieved given a query car irrespective of the content.

- There is big gap in meeting the requirement for a efficient and effective system which can used for various critical applications like
  - Space - Organization like ISRO deals with large amount of image data and refers it frequently for analysis. These organizations manage their big image database by using the proper nomenclature signifying the data and content of the image. But if the context of search changes then information in database can not be clustered based on desired context.
  - Medical field - Radiology departments of hospitals having medical images in big numbers which are varying from sonography, MRI to CT scans needs an image retrieval system to cluster the image and the relevant image for analysis.

The work initiated here is image retrieval system based on color, texture and shape individually. But this was realized that the one feature alone can not characterize the image, and features like color is inevitable property. If combined with texture and shape matches with human perception better. Considering the fact that an IR system should be automatic and efficient to give an acceptable response time with relevant images as output. The use of low-level features in the feature-based image retrieval systems makes the approach automatic but not necessarily efficient. The use of "real" distance in the retrieval process can be computationally very expensive. For example, in the case of retrieval by color, accounting for the effect of color correlations is time consuming. For this, most of the processing that can be done *a priori,* should be done off-line and as little computation as possible should be done on-line. The IR system should also be generic enough to be transparent when different low-level features are used with almost no or very little changes.

The system uses the generic and low level features off line and stores in the database, hence reducing the response time. Combining the generic features for the image retrieval is exploited to cope with the human perception better.

1.3 SCOPE OF WORK

The *computational* approach of image retrieval can be classified based on the dependence of retrieval process on how generic the approach is and the degree of automation. In our context the features are based on color, shape and texture it can be advanced as object, foreground and background or even as specific as Face. There are image retrieval system that are web based, some retrieval systems extracts the image features of the database online if they don't have predefined fixed image database. Also there are systems that have number of query options and out of various query options the user can choose any one. A number of image retrieval systems have been developed over the last few years, notably, IBM's QBIC[1], PhotoBook[2], ImageRover[3], and Webseek[4]. The type of image feature used is mostly color or texture some systems also uses shapes.

The work of dissertation is limited to extraction of images based on color shape and texture independently. Also the combination of these generic properties helps in improving the results to a great extend. The image retrieval system based on color and texture works for all time of images and textures, but it for shape based only the geometric figures can be identified.

1.4 ORGANIZATION OF THE WORK

The work of content base image retrieval is presented in eight chapters. Chapter three gives a bird's eye view to the system and explains how the overall system works. The general computational model and flow of the system is explained in the chapter.

Chapter 2 talks about the existing systems and the literature survey done. It gives a brief description of the related work. The scope of the work defines that the system can retrieve images based on color texture and shape. Chapter four gives an insight to the methodology used in color feature extraction, done by quantization. Chapter five is the introduction to the texture and various models to represent the texture feature. It also contains implementation details of the texture feature extraction. The importance of color feature and the improvement in the results by combining the color feature with the texture feature is represented in chapter six. Retrieval based on shape is explained in chapter seven. The way to get the area of interest and finding the signature to identify the shape is explained in the chapter. Finally chapter eight represents the results obtained on various retrieval options.

2. Literature Survey

2.1 GENERAL

A number of image retrieval systems have been developed over the last few years, notably, IBM's QBIC[3], PhotoBook[4], ImageRover[5], and Webseek[6]. IBM-QBlC lets users find pictorial information in large image and video databases based on color,shape, texture, and sketches. QBIC technology is part of several IBM products. SIMPLIcity (Semantics sensitive Integrated Matching for Picture LIbraries), an image retrieval system, which uses semantics classification methods, a wavelet-based approach for feature extraction, and integrated region matching based upon image segmentation. Webseek is the Content-Based Image and Video Search and Catalog Tool for the Web

2.2 TEXTURE FEATURES

A lot of work has been published since late 1970s on the textural properties of images and many mathematical models have been presented to represent the textural features of the image. Out of so many one or the other fit into all the texture type. The work done by Haralick [3] presents how to apply the statistical technique to the structural primitive. Abbedeni[8] in his work presents auto regressive model, a purely statistical model, and an empirical perceptual model based on perceptual features. Amadasun and Robert king [5] presented a model known as NGTDM(Neighborhood Gray Tone Difference Matrix) and mathematical interpretation of various textural properties such as coarseness, busyness etc. extracted from NGTDM. Co-occurrence matrix is one of the widely used models for textural properties which is also used in the presented work. The limitation of most of the textural models are for gray scale images and not for color images.

2.3 COLOR FEATURES

The idea as suggested in [1] [2] is quantization in color space and reducing the resolution of the image. The authors have experimentally found that dividing the RGB cube into 16 equal parts and then taking center of each of the small cubical as the R G and B values. Now the quantized image consists of only 16 colors instead of 256x256x256 colors. The input image of size MxN is then divided into grid of 3 pixels by 3 pixels each. The most dominant color in this 3x3 block becomes the representative color in the color feature. And hence the color feature is now a matrix of one third size in both the dimensions. The database of color image consists of image name and index key for each color feature. Index key is first five most dominant color in the database which is used to sort the image in order of there color content and hence make sorting fast

2.4 SHAPE BASE

This shape distribution introduces a method for finding distinctive features of a shape that are useful for determining shape similarity. Although global shape descriptors have been developed to facilitate retrieval, they fail when local shape properties are the distinctive features of a class. Alternatively, local shape descriptors can be generated over the surface of shapes, but then storage and search of the descriptors becomes unnecessarily expensive, as perhaps only a few descriptors are sufficient to distinguish classes. The challenge is to select local descriptors from a query shape that are most distinctive for retrieval.

In this writers have proposed and analyzed a method for computing shape signatures for arbitrary (possibly degenerate) 3D polygonal models. The key idea is to represent the signature of an object as a *shape distribution* sampled from a *shape function* measuring global geometric properties of an object. The primary motivation for this approach is to reduce the shape matching problem to the comparison of probability distributions, which is simpler than traditional shape matching methods that require pose registration, feature correspondence, or model fitting. They find that the dissimilarities between sampled distributions of simple shape functions (e.g., the distance between two random points on a surface) provide a robust method for discriminating between classes of objects (e.g., cars versus airplanes) in a moderately sized database, despite the presence of arbitrary translations, rotations, scales, mirrors, tessellations, simplifications, and model degeneracy. They can be evaluated quickly, and thus the proposed method could be applied as a pre-classifier in a complete shape-based retrieval or analysis system concerned with finding similar whole objects.

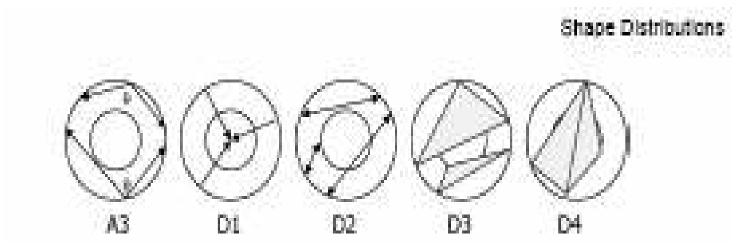


Figure 2.1 . Shape Distribution

In logo database using Fourier descriptor[14] is a system that enables the pictorial specification of queries. The queries are comprised of rectangle, polygon, ellipse, and B-spline shapes. The queries specify which shapes should appear in the target image as well as spatial constraints on the distance between them and their relative position. The retrieval process makes use of an abstraction of the contour of the shape which is invariant against translation, scale, rotation, and starting point that is based on the use of Fourier descriptors. These abstractions are used in a system to locate logos in an image database.

Shape Thesaurus[16]: a system which combines techniques from Computer Vision, such as feature extraction, with a thesaurus for objects / shapes. A thesaurus might alleviate the retrieval process by helping the retrieval system identify images annotated by keywords related to the terms specified in the query. The main purpose of a thesaurus is to give a standardized system of reference, for indexing and searching, to assist the user with locating the correct terms for query formulation and to provide classification hierarchies that allow the broadening and narrowing of the terms given by the user in the current query request. It is my belief that this principle might be translated into a thesaurus for shapes, thus linking shapes that are different in appearance but similar in semantic content together. It is hoped that this might result in more meaningful retrieval results than pure CBIR.

The following definition of a Shape Thesaurus is proposed:

(1) A precompiled list of t shapes representing important visual objects in a given domain of knowledge
(2) statistical descriptors describing these shapes
(3) a textual / semantic description of these shapes
(4) for each shape, a set of related shapes.

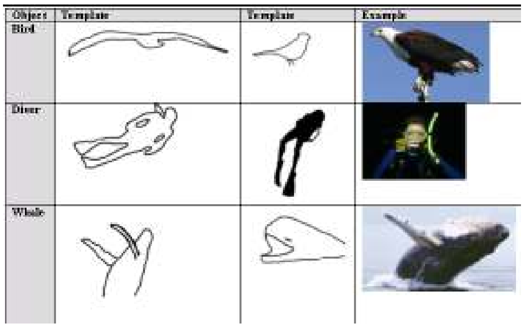


Figure2.2 : Shape Thesaurus

There are also so many approaches presented to features characterize the global shape feature of a 3Dmodel. Examples of these features are the statistical moments of the boundary or the volume of the model, volume-to-surface ratio, or the Fourier transform of the volume or the boundary of the shape.

Zhang and Chen [17] describe methods to compute global features such as volume, area, statistical moments, and Fourier transform coefficients efficiently.

Corney et al. [18] introduce convex-hull based indices like hull crumpliness (the ratio of the object surface area and the surface area of its convex hull), hull packing (the percentage of the convex hull volume not occupied by the object), and hull compactness (the ratio of the cubed surface area of the hull and the squared volume of the convex hull). Global feature methods are able to support user feedback as illustrated by the following research.

Zhang and Chen [19] applied features such as volume-surface ratio, moment invariants and Fourier transform coefficients for 3D shape retrieval. They improve the retrieval performance by an active learning phase in which a human annotator assigns attributes such as airplane, car, body, and so on to a number of sample models.

Elad et al. [20] use a moments based classifier and a weighted Euclidean distance measure. Their method supports iterative and interactive database searching where the user can improve the weights of the distance measure by marking relevant search results. The concept of global feature based similarity has been refined recently by comparing distributions of global features instead of the global features directly.

Osada et al. [21] introduce and compare shape distributions, which measure properties based on distance, angle, area and volume measurements between random surface points. They evaluate the similarity between the objects using a pseudo-metric that measures distances between distributions. In their experiments the D2 shape distribution measuring distances between random surface points is most effective.

Ohbuchi et al. [22] investigate shape histograms that are discretely parameterized along the principal axes of inertia of the model. The shape descriptor consists of three shape histograms: (1) the moment of inertia about the axis, (2) the average distance from the surface to the axis, and (3) the variance of the distance from the surface to the axis. Their experiments show that the axis-parameterized shape features work only well for shapes having some form of rotational symmetry.

Ip et al. [23] investigate the application of shape distributions in the context of CAD and solid modeling. Vrani´c et al. [24] describe a surface by associating to each ray from the origin, the value equal to the distance to the last point of intersection of the model with the ray and compute spherical harmonics for this spherical extent function. Spherical harmonics form a Fourier basis on a sphere much like the familiar sine and cosine do on a line or a circle. Their method requires pose normalization to provide rotational invariance

3.1 DEFINITION

To develop a Content Based Image Retrieval System for retrieval of imagery from a collection of images by means of internal-feature measures of the information content of an image. This is achieved by extracting features of an image based on its pixel values (color), texture, shape and by defining a rule for comparing images.

The system can be exactly defined as:-

- Develop a new and more effective method for storage and retrieval from databases of realistic images, based on the color, texture and shape features of the images.
- Design an interface for retrieval of the images based on texture, shape and color or in combination of color with the other two.
- Providing options for the queries, the user can select from. The query type can be query by example, query by user sketch etc.

3.2 GENERAL COMPUTATIONAL MODEL

Proposed Content Based Image Retrieval System provide techniques to re-encode contents of images and to measure the similarity of two images based on the coding, in order to index and retrieve images which are semantically related to a visual information request, without fully understanding the semantics carried by these images.

The Content Based Image Retrieval System can be divided into the following sub processes:

1. **Feature extraction,** which involves the creation of good mechanisms to code global and local features such as color, texture and shape. The extracted features should be semantically related to the contents of images and at the same time should be economic in terms of time taken and space.

2. **Defining Similarity, Indexing and Retrieving:** In accordance with the various features, a relevant indexing technique is to be identified to minimize the time required for the image retrieval. Similarity measure is another important decision to be taken which gives the appropriate measure of the semantic distance between the two images corresponding to the feature.

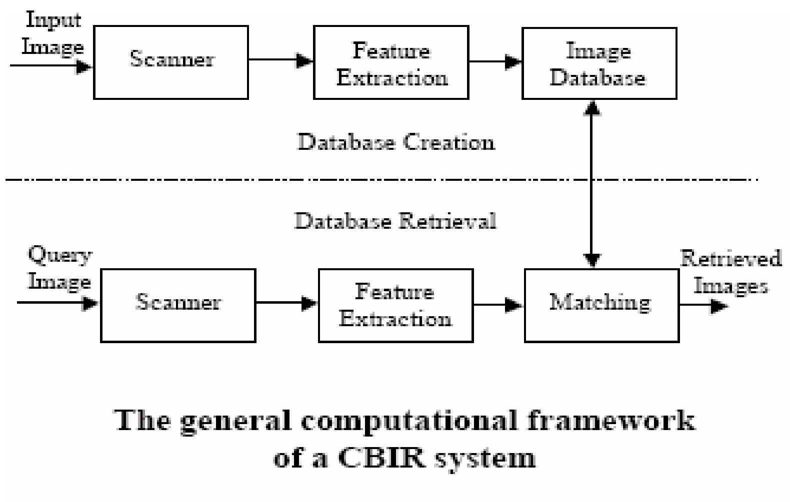


Figure 3.1 Computational Framework

The process of database creation is an offline process, in which the images and the extracted features are added in the database as metadata of image database. When a query is made by presenting an example image, the features are evaluated for the query image which is then compared with the database metadata. The results are sorted, values above the threshold are rejected and rest displayed in the order of similarity with the query image.

3.3 FLOW CHART FOR DATABASE CREATION

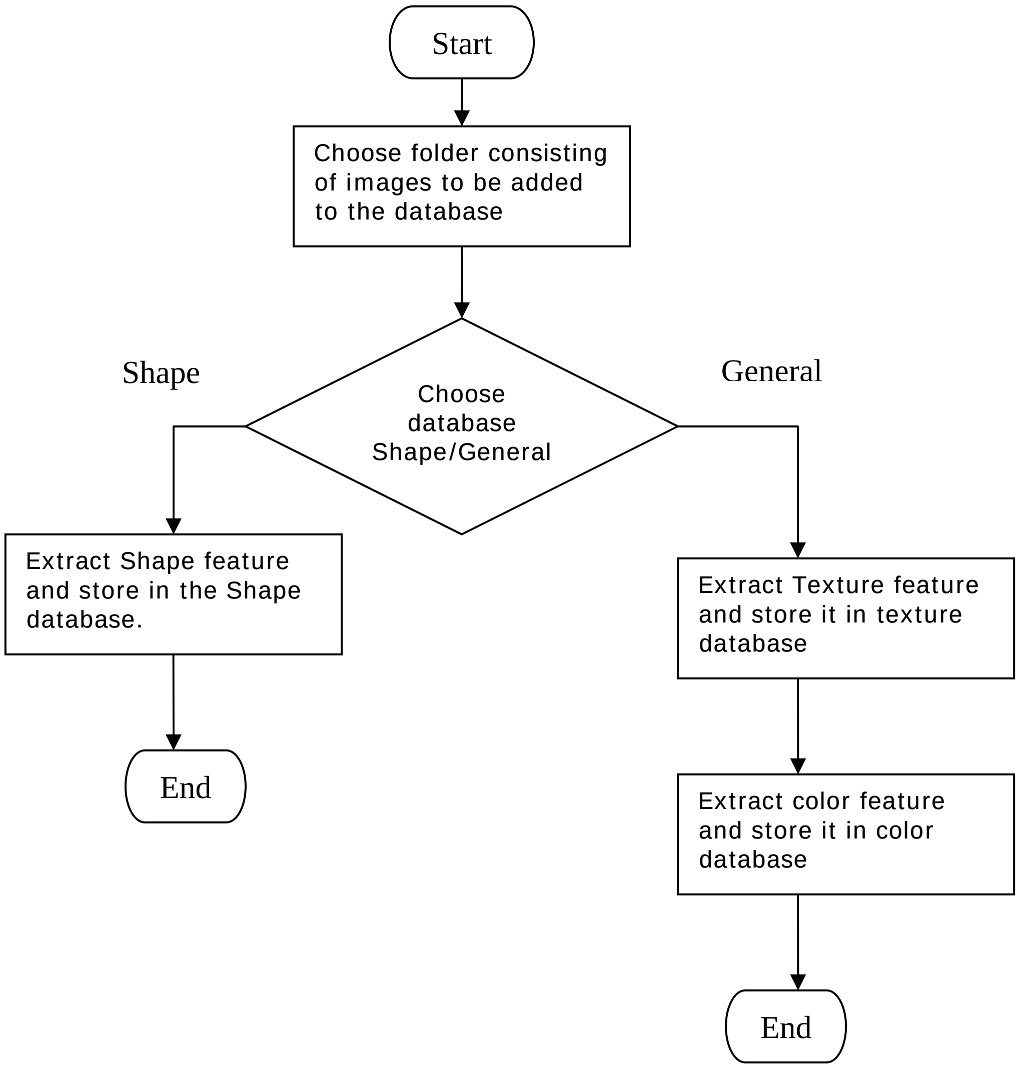


Figure 3.2 : Flow Chart to add images into the database

Feature extraction is explained in the following chapters. The database of the color image and texture images is common. When a query is fired for texture or color same database is searched. For shape based image retrieval the database is separate and the images having only geometric figures and monolithic area can be identified.

