A Study on Content Based Image Retrieval Techniques

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Pranabjyoti Dutta
Registration No: 002 of 2007

School of Engineering
Department of Computer Science and Engineering
Tezpur University
June, 2007

TEZPUR UNIVERSITY

This is to certify that the thesis entitled *A Study on Content Based Image Retrieval Techniques* submitted by *Mr. Pranabjyoti Dutta* to Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment for the award of the Degree of Doctor of Philosophy in *Computer Science* has been examined by us on Dec 2, 2008 and found to be satisfactory.

The Committee recommends for the award of the degree of Doctor of Philosophy.

Signature of:

M. Dutta
Principal Supervisor

P Gupta 2/12
External Examiner

Associate Supervisor

Co- Supervisor

Date: 2.12.08

TEZPUR UNIVERSITY

This is to certify that the thesis entitled *A Study on Content Based Image Retrieval Techniques* submitted to Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfilment for the award of the Degree of Doctor of Philosophy in *Computer Science* is a record of research work carried out by *Mr. Pranabjyoti Dutta* under my personal supervision and guidance.

All helps received by him from various sources have been duly acknowledged.

No part of this thesis has been reproduced elsewhere for award of any other degree.

M. Dutta

Date: 4.8.08

Place: Tezpur

Signature of Principal Supervisor
Designation: Professor
School: Engineering
Department: Computer Science and Engineering.

Abstract

The explosive growth of multimedia data, such as images, has led to a strong demand for effective access to large image collections. Content Based Image Retrieval (CBIR) is a technique to solve this problem. Digitized images consist purely of arrays of pixel intensities, with no individual meaning. In CBIR, useful information has to be extracted from this raw data (such as recognizing the presence of a particular shape or texture) and retrieval has to be done based on the similarity measure of this raw data but not by using assigned keywords to the images. Thus it is a challenging task to develop a fully automatic CBIR system that operates on the raw data of an image. Various CBIR systems have been discussed and none of them has been found to be fully automatic.

In our work an attempt has been made to explore the possibility of designing a fully automatic CBIRSs for a specific range of application domains. The underlying concept is that, images consist of color clusters and combination of these color clusters forms objects of interest within the image. A color cluster consists of a single color and hence those pixels that form the cluster will have a similar intensity values. Hence, those pixels that have similar intensity values and spatially connected can be grouped together to form a cluster. A single or combination of such color clusters may form an object. A cluster or an object can be represented by a single or a few numerical value or values using the pixel intensities which will form the index of the cluster or the object. Thus, a number of cluster or object indices will represent the index of the image which will help in the similarity measurement between images.

To achieve this, first we have quantized the image to smooth the color content of the image. After that we have employed a clustering algorithm to find out the color clusters of an image. Once the color clusters are obtained, objects within the image are identified. These color clusters and objects are then indexed using a suitable translation, rotation and scale invariant indices. These indices are then stored in a suitable database structure. Matching engine has been developed which searches in the database structures for similarity measurement and it has the facility for global and regional search.

While querying, an input image is given to the system. The system automatically calculates the clusters present in the image. Once the clusters are identified, the interesting objects in the image are also determined. Now search can be initiated based on particular objects present in the image (regional search) or all the objects present in the image (global search). The matching engine will return a set of most relevant images for the said query. The proposed retrieval systems