3.4 FLOW CHART FOR IMAGE RETREIVAL

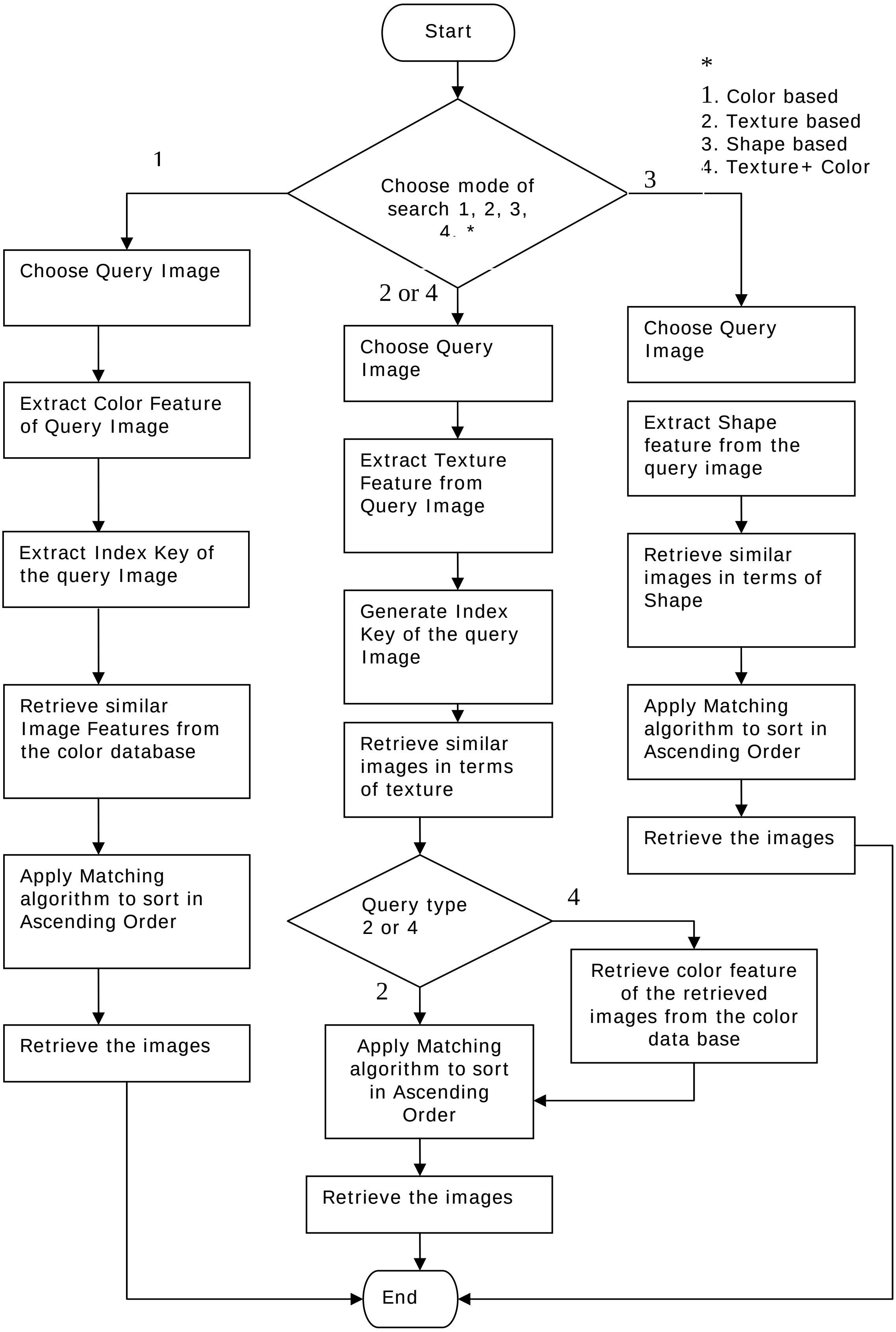


Figure 3.3 : Flow Chart to retrieve images from the data base

4 COLOR BASED FEATURE EXTRACTION

4.1 INTRODUCTION OF COLOR BASE FEATURE EXTRACTION

Color features are among the most important and extensively used low-level features in image database retrieval. They are usually robust in noise, resolution, and orientation and resizing. Due to their little semantic meaning and compact representation, color features tend to be domain independent compared to other features. Combined with image segmentation, color features can be used to describe the appearance of the image and even generate semantic annotations.

In color based indexing, given a query image, the goal is to retrieve all the images whose color compositions are similar to the color composition of the query image. Typically, the color content is characterized by color histograms, which are compared using the histogram distance measure. As an improvement of a basic histogram search, several more sophisticated distance metrics have been developed.

The histogram is a list of "bins" showing the number of pixels being classified into the different color or color groups from the color model. If true color images are used and tried to find the histogram, the statistics say R*G*B = 256*256*256=167772216 colors. That is, the memory requirement would be an array of 167772216 integers for one image. Therefore to reduce the memory requirement and in accordance to the psychology of vision that states "generally people are not able to identify many different colors", *Irena Valova*, *Boris Rachev* [6] suggested ten colors for the purpose of conducting the experiments.

Colour Descriptor	Colour Mapped
0	Uncertain Colours: "very dark" or "very bright"
1	White
2	Grey
3	Black
4	Red, Pink
5	Brown, Dark Yellow, Olive
6	Yellow, Orange, Light Yellow
7	Green, Lime
8	Blue, Cyan, Aqua, Turquoise
9	Purple, Violet, Magenta

Fig. 4.1 Color model suggested by Boris and Rechev

4.2 COLOR DESCRIPTORS

The color descriptor as defined in [5, 6] :

Color Descriptor $= \{\{c_i, p\}, i=1..M\};$

where M is the total number of color clusters in the image, c is numeral corresponding to the color *i and* *p* is its percentage, and $\sum p\ i=1$. Note that M can vary from image to image. First, the most dominant color in the image is added and then the less dominant color follows. As each pair $\{c_i, p\}$ is added, the color descriptor becomes more expressive representation of the color distribution of an image. This means that the color covering the largest area of an image is considered as the main color feature and its pair is the first one in the color descriptor. The second largest area of the same image defines the second one and the rest of pairs.

Based on the above color model, to store the color image features, [6] proposes two index structures: index key - for the global color features and index matrix - for the spatial information in every image.

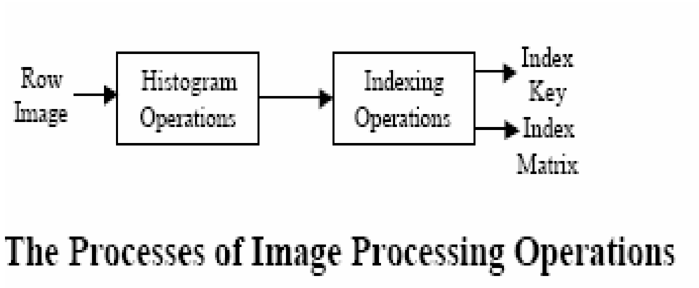


Fig. 4.2 Color Feature Extraction

Index key – representing the global features and would be used in the indexing of the images based on colors. Where C_i is color descriptor for the i color from the perceptual color model .To generate the Index Key the histogram can be used to indicate the different counts for each color descriptor in an image, the perceptual color group, which has the largest count in the histogram, may be regarded as the most dominant color group. Thus the less dominant perceptual color group should contain the lesser amount in the histogram.

Index Key C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9

Index Matrix - Index Matrix is representation of spatial Information as index key do not retain the information about the distribution of the color in the image space. In order to create this index structure the whole image is divided into 256 equal parts. In this index matrix is stored the coefficient of the dominant color in the corresponding part. The original images were 16x16 quantized and were represented as16x16 color blocks.

Figure.4.3 Index Matrix

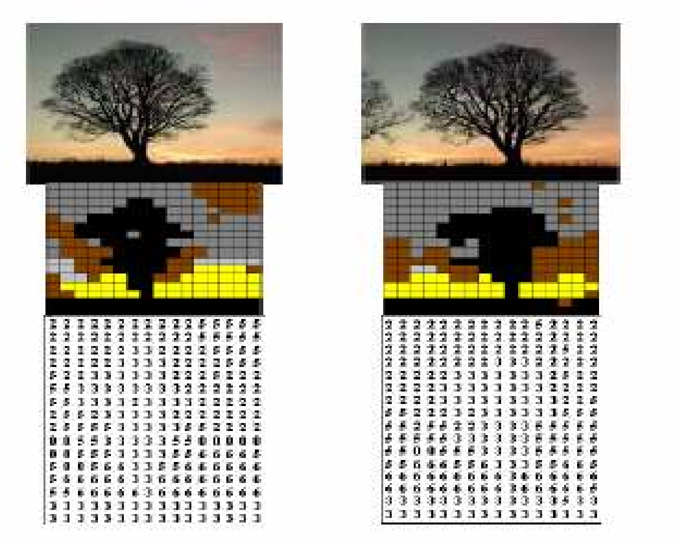


Fig 4.4 Representation of the color features

4.3 COLOR QUANTIZATION USED IN THE SYSTEM

The approach used in the system is not the exact but a bit modification in the above proposed algorithm for the reason that the number of colors in the suggested model is very small and hence gave much unexpected results. So instead of following the mentioned schema, 16 color schema has been used by dividing the RGB color cube into 16 equal colors. As shown in the figure 4.5, for every division the middle color is selected as the representative color. All the colors in the image is replaced by one of the representative colors. So instead of having 16 million colors now the image has just 16 colors to be compared as in figure 4.6.

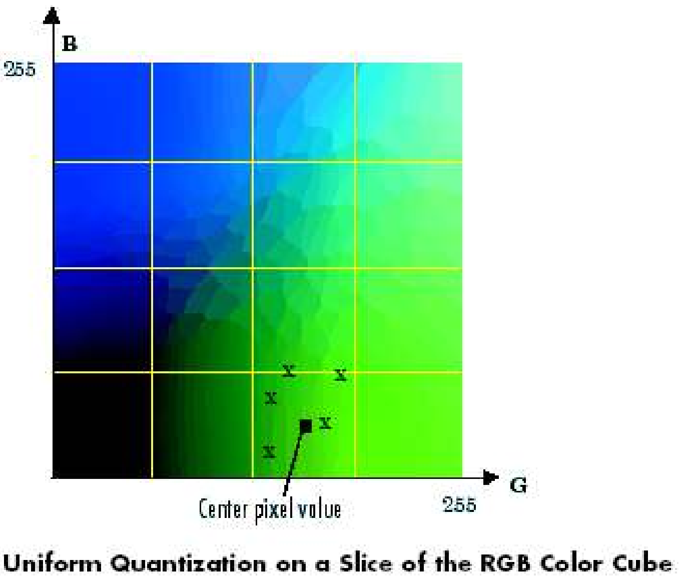


Fig 4.5 16 color model

Fig 4.6 Quantizing image into 16 colors

4.4 DIVIDING INTO GRIDS

The quantized image is divided into 5X5 matrix and the 16 bin histogram is calculated for each sub image and the most dominant color is the new replacement of the small block and hence the 125X83 images are reduced to 25X16. Now every element in the small matrix roughly has information of the region.

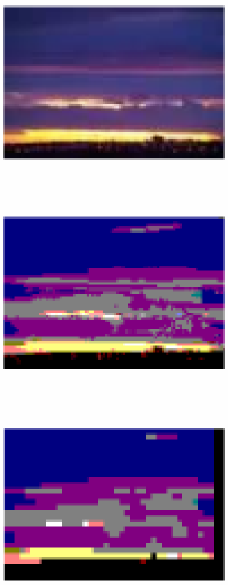


Figure 4.7 Image, Quantized Image, Index matrix

4.5 DATABASE MANAGEMENT OF THE SYSTEM AND INDEXING

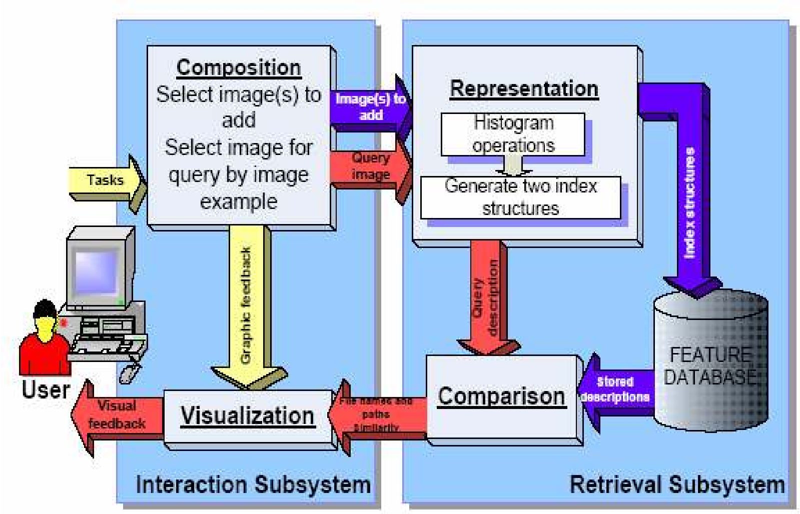


Figure 4.8 Data Base Management

The DBMS used is postgreSQL 8.1 for windows. The color descriptors as explained in section 4.2 are evaluated using histogram operations. The two index structure is generated for each image and stored in the database as array of integers. PostgreSQL supports array of basic data structures as data types hence saving index structure as one dimension and two dimension is simplified. When query is given by presenting the example image, similar processing is performed on the query image and used for comparison with the data base.

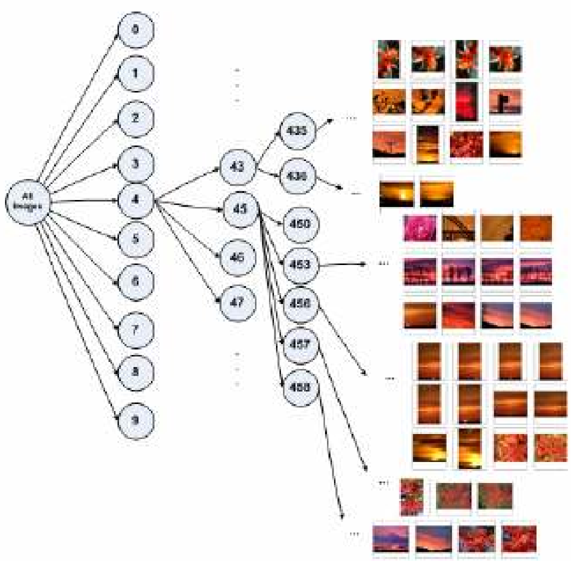


Figure 4.9 Indexing Image based on color

The diagrammatic presentation of the data base management is in figure 4.8. The Indexing Technique [9] can be used for hierarchical structure of the database. Figure 4.9 above demonstrates the relationships between the color descriptors as a graphical representation of the hierarchical classification.

4.6 QUERY FORMS

The query image is the image, similar to which the system is expected to retrieve the class of image. The query can be made in one of the following ways.

4.6.1 QUERY BY EXAMPLE: A user can select an image from the browser to retrieve images similar to the selected image.

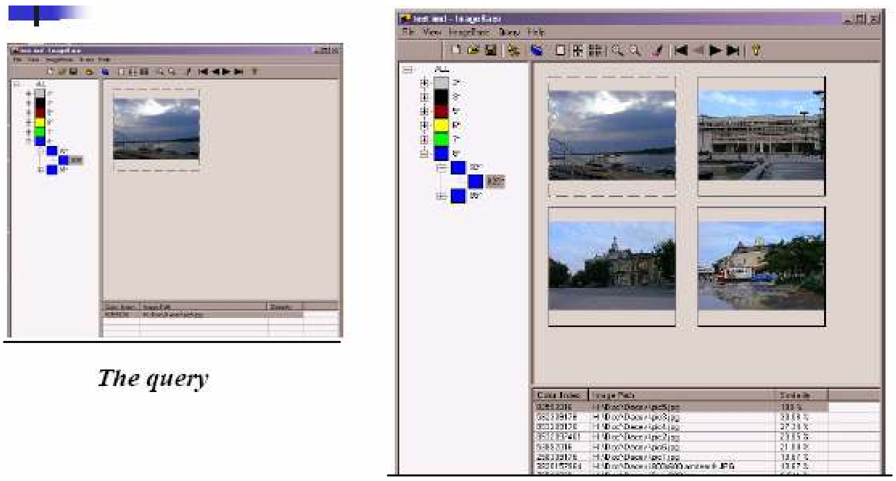


Figure. 4.10 Query By example

4.6.2 QUERY BY USER CONSTRUCTION:- A grid is provided with the choice of the representative colors from the proposed color models. The user can construct their own color distribution as shown in the figure 4.11 below.

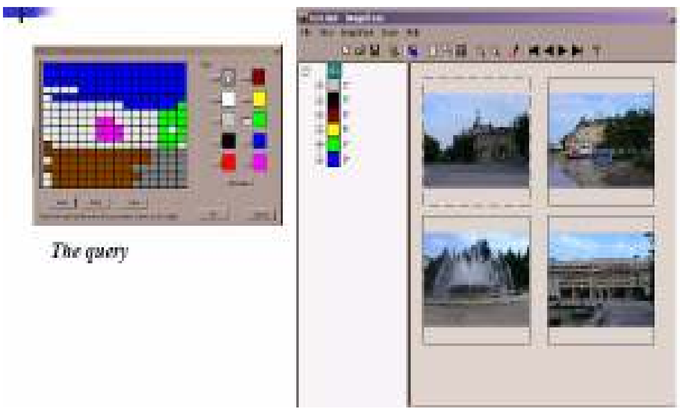


Figure 4.11 Query by user sketch

4.7 SIMILARITY MATCHING

Algorithms of similarity measures are used to perform the degree of similarity between the Query Image and the Images from database, which provide a ranking to arrange the order of images. The similarity measure we use reflects our understanding of similarity and includes global and spatial local factors. The global factor takes into account the presence of common global color distribution for both images, while the spatial local factor accounts for similarity of the spatial locations of the images common for both images.

Similarity measures are usually associated with the query types. Image retrieval systems usually support several types of queries – query by global features, query by image example or query by user sketch. As a global factor we suggest and use the representation of the database as an ordered hierarchical tree. This representation is very comfortable for fast execution of global features queries, because the color descriptor is sorted by dominant color and it is easy to create an ordered tree like this in figure 1 as well as to search in it for images with similar dominant colors or similar color distribution.

$$Sim(I,Q) = \left|\frac{I \cap Q}{I \cup Q}\right|$$

Where I is the ordered set of color descriptors representing the 2-D Vector of the images in the image database and Q is the ordered set of color descriptors representing the 2-D Vector of the query image. Since this algorithm supports only the queries by image example and by user sketch, the size of $Q\ I$ U is always the same as the size of the Color Descriptor Matrix (which is 256 in our prototype system). The size of $Q\ I$ I is determined by the common by value and position descriptors in the 2-D Vectors of the image and the query. The more common the descriptors in I and Q are, the higher the similarity degree.

Some of the similarity measure parameters are described here:

- **Simple matching coefficient[6]**=12 (12 color coefficients are equal in the two matrixes)

- **Jaccard's coefficient [25]**= 12/16=0.75 (represent the ratio between the count of the equal coefficients in the two matrixes and the total count of the coefficients in the matrix)

- **Euclidian Distance[7]** Given two feature points, our distance measure is the Euclidean distance between the points multiplied by a Bottleneck factor (bnf). This factor receives a low value (bnf < 1) for points which our algorithm decided are likely belong to the same cluster and, a high value (bnf > 1) for points which our algorithm decided belong to different clusters.

In this system the metadata is subtracted from other and the methods used to judge similarity are number of zeros in the difference matrix. Higher the number of zeros more is the similarity. Also the absolute sum is used as the measure to compare the similarity, and the Euclidian distance but all the methods give similar results.

5 TEXTURE BASE FEATURE EXTRACTION

5.1 INTRODUCTION TO TEXTURE

Texture refers to the arrangement of the basic constituents of a material. In a digital image, texture is depicted by spatial interrelationships between, spatial arrangements of the image pixels**.**

Figure 5.1 Textured image

Texture is an important item of information that human use in analyzing a scene. It is particularly useful in the analysis of natural environments, as most natural scenes consist of textured surfaces. Literally, texture refers to the arrangement of the basic constituents of a material. In a digital image, texture is depicted by spatial interrelationships between, and/or spatial arrangement of the image pixels. Visually, these spatial interrelationships, or arrangement of image pixels, are seen as changes in the intensity patterns, or gray tones. Thus in automatic analysis, information about texture has to be derived from the gray tones of the image pixels.

Haralick [8] categorized the various textures into three groups: the statistical techniques, the structural methods, and the statistical-structural approaches. A major disadvantage of almost all of these approaches is that they do not have general applicability - they cannot be applied to different classes of textures with reasonable success. For instance while the statistical techniques are generally good for micro textures and are poor performers on macro textures, the reverse is the case for the structural techniques. Another disadvantage of some of the existing methods is the computational cost involved, either in terms of memory requirement, computation time or implementation complexity.

The limitation of most of the textural models is that they are for gray scale images and not for color images. The color property of the image can not be ignored if the model wants to follow human perception. Textures are also clustered or identified as similar, based on the color dexterity if we talk about human perception. In human perception an image with similar color organization will appear closer as compared to an image with different color organization when they have same textural behavior.

But extending the co-occurrence for color image has a limitation. The co-occurrence matrix for the color image would require matrix of order 16 million by 16 million. The feature extracted would require large amount of computation to be done and if the size of the database is substantial the feature extraction would take significant time. The solution to this could be, converting the image to gray scale and then evaluating the texture features but it looses the color features. Motivated by the limitations, extracting the texture feature of the image in grayscale and retaining the color feature in a different database is the proposed solution.

5.2 PROPOSED SYSTEM

All texture analysis techniques are proposed for gray scale images and not for color images. But the computational time and resource required for color images is at least three times that of gray scale images. And since the color features are already extracted in the previous section and stored and thus the image can be reduced to grey scale and then the textural features can be extracted.

5.3 CO OCCURRENCE MATRIX

Co-occurrence matrix can be defined as the co-occurrence of the intensity values at a particular distance and at a particular angle. So for an image of n intensity levels the co-occurrence matrix would have the nXn dimension for a particular offset and in a particular direction. This signifies how the intensity values are distributed in the specified direction and in a particular angle. The co-occurrence matrix then can be used to calculate various parameters like correlation, contrast etc. Correlation, also called correlation coefficient, indicates the strength and direction of a linear relationship between two random variables. In general statistical usage, correlation or co-relation refers to the departure of two variables from independence. As shown in figure 5.2, for an image of size 5X5 and 3 intensity levels [0 1 2] the co-occurrence matrix for angle 0^{o} and off set of (0,1) would be a matrix of size 3X3.

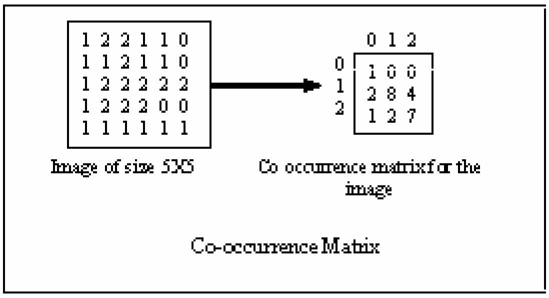


Figure 5.2 Co occurrence Matrix

Usually, four statistic features are used to describe the texture characteristic in retrieval: they are Contrast, Entropy, Energy and Homogeneity. In practice, four directional matrixes are evaluated according to angles of 0, 45, 90 and 135 degrees. As can be shown by a diagram which find co occurrence matrix in the direction 0 degrees and offset d=1

Property	Description	Formula
'Contrast'	Returns a measure of the intensity contrast between a pixel and its neighbor over the whole image. `Range = [0 (size(GLCM,1)-1)^2]` Contrast is 0 for a constant image.	$\sum_{i,j} \lvert i-j \rvert^2 p(i,j)$
'Correlation'	Returns a measure of how correlated a pixel is to its neighbor over the whole image. `Range = [-1 1]` Correlation is 1 or -1 for a perfectly positively or negatively correlated image. Correlation is `NaN` for a constant image.	$\sum_{i,j} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j}$
'Energy'	Returns the sum of squared elements in the GLCM. `Range = [0 1]` Energy is 1 for a constant image.	$\sum_{i,j} p(i,j)^2$
'Homogeneity'	Returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal. `Range = [0 1]` Homogeneity is 1 for a diagonal GLCM.	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j \rvert}$

Fig 5.3 Mathematical Interpretation of Textural Properties

5.3.1 Comparing properties

To an image, a four dimensions vector T=[T1,T2,T2,T3] will be obtained as the final texture feature in retrieval. Here we use the Euclidean distance formula in matching of features.

$$D_{i,j} = \sqrt{(T_{i,1} - T_{j,1})^2 + (T_{i,2} - T_{j,2})^2 + (T_{i,3} - T_{j,3})^2 + (T_{i,4} - T_{j,4})^2}$$

In view of the meaning and value mange of the four components in the vector, it is necessary to normalize them in retrieval.

5.4 NEIGHBORHOOD GRAY-TONE DIFFERENCE MATRIX (NGTDM)

This method is proposed by Moses Amadasun., and Robert King in his work [9] to extract various textural properties. This is a column matrix formed as follows. Let *f (k, I)* be the gray tone of any pixel at *(k , I)* having graytone value *i.* Then we find the average gray-tone over a neighborhood centered at, but excluding *(k, I).*

$$\overline{A} = \overline{A(k,t)} = \frac{1}{w-1}\left[\sum_{m=-d}^{d}\sum_{n=-d}^{d} f(k+m, l+n)\right](m,n) \neq (0,0)$$

where d specifies the neighborhood size and W = $(2d + 1)^2$ .Then the ith entry in the NGTDM is

$$s(i) = \sum |i - A'_i|, \quad \text{for } i \in N_j \text{ if } N_j \neq 0,$$
$$= 0, \quad \text{otherwise}$$

where {N} is the set of all pixels having gray tone *i* (except in the peripheral regions of width *d)*

Illustration

Consider the 5 x 5 sample image shown in Fig. 5.4 Specifying a distance, d = 1, results in a 3 x 3 neighborhood. This neighborhood can only be centered on pixels within the indicated square.

1	1	4	3	1
3	4	0	1	1
5	4	2	2	2
2	1	1	4	4
0	2	2	5	1

i	$s(i)$
0	2.750
1	4.125
2	0.250
3	0.000
4	4.875

Figure 5.4 NGTDM for Sample Images.

The other pixels are considered as being in the periphery of the image.
There are two pixels within the indicated square with gray tone = 2. Thus for this image

$$s(2) = \left|2 - \frac{17}{8}\right| + \left|2 - \frac{15}{8}\right| = 0.250.$$

In similar fashion, we have s(0) = 2.750; s(1) = 4.125; s(4) = 4.875; and s(3) is necessarily zero. The NGTDM for this sample image is as shown in Fig

5.4.1 Textural Properties and their Computational Approximations in NGTDM:-

A. Coarseness

This is the most fundamental property of texture, and in a narrow sense, it is used to imply texture. In a coarse texture, the primitives or basic patterns making up the texture are large. As a result, such a texture tends to possess a high degree of local uniformity in intensity, even over a fairly large area. In other words, the spatial rate of change in intensity is slight. Therefore the intensities of neighboring pixels would tend to be similar; thus there would be small differences between the gray tones of pixels and the average gray tones of their neighborhoods. Hence the summation of such differences computed over all image pixels would give an indication of the level of spatial rate of change in intensity, and thereby (in an inverse manner) show the level of coarseness of the texture.

This summation is the same **as** adding up the entries in the NGTDM. However in the summation each entry is weighted by the probability of occurrence of the corresponding intensity value.

$$f_{\cos} = \left[\varepsilon + \sum_{i=0}^{G_h} p_i s(i) \right]^{-1}$$

B. Contrast

Perceptually, an image is said to have a high level of contrast if areas of different intensity levels are clearly visible. Thus high contrast means that the intensity difference between neighboring regions is large. This is usually the case when the dynamic range of gray scale is large or when it is stretched. Also the spatial frequency of the changes in intensity (i.e., the amount of local intensity variations) will affect the contrast of an image. For instance, a small checkerboard will appear to have a higher contrast than a coarse checkerboard for the same gray scale range.

$$f_{con} = \left[\frac{1}{N_g(N_g - 1)} \sum_{i=o}^{G_h} \sum_{j=0}^{G_h} p_i p_j (i - j)^2 \right] \left[\frac{1}{n^2} \sum_{i=o}^{G_h} s(i) \right]$$

C. ***Busyness***

A busy texture is one in which there are rapid changes of intensity from one pixel to its neighbor; that is the spatial frequency of intensity changes is very high. If these changes are very small in magnitude, they may not be visually noticeable and a high level of local uniformity in intensity may be perceived. On the other hand, if the spatial frequency of changes in intensity is very low, a high degree of local uniformity may still be perceived, even if the magnitude of the changes is large. While the spatial frequency of intensity changes reflects the level of busyness, the magnitude of these changes depends upon the dynamic range of gray scale, and thus relates to contrast.

Therefore a suppression of the contrast aspect from the information about spatial rate of change in intensity may indicate the degree of busyness of a texture. The following computational measure is proposed:

$$f_{bus} = [\sum_{i=0}^{G_h} p_i s(i)] \Big/ [\sum_{i=0}^{G_h} \sum_{j=0}^{G_h} ip_i - jp_j], \quad p_i \neq 0, p_j \neq 0$$

D .Complexity

Complexity refers to the visual information content of a texture. **A** texture is considered complex if the information content is high. This occurs when there are many patches or primitives present in the texture, and more so when the primitives have different average intensities. (Complexity could also be related to the shape of the individual primitives, but this is not used in this definition.) Again a texture with a large number of sharp edges and/or lines may be considered as complex. All these depend upon the spatial period of pattern repetition and on the dynamic range of gray scale. Thus complexity is in part correlated with busyness and contrast.

$$f_{com} = \sum_{i=o}^{G_h} \sum_{j=0}^{G_h} \left\{ (\mid i - j \mid) \Big/ \left(n^2 (p_i + p_j) \right) \right\} \left\{ p_i s(i) + p_j s(j) \right\}$$

$$p_i \neq 0, p_j \neq 0$$

E. Texture Strength

The term texture strength is a difficult concept to define concisely. However a texture is generally referred to as strong when the primitives that comprise it are easily definable and clearly visible. Such textures generally tend to look attractive, as they present a high degree of visual feel. But the ease with which distinctions can be made between the component primitives of a texture depends to a considerable extent upon the sizes of the primitives and the differences between their average intensities. For instance it may be possible to distinguish between large primitives (coarse texture) even with small differences between their average intensities.

However for such distinctions to be made between small primitives (e.g., in busy textures), there must be wide differences between their intensities. Thus, in part, texture strength may be correlated with coarseness and contrast. We therefore propose the following computational approximation for this property

$$f_{str} = [\sum_{i=o}^{G_h} \sum_{j=o}^{G_h} (p_i + p_j)(i-j)^2] \Big/ [\varepsilon + \sum_{i=0}^{G_h} s(i)],$$

$$p_i \neq 0, p_j \neq 0$$

Again the Euclidean distance can be found to measure the similarity measure in between the images.

5.6 Methodology Used For implementation

To extract the texture feature the image is converted to gray scale since the color features has already been reserved in the database. The parameters to be fixed for co-occurrence matrix is an offset and at a particular angle. Observing that the size of the pattern repeated also known as texture, vary from three to five pixels to almost one third of the image size. By experimentation and observing on various types of images the co-occurrence matrix was calculated for the offset of 5, 10, 15 up to approximately one third the size of the image. The angles at which the values are calculated are 0 degree, 45 degree and 90 degree so that it can identify all orientation of the textures . The texture database consists of eight values at mentioned offset for all the three angles. The above conclusion was drawn after observing the values of correlation obtained from each co-occurrence matrix.

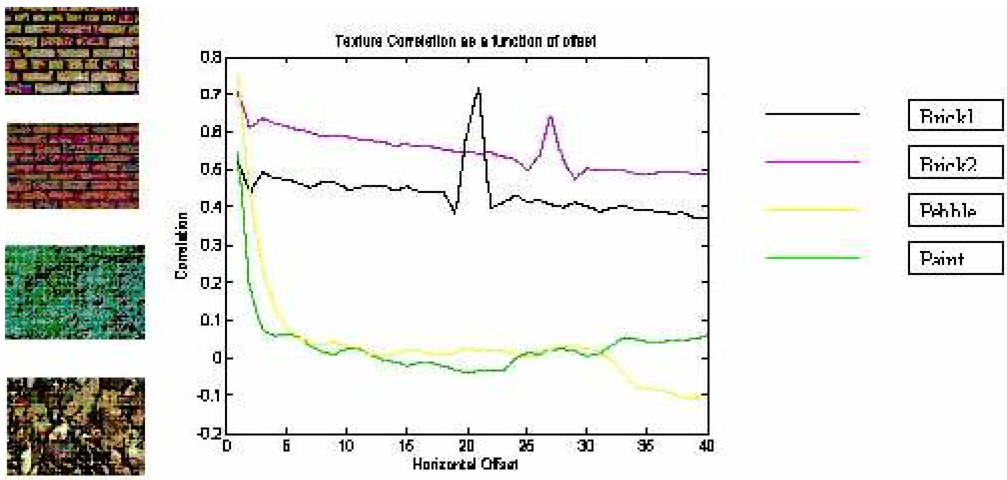


Figure 5.5 Correlation for different texture.

As in figure 5.5 correlation it is obvious that the similar textures have similer correlation nature, When Euclidian distance between the values of correlation at various offset was found, images with similar texture had small distances. And also the peaks in the graph shoes high values of correlation at particular offset, this means that the images has regular texture or repetition of a particular pattern. The Standard deviation of the values was a good measurement to say the amount of variation in the correlation factor. And hence along with the correlation values standard deviation was also the field used as the measurement of the uniformity in the texture.

5.6 TEXTURE DATA BASE

As discussed earlier the parameters were fixed based on analysis and mainly tried with 3 approaches .The approaches are Categorized based on the database used and the parameters extracted from the co-occurrence matrix.

Approach 1

Approach 2

Approach 3

The modules used in all the three approaches are

1. Extracting the features from the images and Writing into the database
2. Extracting the feature of the query image.

3. Reading from the database.
4. Finding Euclidian distance from the Query image.
5. Sorting the distances in ascending order.
6. Displaying the top 24 images.

5.6.1 Approach 1

In this approach we used four tables each with 9 columns. Table names and their fields are as shown.

- Image_correlation: (Image_name, Corr1, Corr2 , Corr3, Corr4, Corr5, Corr6, Corr7, Corr8)

- Image_homogeneity (Image_Name,hom1, hom2, hom3, ho4, hom5, hom6, hom7, hom8);

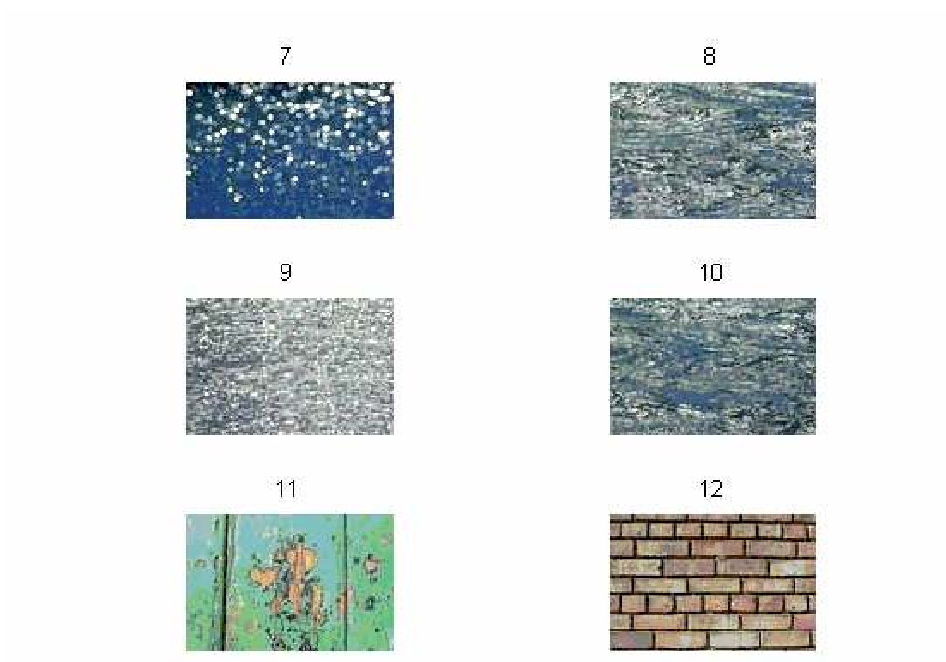
- Image_contrast (Image_name, con1, con2, con3, con4, con5, con6 , con7, con8)

- Image_energy (Image_name, ene1, ene2, ene3, ene4, ene5, ene6, ene7, ene8)

Where ***cor*** represents correlation ***con*** represents contrast ***ene*** represents energy and ***hom*** represents homogeneity calculated as per the mathematical models given above. The suffix represents values of the same at various offsets. The results obtained were not satisfactory . This could cluster with high contrast but clustering of even images were not to the mark. As shown below in the images. After observation it was sorted out that images of uniform nature has high value of correlation at some points and hence to exploit that second approach was thought.


Query Image


1
2
3
4
5
6


7
8
9
10
11
12

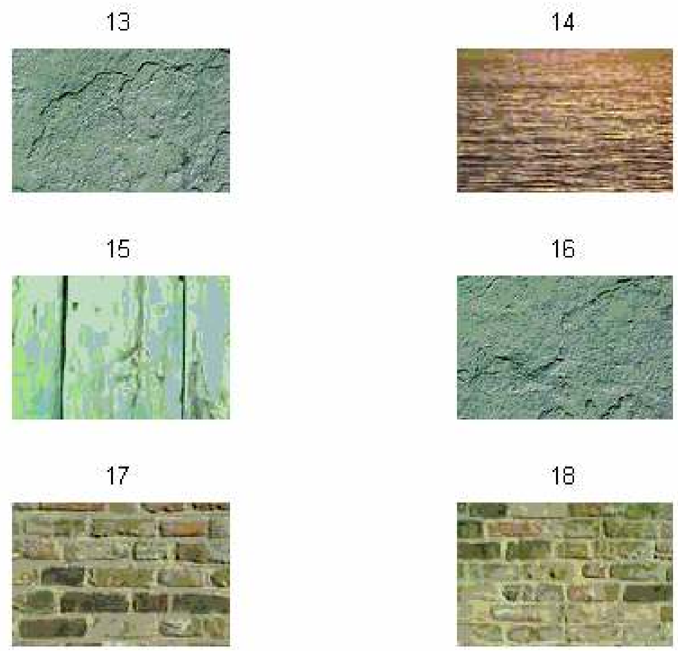


Figure 5.6 Image retrieved by Approach 1

5.6.2 Approach 2

Trying to exploit the behavior of the curve in the graph a new data base was designed on the above features with just one table containing maximum and median values of the various properties. Thus the table had following structure

Textural_feature(Image_name, corr_median, corr_max, con_median, con_max, hom_median, hom_max, ene_median, ene_max)

When corr_median gives median of correlation values calculated at various offset.
Also corr_max was the maximum value of the correlation values calculated at various offsets. But this features gave even bad results, as it was not able to give proper signature of the textural properties.

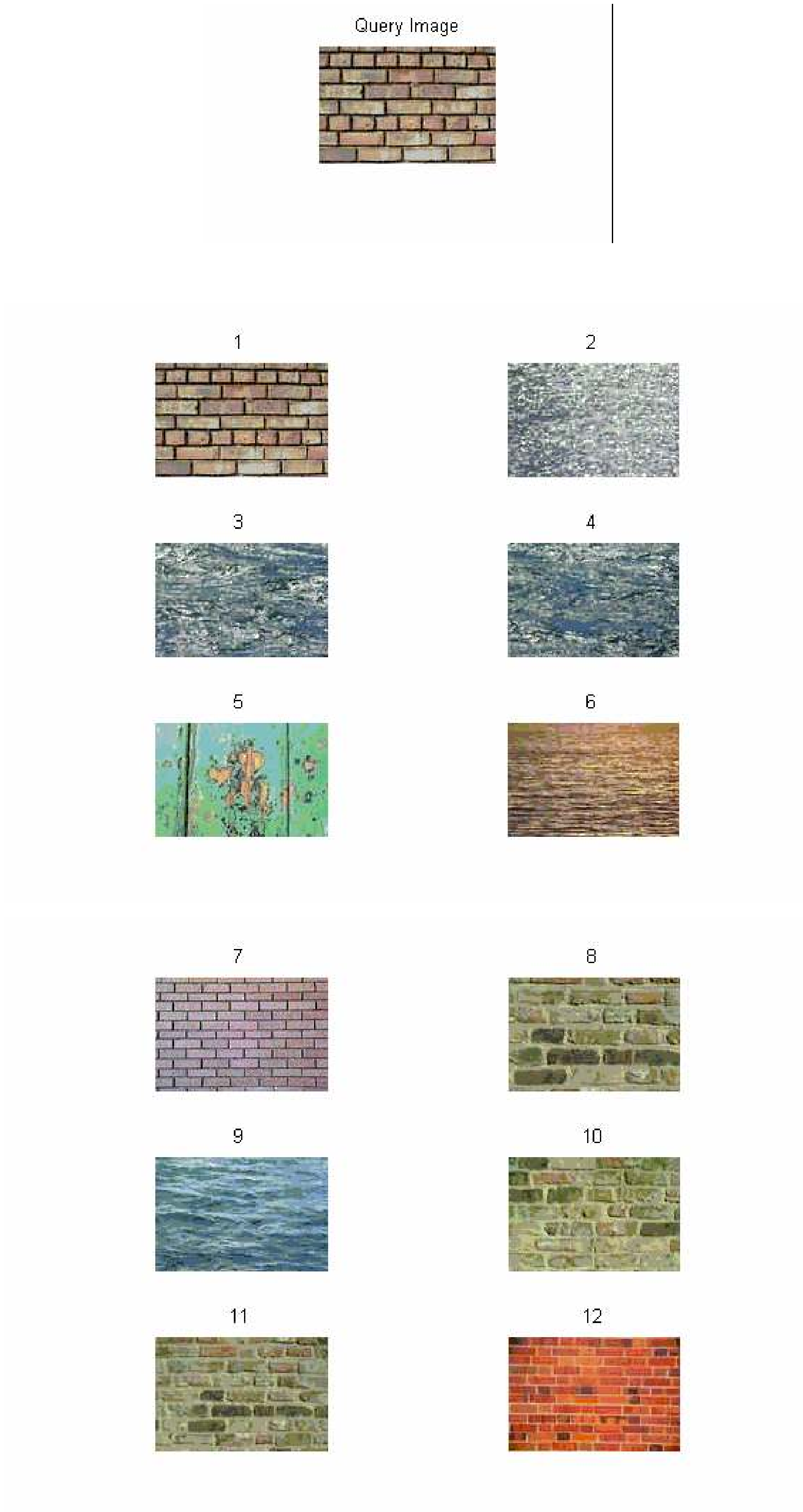
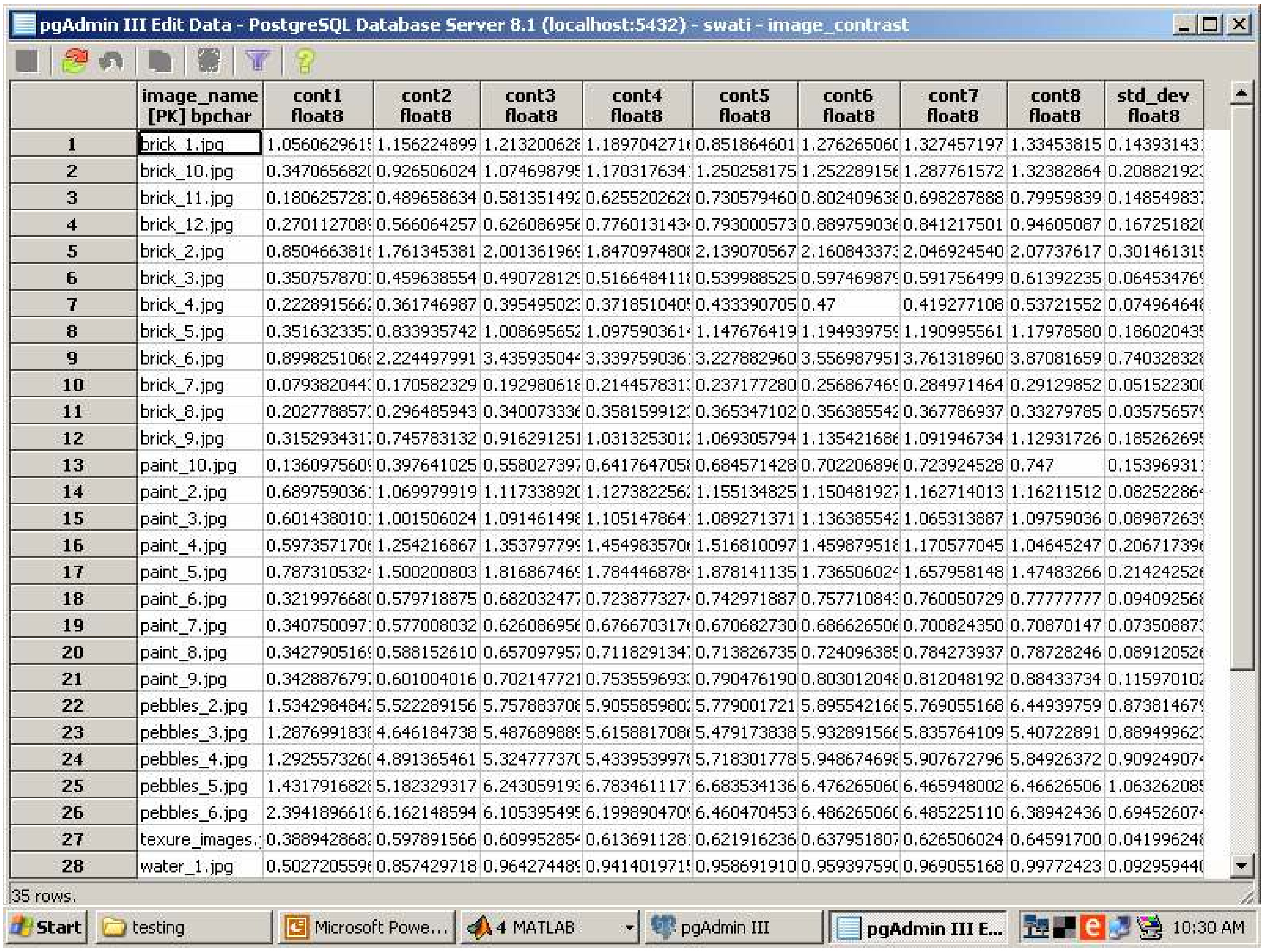


Figure 5.7 Image retrieved by Approach 2

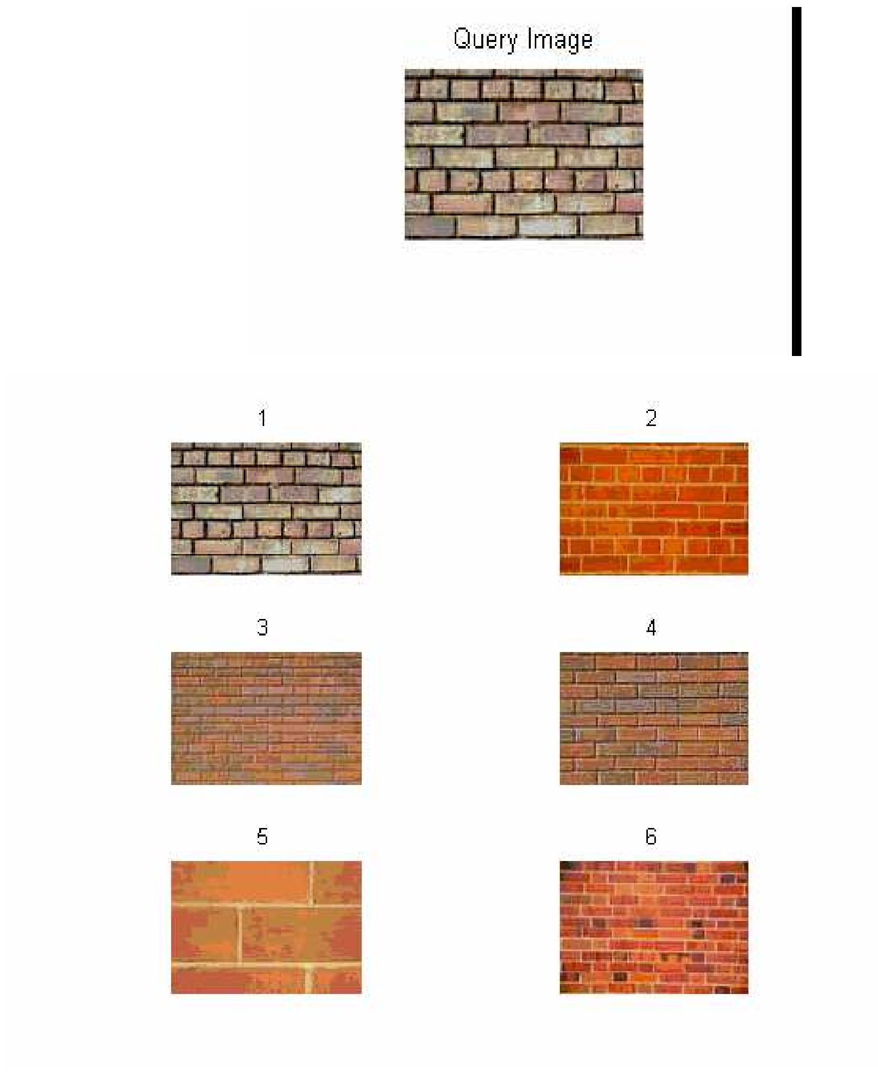
5.6.3 Approach 3

Hence added a field in the database of approach 1 , known as standard deviation of all the four properties in the respective table this enhanced the performance to a great extend. And hence one of the table for the database looks as shown in figure 5.8.



	image_name [PK] bpchar	cont1 float8	cont2 float8	cont3 float8	cont4 float8	cont5 float8	cont6 float8	cont7 float8	cont8 float8	std_dev float8
1	brick_1.jpg	1.056062961	1.156224899	1.21320062	1.189704271	0.851864601	1.27626506	1.327457197	1.33453815	0.14393143
2	brick_10.jpg	0.347065682	0.926506024	1.07469879	1.170317634	1.250258175	1.25228915	1.287761572	1.32382864	0.20882192
3	brick_11.jpg	0.180625728	0.489658634	0.58135149	0.625520262	0.730579460	0.80240963	0.698287888	0.79959839	0.14854983
4	brick_12.jpg	0.270112708	0.566064257	0.62608695	0.776013143	0.793000573	0.88975903	0.841217501	0.94605087	0.16725182
5	brick_2.jpg	0.850466381	1.761345381	2.00136196	1.847097480	2.139070567	2.16084337	2.046924540	2.07737617	0.30146131
6	brick_3.jpg	0.350757870	0.459638554	0.49072812	0.516648411	0.539988525	0.59746987	0.591756499	0.61392235	0.06453476
7	brick_4.jpg	0.222891566	0.361746987	0.39549502	0.371851040	0.433390705	0.47	0.419277108	0.53721552	0.07496464
8	brick_5.jpg	0.351632335	0.833935742	1.00869565	1.097590361	1.147676419	1.19493975	1.190995561	1.17978580	0.18602043
9	brick_6.jpg	0.899825106	2.224497991	3.43593504	3.339759036	3.227882960	3.556987951	3.761318960	3.87081659	0.74032832
10	brick_7.jpg	0.079382044	0.170582329	0.19298061	0.214457831	0.237177280	0.25686746	0.284971464	0.29129852	0.05152230
11	brick_8.jpg	0.202778857	0.296485943	0.34007333	0.358159912	0.365347102	0.35638554	0.367786937	0.33279785	0.03575657
12	brick_9.jpg	0.315293431	0.745783132	0.916291251	1.031325301	1.069305794	1.13542168	1.091946734	1.12931726	0.18526269
13	paint_10.jpg	0.136097560	0.397641025	0.55802739	0.641764705	0.684571428	0.70220689	0.723924528	0.747	0.15396931
14	paint_2.jpg	0.689759036	1.069979919	1.11733892	1.127382256	1.155134825	1.15048192	1.162714013	1.16211512	0.08252286
15	paint_3.jpg	0.601438010	1.001506024	1.09146149	1.105147864	1.089271371	1.13638554	1.065313887	1.09759036	0.08987263
16	paint_4.jpg	0.597357170	1.254216867	1.35379779	1.454983570	1.516810097	1.45987951	1.170577045	1.04645247	0.20671739
17	paint_5.jpg	0.787310532	1.500200803	1.81686746	1.784446878	1.878141135	1.73650602	1.657958148	1.47483266	0.21424252
18	paint_6.jpg	0.321997668	0.579718875	0.68203247	0.723877327	0.742971887	0.75771084	0.760050729	0.77777777	0.09409256
19	paint_7.jpg	0.340750097	0.577008032	0.62608695	0.676670317	0.670682730	0.68662650	0.700824350	0.70870147	0.07350887
20	paint_8.jpg	0.342790516	0.588152610	0.65709795	0.711829134	0.713826735	0.72409638	0.784273937	0.78728246	0.08912052
21	paint_9.jpg	0.342887679	0.601004016	0.702147721	0.753559693	0.790476190	0.80301204	0.812048192	0.88433734	0.11597010
22	pebbles_2.jpg	1.534298484	5.522289156	5.75788370	5.905585980	5.779001721	5.89554216	5.769055168	6.44939759	0.87381467
23	pebbles_3.jpg	1.287699183	4.646184738	5.48768988	5.615881708	5.479173838	5.93289156	5.835764109	5.40722891	0.88949962
24	pebbles_4.jpg	1.292557326	4.891365461	5.32477737	5.433953997	5.718301778	5.94867469	5.907672796	5.84926372	0.90924907
25	pebbles_5.jpg	1.431791682	5.182329317	6.24305919	6.783461117	6.683534136	6.47626506	6.465948002	6.46626506	1.06326208
26	pebbles_6.jpg	2.394189661	6.162148594	6.10539549	6.199890470	6.460470453	6.48626506	6.485225110	6.38942436	0.69452607
27	texure_images.	0.388942868	0.597891566	0.60995285	0.613691128	0.621916236	0.63795180	0.626506024	0.64591700	0.04199624
28	water_1.jpg	0.502720559	0.857429718	0.96427448	0.941401971	0.958691910	0.95939759	0.969055168	0.99772423	0.09295944

Figure 5.8 image_correlation table with added std deviation


Query Image
1
2
3
4
5
6

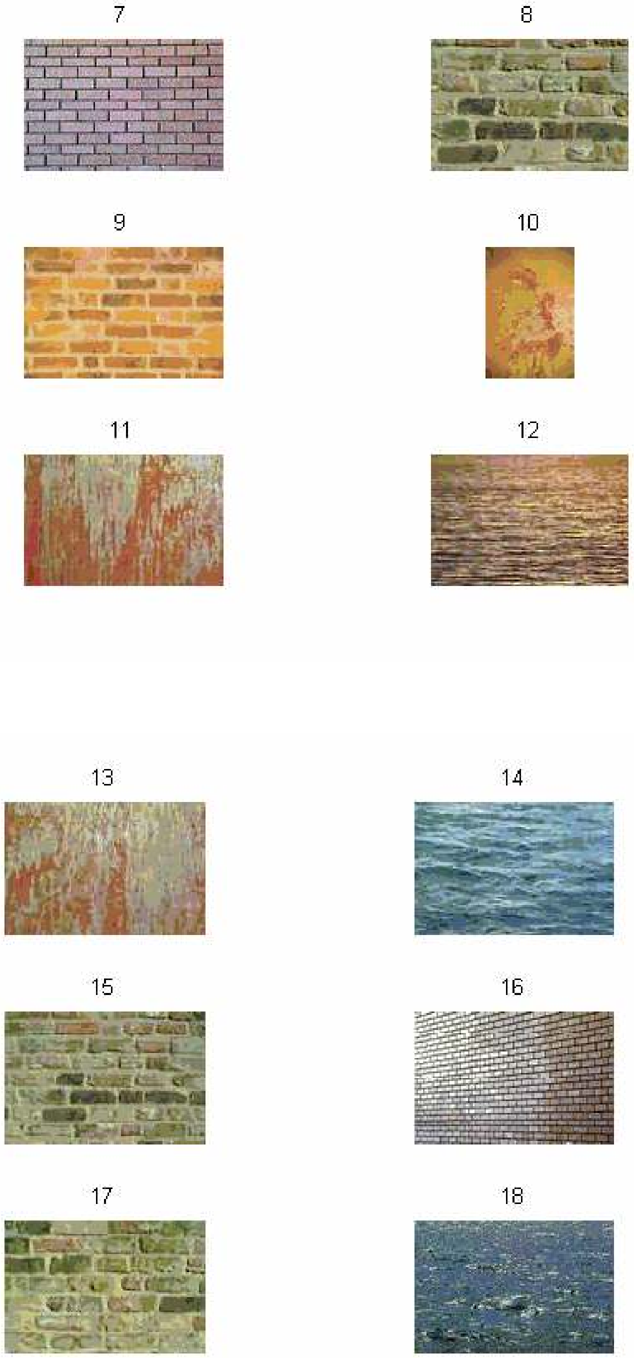


Figure 5.9 Images retrieved by Approach 3

6 COMBINING TEXTURE WITH COLOR

6.1 WHY COMBINE TEXTURE WITH COLOR

The limitation of most of the textural models is that they are for gray scale images and not for color images. The color property of the image can not be ignored if the model wants to follow human perception. Textures are also clustered or identified as similar, based on the color dexterity if we talk about human perception. In human perception an image with similar color organization will appear closer as compared to an image with different color organization when they have same textural behavior.

As discussed extending the co-occurrence for color image has a limitation. The co occurrence matrix for the color image would require matrix of order 16 million by 16 million. The feature extracted would require large amount of computation to be done and if the size of the database is substantial the feature extraction would take significant time. The solution to this could be, converting the image to gray scale and then evaluating the texture features but it looses the color features. Motivated by the limitations, extracting the texture feature of the image in grayscale and retaining the color feature in a different database is the proposed solution.

If the co-occurrence matrix is evaluated for the color images the number of intensity values are 256x256x256. And hence the computational power required for each image is not economical. So first converting the image to gray scale and then finding the co-occurrence matrix would generate the matrix of size 256x256. Also one matrix is calculated for each offset and angle. Usually the texture behavior cannot be reflected just by evaluating co-occurrence matrix only for one value. To properly extract the textural feature the co-occurrence matrix should be calculated at various values of offset and angles, so that the over all textural behavior can be reflected. The limitation to the above argument is calculating the co-occurrence matrix for how many values and creating database space for how many correlation property??. The decision taken for the value of offset and angle should be proper trade off between the precision of the extracted features and the amount of calculation required to evaluate the texture feature of the query image and time required to compare the images in database which may count in thousands.

6.2 COMPUTATIONAL MODEL FOR PROPOSED METHOD

The proposed method reduces the computational requirement to almost one fifth in extracting the textural feature. The experiments conducted in the lab shows that it is able to identify almost all types of texture. Once the similar texture images are extracted, as explained in chapter 5, the color similarity is now to be identified by the method explained in chapter 4 and that can be done through color feature extracted and stored in a separate database. The quantization in RGB space with reduction in resolution is done to save the space. Figure 6.1 shows the computational model for the color image retrieval system based on textural properties. The whole process can be divided into Color Feature Extraction and Texture Feature Extraction. The features of the database image are extracted offline and are saved in the color feature database or textural feature database. The online operation comprise of extracting the similar feature from the query image, by generating the query to get the similar value records from the database and then applying the matching algorithm to grade the similarity.

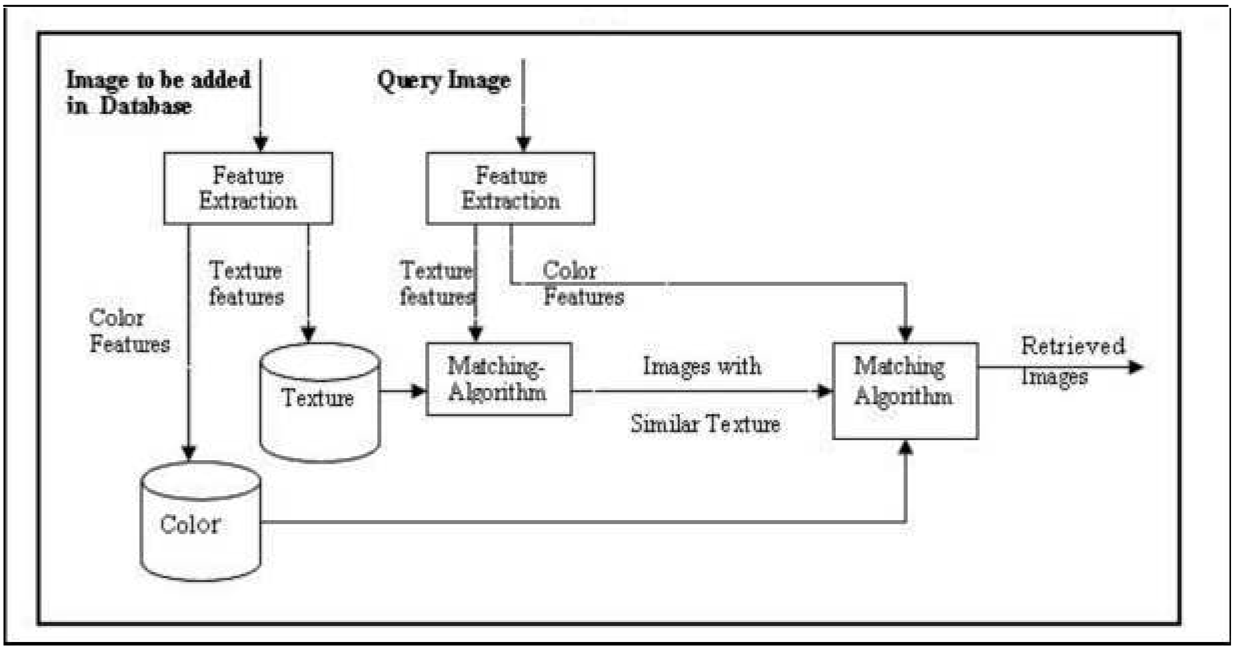


Figure 6.1 Computational model for combining color with texture

6.2 RETRIEVED IMAGES

As shown in figure 6.2 the images retrieved based on texture only can retrieve similar texture but result number 30 and 33 which looks quite similar are ranked far behind but they look very similar due to its color dexterity. The images clustering do not look appropriate. Figure 6.3 shows that the same images appear in number 2 and 3 and the results appear clustered which is more near to human perception.

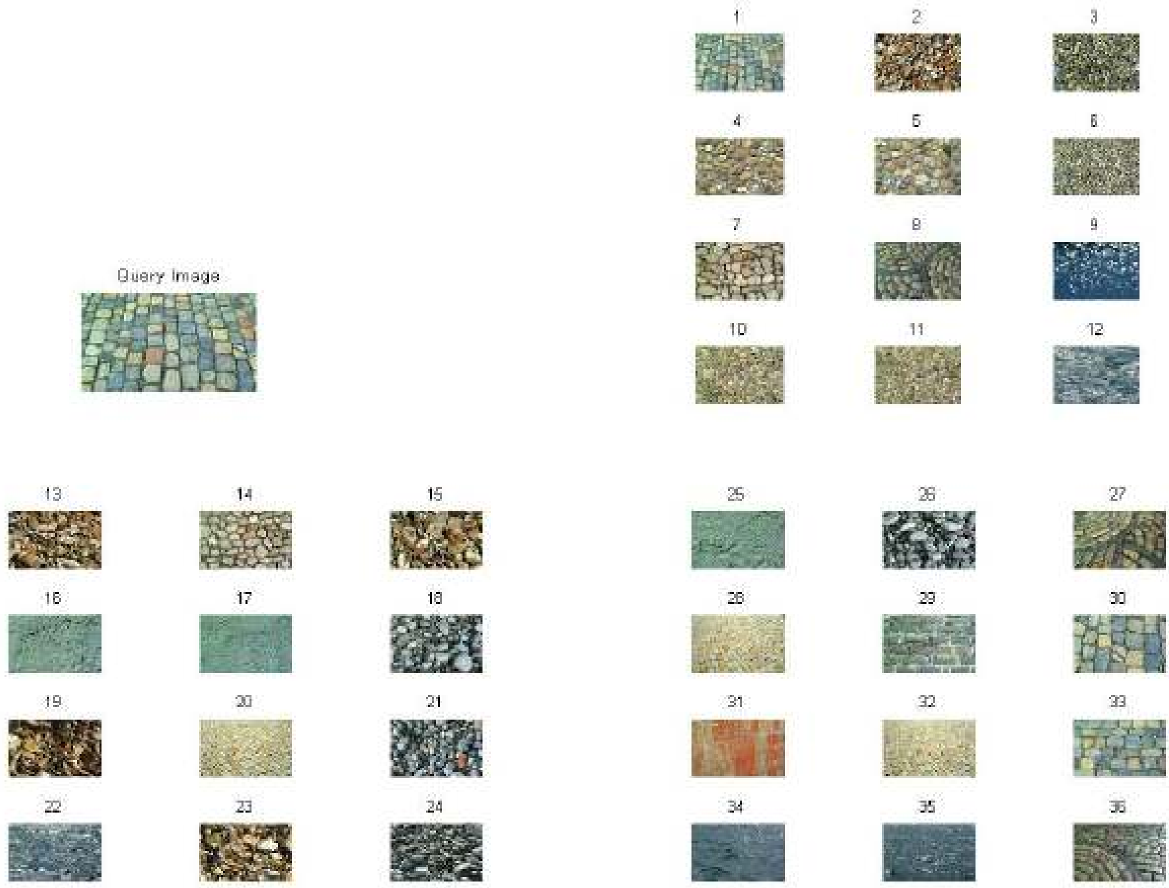


Figure 6.2 Image retrieved based on texture only

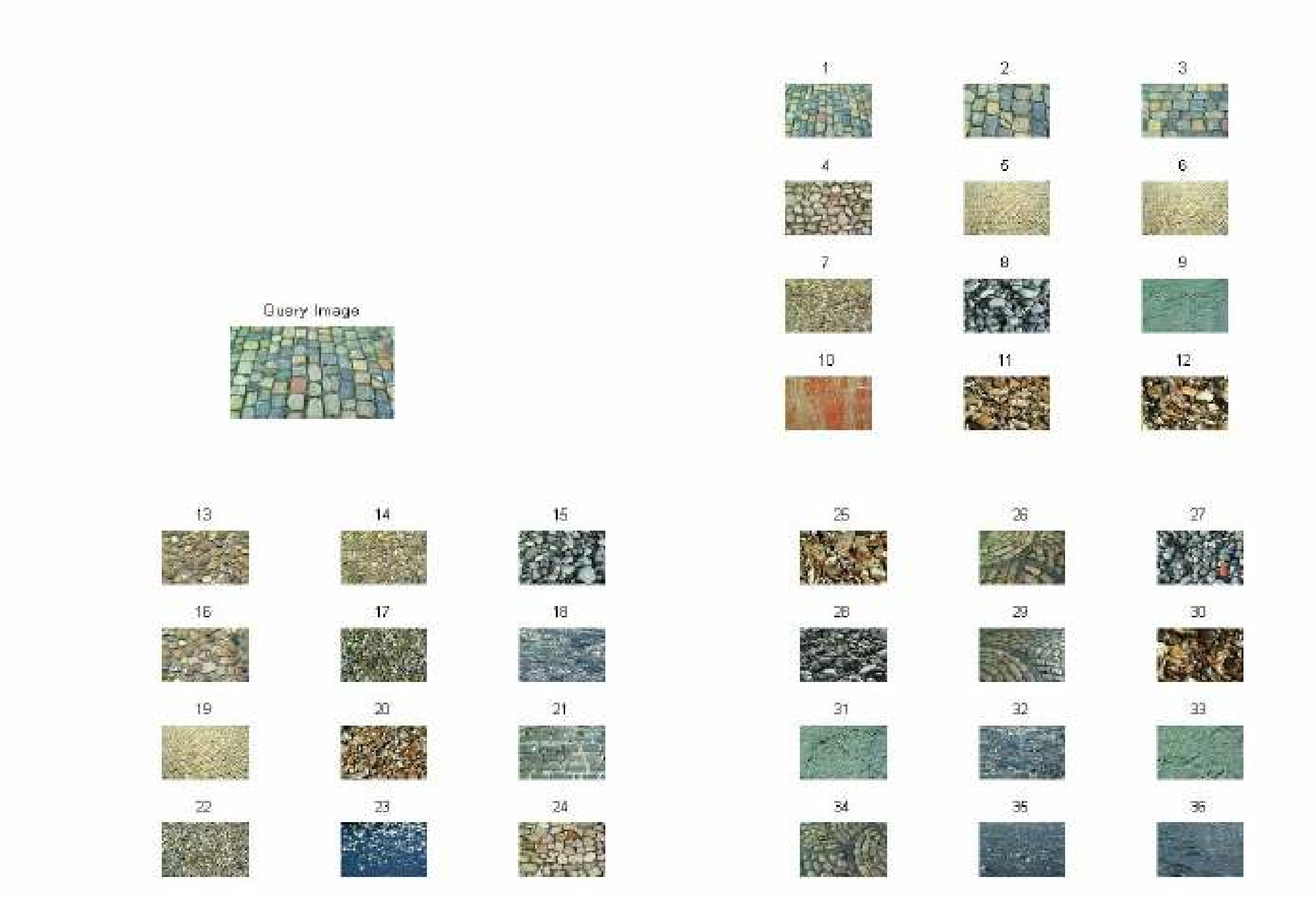


Figure 6.3 Image Retrieved when texture combined with color

7.1 SHAPE BASE FEATURE EXTRACTION

Shape is one of the major features in the images and various models to extract shape base feature is discussed in the section 2.4 .There are various 2-d models presented which are discussed in detail later in the chapter.

7.2 MODELS OF 2D SHAPE DESCRIPTORS

List of various shape descriptors for two dimension images is

1. Fourier Descriptor
2. Turning Functions
3. Bending energy function
4. Order Structure
5. Shape Thesaurus
6. Geometrical figure based search

7.2.1 Fourier Descriptor:-First of all the boundary points are to be calculated. The boundry point is found by eight connectivity method. Starting at an arbitrary point (x_0,y_0) and the remaining coordinate points .

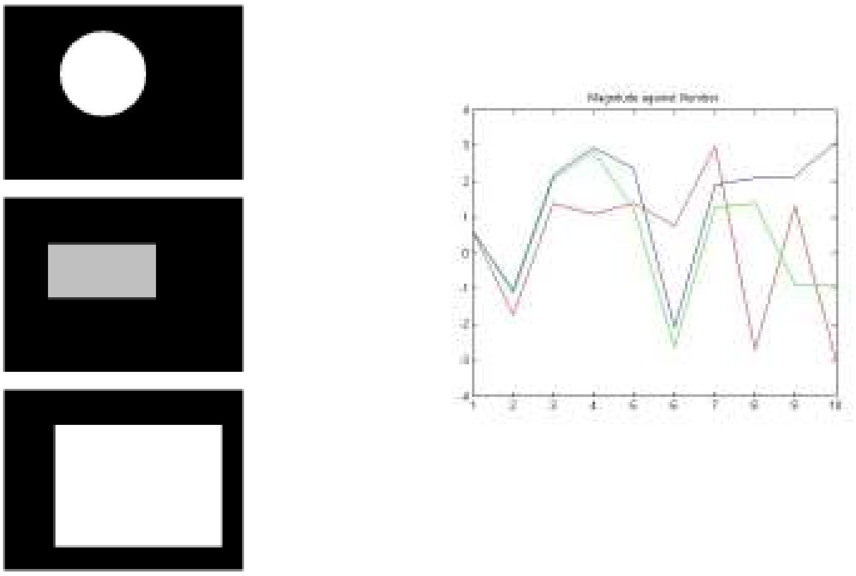


Figure 7.1 Fourier Descriptor

are (x_0,y_0), (x_1,y_1), (x_2,y_2)……… (x_{k-1},y_{k-1}) are encountered in traversing the boundary, say, in the counterclockwise direction. These coordinates can be expressed in the formx(k)= x_k and y(k) =$y_{k.}$ with this notation the boundary itself can be presented as the sequence of coordinates s(k)=[x(k),y(k)], for k=0, 1, 2,…., k-1. Moreover, each coordinate pair can be treated as a complex number so that

$$s(k) = x(k) + i\ y(k) \text{……………………….Equation 7.1}$$

for k=0,1,2,..,k-1. That is, the x-axis is treated as the real axis and the y-axis as the real axis and the y-axis as the imaginary axis of a sequence of complex numbers. Although the interpretation of the sequence was recast, the nature of the boundary itself was not changed. The discrete Fourier transform of s(k) is

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-i2\Pi uk/K} \text{………………………..Equation 7.2}$$

for u=0,1,2,….,k-1. the complex coefficient a(u) are called the Fourier descriptors of the boundary. The inverse Fourier of these coefficients restores s(k). That is

$$s(K) = \sum_{u=0}^{K-1} a(u)e^{i2\Pi uk/K} \text{………………………..Equation 7.3}$$

As can be seen from the figure 7.1 the coefficients of two rectangles and circle are plotted. But instead of two rectangles showing similar behavior shapes with similar size is showing similar behavior. As discussed earlier Fourier descriptor is good if we want to retain back the images. But they are not good descriptors if they are to be used for the comparison of the shapes.

7.2.2 Shape Distributions

- The key idea is to represent the signature of an object as a shape distribution sampled from a shape function measuring global geometric properties of an object.
- The primary motivation for this approach is to reduce the shape matching problem to the comparison of probability distributions, which is simpler than traditional shape matching methods.

The shape distribution are to be evaluated based on five functions described as below.

- ***A3:*** Measures the angle between three random points on the surface of a 3D model.
- ***D1:*** Measures the distance between a fixed point and one random point on the surface. We use the centroid of the boundary of the model as the fixed point.
- ***D2:*** Measures the distance between two random points on the surface.
- ***D3:*** Measures the square root of the area of the triangle between three random points on the surface.
- ***D4:*** Measures the cube root of the volume of the tetrahedron between four random points on the surface.

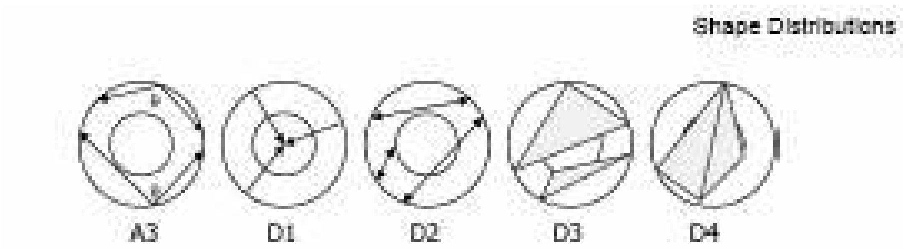
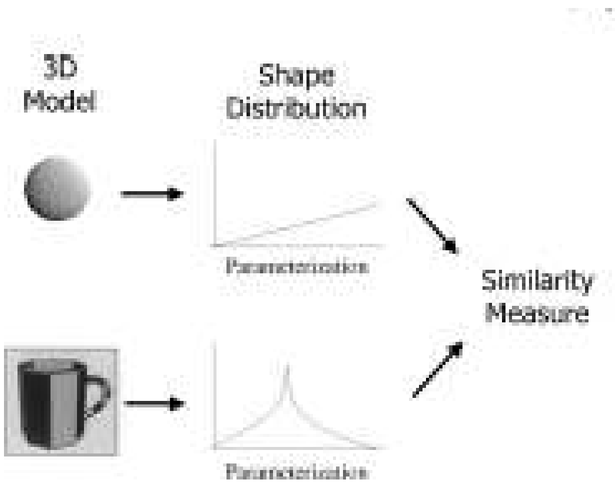


Figure 7.2 Shape Distribution

Hence shape distribution for the shapes look like



7.3 Shape distribution similarity measure

7.2.3 Geometrical Figure Based Search

This approach can identify shapes such as circle, ellipse ,triangle, quadrilateral, pentagon, hexagon, septagon and octagon. Other figures are assumed to be merging to a circle. The approach is to find the shape signature of the area. To do so the boundary points are to be evaluated. To begin with, the image is segmented first and then it is converted to logical image or black and white image. The left most white pixel is identified as the beginning of the trace boundary. Using eight connectivity the image is traced for boundary point. Now the distance of boundary point to the centroid is the shape signature. As shown in the figure 7.4 for shape signature.

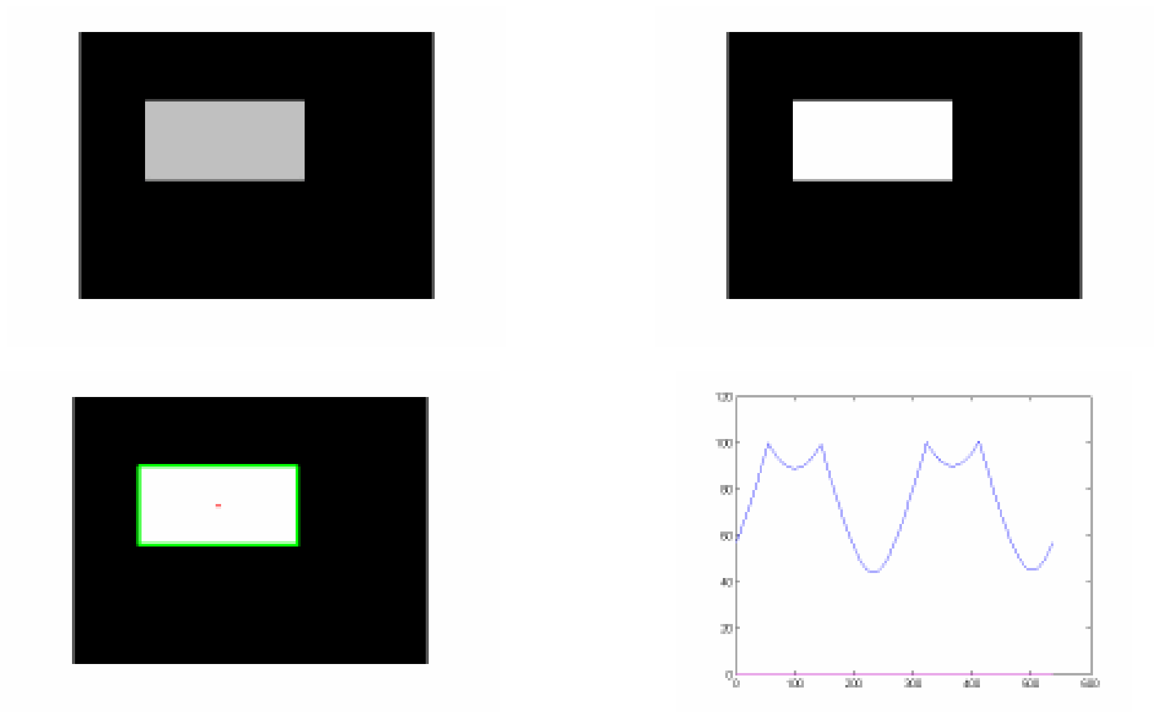


Figure 7.4 Shape signatures for the rectangle.

After observing various signatures of the shapes it was concluded that the number of peaks in the signature indicates the number of vertices in the geometric figure. Hence the problem of finding the number of vertices now reduces to finding the number of peaks or the number of maximas.

Finding Maxima

And hence in the distance plot now the number of maximas are to be calculated. To find the Maxima following steps are followed.

- Finding the derivative of the distance plot: the distance function differentiation is calculating by finding the difference between x(n+1) to x(n). the function is smoothened by finding the running sum of order 10.
- Finding the second derivative: the output of first step is then again derived to find the second derivative. To find the points when second derivative becomes zero ,we also need to consider non integer values.
- In second derivative find the zero crossing points: The zero crossing value is found by seeing in x(n) is positive and x(n+1) is negative or vise versa. The zero crossing points are then identified.
- Corresponding value in first derivative if negative then are identified as maxima: the corresponding value in first value are checked for negativity. All the zero crossing point for which they are also negative in first derivative is identified as maximas.

Figure 7.5 (a) Input Image (b) Boundary and centroid marked.

As Visible from the figure7.6 that irregularity in the signature graph can result in many local maxima. Figure 7.6(c) has many zero crossing and hence results in marking of local maxima. Hence for the smoothening of the curve running sum is applied which means replacing n^{th} value with previous n values and then finding the differenciated using equation 7.2

$$RSum(x) = \sum_{n=1}^{11} \mathrm{Dist}(x+n)$$ ....Equationion 7.4

$$Diff(x) = \frac{\mathrm{RSum(x) - RSum(x - d)}}{\mathrm{d}}$$ ....Equation 7.5

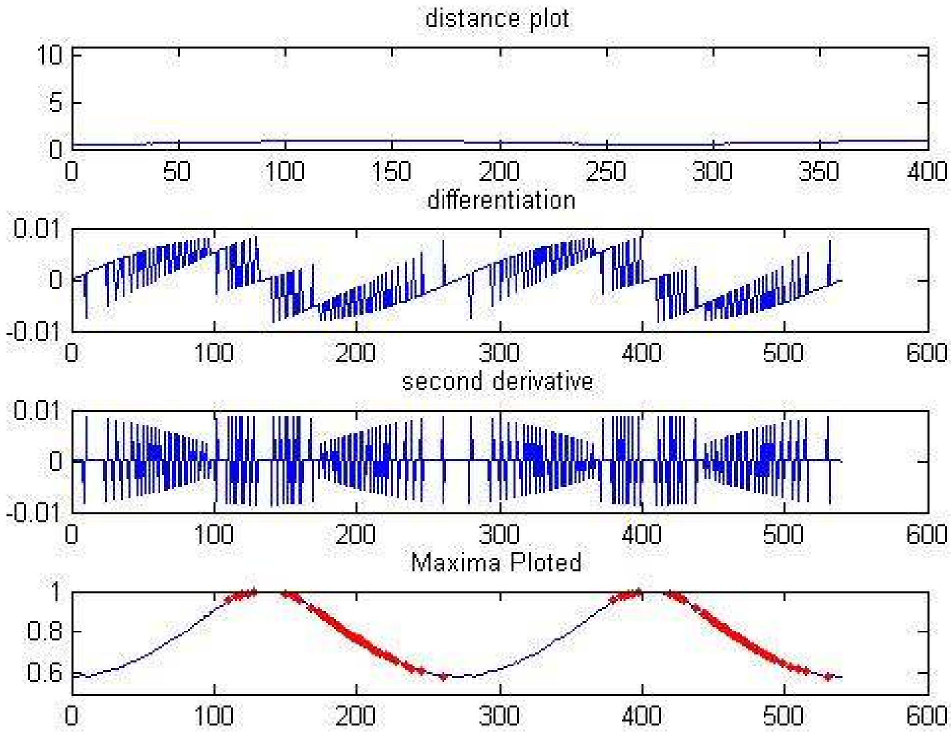


Figure 7.6 (a) Signature plot (b) First Derivative (c) Second Derivative (d) Maxima Plotted (Without running sum)

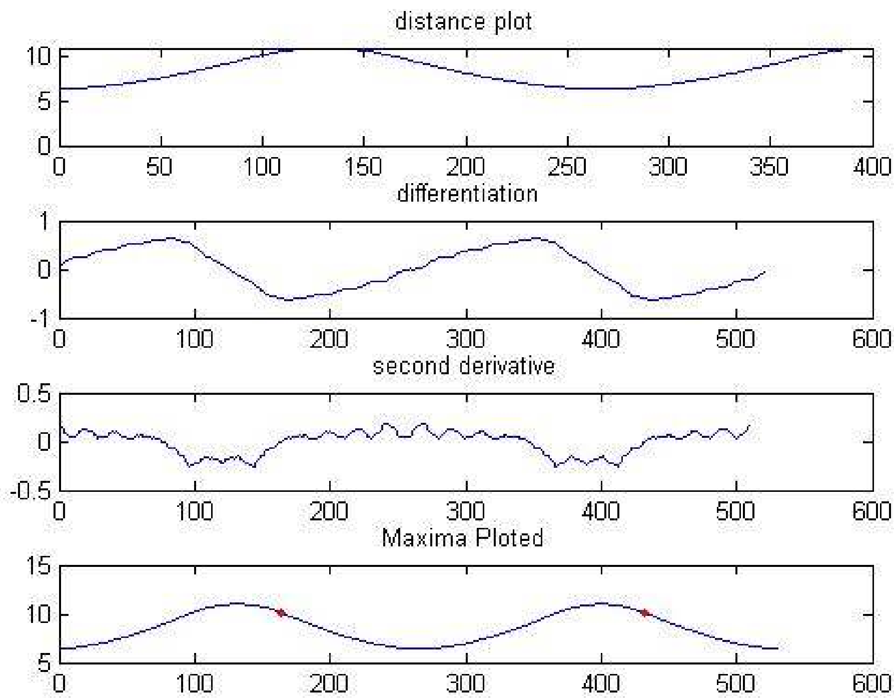
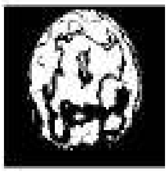


Figure 7.7 (a) Signature plot (b) First Derivative (c) Second Derivative (d) Maxima Plotted (With running sum)

The problem that may arise now is the segmentation of the area of interest. After trying technique of threshold to adaptive threshold to segment the image the output generated are as shown if figure 7.8

(a)



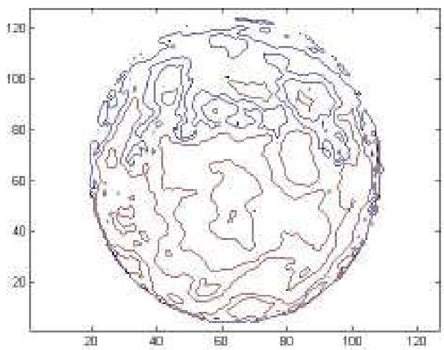
(b)

(c)

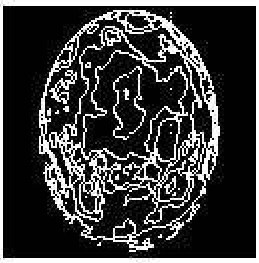
(d)

Figure7.8 (a),(c) Input Images (b),(d) Segmented images

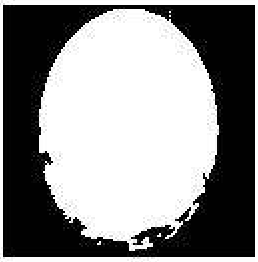
To segment the area of interest properly contour in the images are identified. Contour method of MATLAB gives coordinates of the contours in the images. Which are then transferred into the image to get the contour image the sequence of processing in the image 7.8(a) is shown in figure 7.9. Figure 7.9 (a) shows the contour coordinates. Figure 7.9(b) is the contour coordinates transferred into the matrix of input image size. Figure 7.9(c) is the filled image obtained by applying flood fill in the image 7.9(b). And 7.9(d) is boundary and centroid marked.

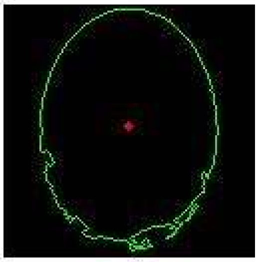


(a) Contour co ordinates of the input image



(b) Contour points in Image



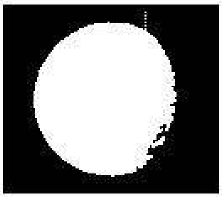


(c)Flood filled image (d) Boundary and Centroid Marked

Figure 7.9(a), (b), (c), (d)

If the image is segmented properly in all the cases the number of peaks is equal to the number of edges in the shape.

In the case of Ellipse the number of peaks obtained is two. In the case of circle when there is no edges it is observed that the distance plot show a straight line or variation nor more that two pixels.



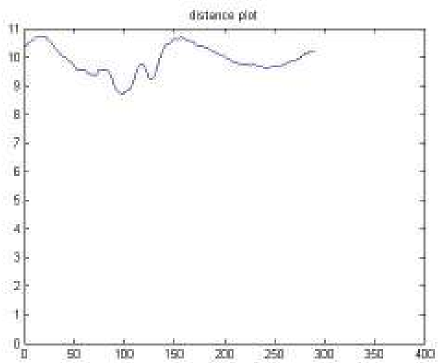


Figure 7.10 Distance plot for Circle

8 RESULTS AND CONCLUSION

8.1 IMAGE RETRIEVAL BASED ON COLOR

In color base image retrieval a query image is to be selected first, the database contains color feature in the form of index matrix and index key as discussed in chapter four. The color features (Index matrix and Index key) are extracted from the query image. Query is made based on index key, and then Euclidian distance is calculated between query image and others.

8.1.1 Index Matrix:

This is obtained by quantizing the color into 16 levels and dividing image into grid of 5x5 Pixel each.

.

Figure 8.1 Index Matrix Representation of Images (From 125X83 to 25X16)

8.1.2 Image Retrieval Based On Color Only Given A Query Image

Figure 8.2 shows the retrieved images based on the query image.1^{st} result image is exactly the same image and second and third images are very similar to it.

Figure 8.2 Image retrieval based on color given a query image

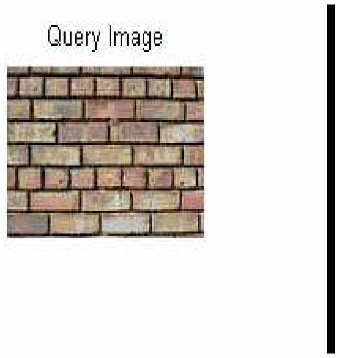
8.1.3 Bad Result

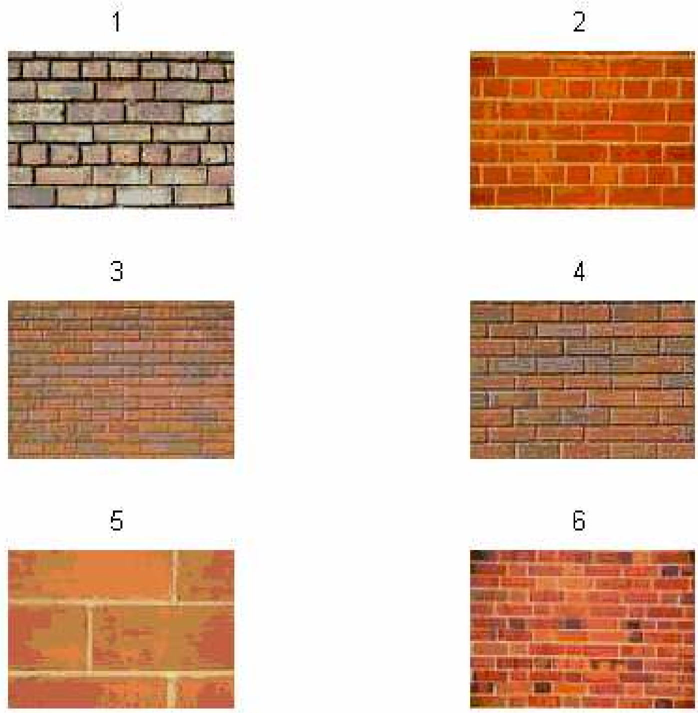
Color quantization some times may be responsible for unsatisfactory results due to change in brightness.

Figure 8.3 Index matrix of similar image is very Different

8.2 TEXURE BASED IMAGE RETRIEVAL

The texture features are the numbers, ie correlation of co occurrence matrix at various off set. Query image is selected and similar feature is extracted from it. The images from the database is retrieved and sorted in the order of their similarity. Out of various approaches discussed in chapter 5 Approach 3 is the one used in the System. The image Database is combination of bricks, pebbles, paint water and clouds. When given brick as the query image the results obtained at the higher positions are brick as in figure 8.3 . For all other textures the results obtained are very good.




1
2
3
4
5
6

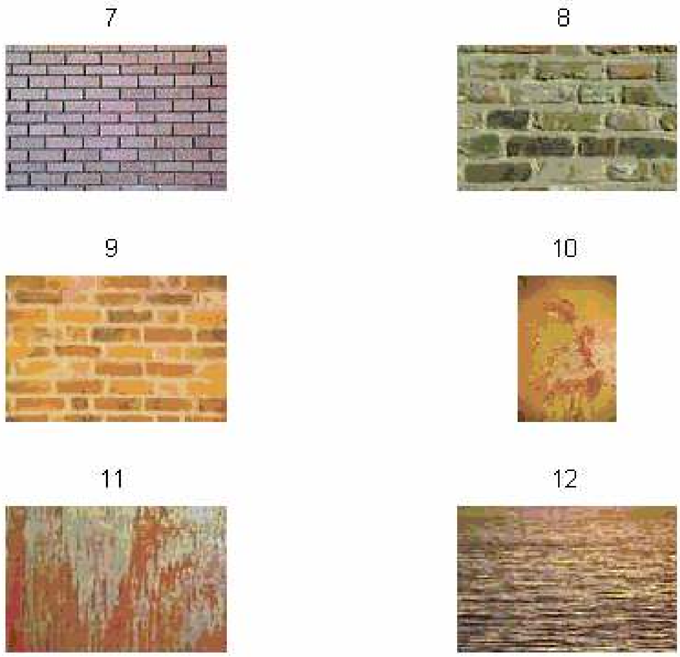

7
8
9
10
11
12

Figure 8.4 Texture based Image retrieval

8.3 COMBINING COLOR WITH TEXTURE

Chapter 6 discusses that how combining color with texture improves the results to a great extend, and bridges the gap between the human perception and features extracted.

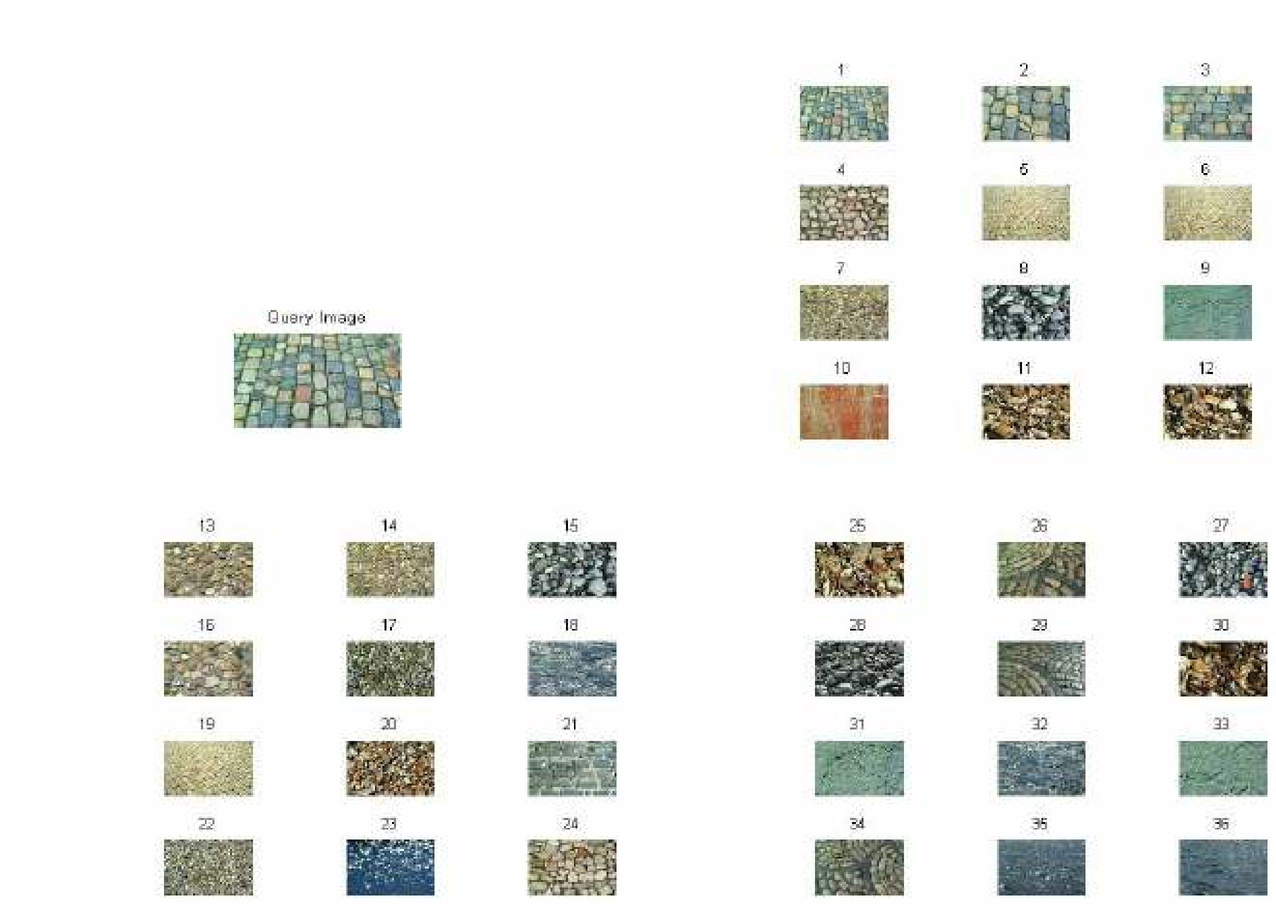


Figure 8.4 Image retrieval Based on Color and Texture

8.4 SHAPE FEATURE EXTRACTION

The geometric shapes can be identified and the database for the geometric shapes are separated from the remaining database.

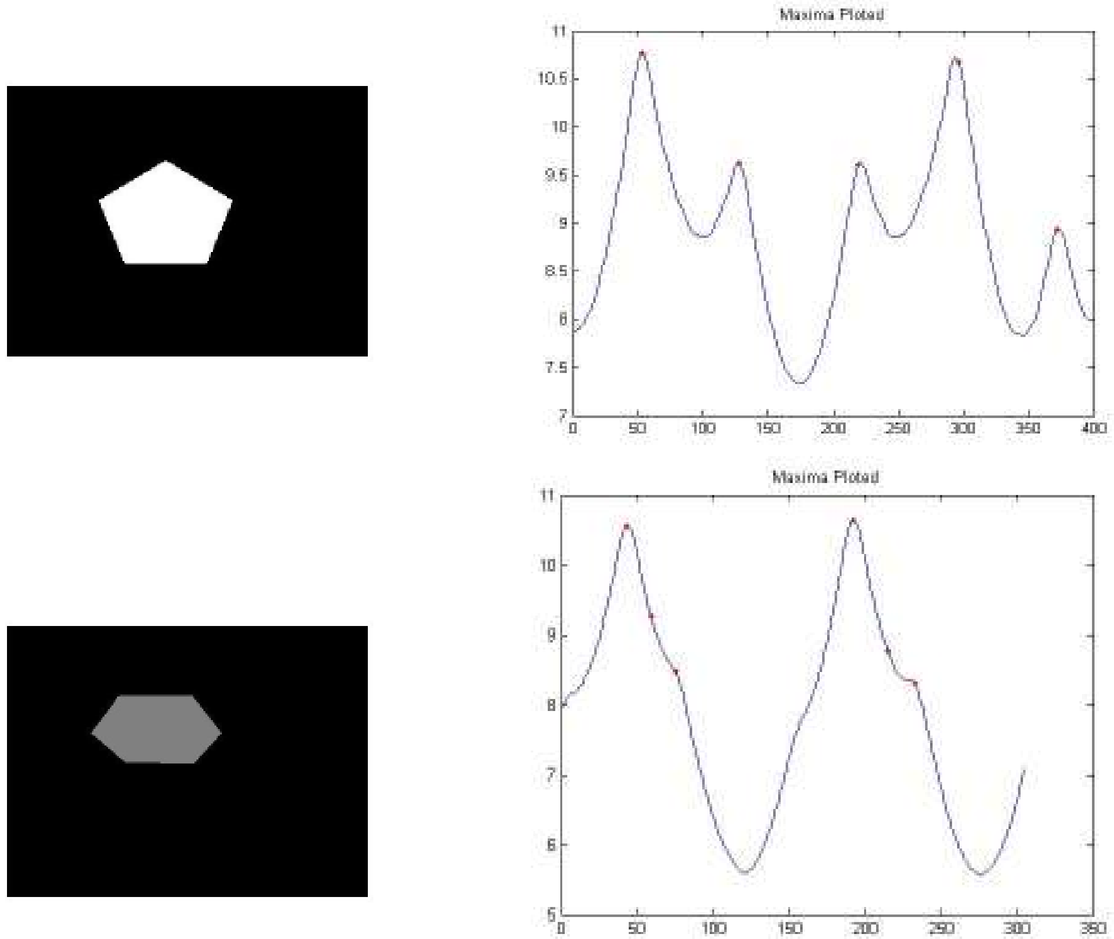


Figure 8.5 Output for Pentagon and Hexagon

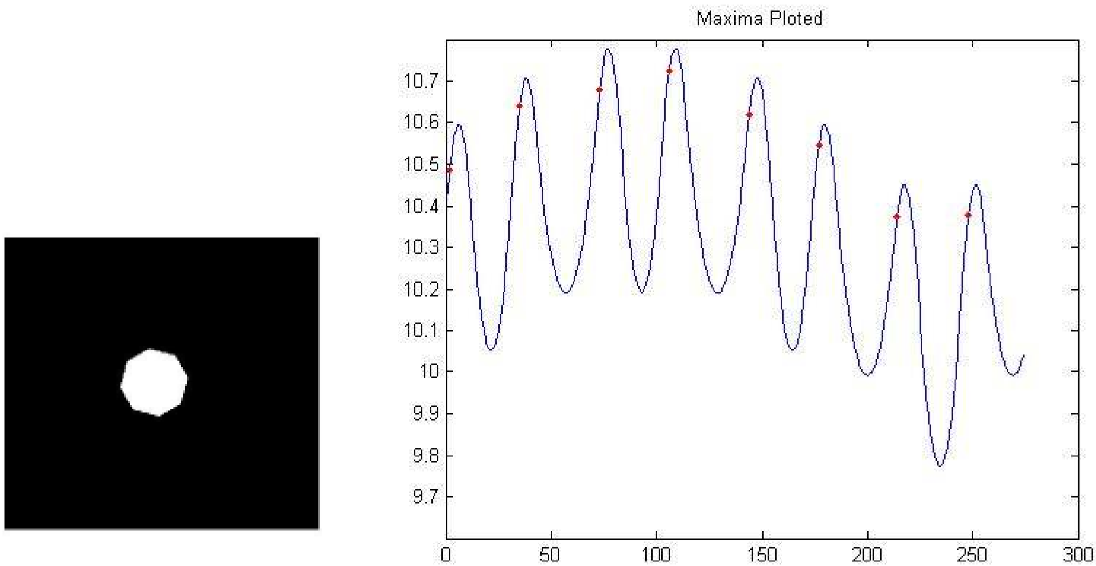


Figure 8.6 Output for Octagon

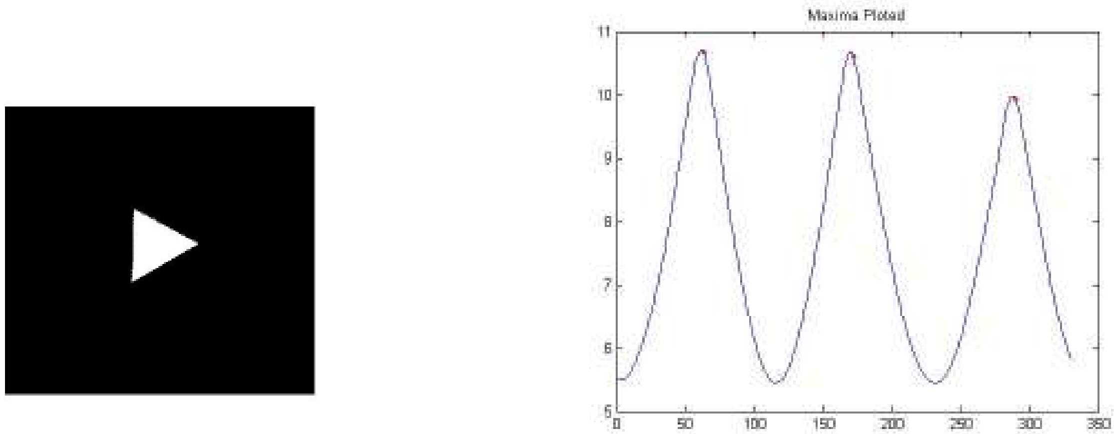


Figure 8.7 Outputs for Triangle

8.5 CONCLUSION

The experiments were conducted with the database of images, divided into two database shape images and other images. The color based image retrieval shows very good results. And in all the cases the results obtained were very encouraging. The texture features are measure of uniformity, energy, homogeneity and contrast. Hence the images retrieved are similar in texture but a gap is observed in human observation as color is not included in the comparison. The combining of color feature with texture helped in result improvement to a great extent. The shape based feature although can identify only geometric shapes, but efficiency of identification of geometric figures is 100% if the segmentation results in proper area of interest.

8.6 FUTURE WORK

- Among the mentioned modules color and texture based feature extraction are giving acceptably good results, but shape based features can still be worked upon. The domain of shape identification and comparison can be extended.
- The content based retrieval can be extended for the movies along with the database of images. More optimization would be required as each movie clip may contain n number of frames (images) to be analyzed.
- The system can be made specific to retrieve images from a domain restricted database, for example image retrieval system for satellite images. The feature descriptors have to be specific for capturing typical features from such images.

References

1) James Z. Wang, Jia Li, and Gio Wiederhold ,*SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries* , IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 23, No. 9, September 2001

2) A. Pentland, R. W. Picard, and S. Sclaroff, "*Photobook: Tools for content- base manipulation of image databases,*" presented at SPIE Conference on Storage and Retrieval of Image and Video Databases II, San Jose, CA, 1994.

3) S. Sclaroff, L. Taycher, and M. LaCascia, "*ImageRover: A content-based image browser for thr World Wide Web,*" presented at IEEE Content-Based access of Image and Video Libraries, 1997.

4) J. R. Smith and S.-F. Chang, "*Searching for images and video on the World Wide Web,*" *Multimedia Systems*, vol. 3, pp. 3-14, 1995.

5) P. Suda, C. Bridoux, B. K¨ammerer, and G. Maderlechner, *Logo and word matching using a general approach to signal registration*. In *Proc. ICDAR '97*, pages 61–65, Ulm, Germany, Aug. 1997.

6) Irena Valova, B. Rachev I., *Image organization, querying and retrieval by color distribution features*. International Conference CompSysTech'2002, Sofia, Bulgaria. (2002) 20–22

7) Irena Valova, B. Rachev, *Image Databases - An Approach for Image segmentation Color reduction Analysis Synthesis*, International Conference CompSysTech 2003, Sofia, Bulgaria. (2003) 19–20

8) Robert M. Hralick: *Statistical and Structural Approaches to texture,* proceedings of IEEE .67.(1979) 786–805 8

9) Irena Valova, B. Rachev, Silyan Arsov , *An Approach For Image Organization and Retrieval In Realistic Image Databases*, 7th GIS Workshop Potsdam, Germany.(2001)

10) Moses Amadasun and Robert King , *Texural Features Corresponding to Texural Properties* .IEEE Transaction on system, Man and Cybernatics .19 .(1989)

11) Pranam Janney, Sridhar G, Sridhar V.,*Enhancing capabilities of Texture Extraction for Color Image Retrieval*, Transaction on Engineering, Computing and Technology 5.(2005)

12) Gupta A., Jain R., *Fast image retrieval using color-spatial information The VLDB Journal*. Visual Information Retrieval, Communications of the ACM .40. (1997).

13) Abbadini Noureddine, *Multiple representations, similarity matching, and results fusion for content-based image retrivel*. Multimedia systems (Multimedia syst.)10(2005)444–456

14) Andre Folkers, Hanan Samet, *Content-based Image Retrieval Using Fourier Descriptors on a Logo Database*, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 3 p. 30521

15) Robert Osada, Thomas Funkhouser, Bernard Chazelle , David Dobkin *,Matching 3D Models with Shape Distributions*, Proceedings of the Interntional Conference on Shape Modeling & Applications**,** 154 , 2001

16) Lars-Jacob Hove , *Extending Image Retrieval Systems with a Thesaurus for Shapes,* Master Thesis , Institute for Information and Media Sciences University of Bergen, 2004

17) C. Zhang and T. Chen. *Efficient feature extraction for 2D/3D objects in mesh representation.* In *CIP 2001*, 2001.

18) J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. Macleod. *Coarse filters for shape matching. IEEE Computer Graphics and Applications*, 22(3):65–74, 2002.

19) C. Zhang and T. Chen. *Indexing and retrieval of 3D models aided by active learning.* In *ACM Multimedia*, pages 615–616, 2001.

20) M. Elad, A. Tal, and S. Ar. *Content based retrieval of VRML objects - an iterative and interactive approach*. In *Eurographics Multimedia Workshop*, pages 97–108, 2001.

21) R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. *Shape distributions. ACM Transactions on Graphics*, 21(4):807–832, 2002.

22) R. Ohbuchi, T. Otagiri, M. Ibato, and T. Takei. *Shape-similarity search of three-dimensional models using parameterized statistics.* In *Pacific Graphics 2002*, pages 265–274, 2002.

23) C. Y. Ip, D. Lapadat, L. Sieger, and W. C. Regli. *Using shape distributions to compare solid models.* In *Solid Modeling '02*, pages 273–280, 2002.

24) D. V. Vrani´c, D. Saupe, and J. Richter. *Tools for 3D-object retrieval Karhunen-Loeve-transform and spherical harmomics.* In *Proc. IEEE 2001 Workshop on Multimedia Signal Processing*, 2001.

25) Ido Omer and Michael Werman, Image Specific Feature Similarities

Appendix A

Database Management and Connectivity TO MATLAB

The database used is PostgreSQL and the system architecture is explained. Understanding how the parts of PostgreSQL interact will make this chapter somewhat clearer. In database family, PostgreSQL uses a client/server model. A PostgreSQL session consists of the following cooperating processes (programs)

- A server process, which manages the database files, accepts connections to the database from client applications, and performs actions on the database on behalf of the clients. The database server program is called postmaster.

- The user's client (front-end) application that wants to perform database operations. Client applications can be very diverse in nature: a client could be a text-oriented tool, a graphical application, a web server that accesses the database to display web pages, or a specialized database maintenance tool. Some client applications are supplied with the PostgreSQL distribution; most are developed by users. As is typical of client/server applications, the client and the server can be on different hosts. In that case they communicate over a TCP/IP network connection.

- The PostgreSQL server can handle multiple concurrent connections from clients. For that purpose it starts ("forks") a new process for each connection. From that point on, the client and the new server process communicate without intervention by the original postmaster process. Thus, the postmaster is always running, waiting for client connections, whereas client and associated server process come and go. (All of this is of course invisible to the user. We only mention it here for completeness.)

The Data types supported are as in table A.1. The Array of almost all data types is supported which is very useful in storing the color feature like index matrix and index value, in a single column.

Name	Aliases	Description
bigint	int8	signed eight-byte integer
bigserial	serial8	autoincrementing eight-byte integer
bit [(*n*)]		fixed-length bit string
bit varying [(*n*)]	varbit	variable-length bit string
boolean	bool	logical Boolean (true/false)
box		rectangular box in the plane
bytea		binary data ("byte array")
character varying [(*n*)]	varchar [(*n*)]	variable-length character string
character [(*n*)]	char [(*n*)]	fixed-length character string
cidr		IPv4 or IPv6 network address
circle		circle in the plane
date		calendar date (year, month, day)
double precision	float8	double precision floating-point number
inet		IPv4 or IPv6 host address
integer	int, int4	signed four-byte integer
interval [(*p*)]		time span
line		infinite line in the plane
lseg		line segment in the plane
macaddr		MAC address
money		currency amount
numeric [(*p*, *s*)]	decimal [(*p*, *s*)]	exact numeric of selectable precision
path		geometric path in the plane
point		geometric point in the plane
polygon		closed geometric path in the plane

Table A.1 Data types in Postgre SQL

Connectivity to PostgreSQL:- The initial Process of data base creation is to be done when we can create a database space as in any other DBMS. The tables can be created at the console by the command as

```
CREATE TABLE color feature      (image name        char [],
                                 Index key          integer [ ],
                                 Index matrix       text [ ] [ ],
                                 );
```

after the tables are created at the PostgreSQL console now it is ready to be operated at the front end (MATLAB). The ODBC connection is established by first making a system DSN(Data Source Name) using proper username and password. Using the connection name now MATLAB database toll box has functions to execute the queries at PostgreSQL

1) Establishing Connection

conn = database('datasourcename', 'username', 'password')

connects a MATLAB session to a database via an ODBC driver, returning the connection object to conn. The data source to which you are connecting is datasourcename.

2) Inserting Data

Insert (conn,’table_name’,colnames,data_for_colnames);

This inserts the data in the ‘data_for_colnames’ in the columns specified in ‘colnames’ of the table ‘table_name’ against conn.

3) Retrieving Data

curs = exec(conn, 'sqlquery')

Executes the valid SQL statement sqlquery, against the database connection conn, and opens a cursor. Running exec returns the cursor object to the variable curs, and returns information about the cursor object. The data is to be fetched using fetch command.

ret data=fetch(curs);

This return an object to ret_data and data is in ‘data’ fields of the object. That can be accessed by

Info= ret_data.data

ret_data. data is the grid of cell of order no_of column to number of rows retrieved. The data can be accessed as info (i, j) to access data at i^{th} row and j^{th} column.

Appendix B

Paper submitted at '*Second International Conference on Pattern Recognition and Machine Intelligence (PReMI'07)'* Organized by Machine Intelligence Unit Indian Statistical Institute, Kolkata, India. Title of the paper '**Retrieval of Color Images Based on Textural Property'.**

Acceptance notification is due by 31st July 2007.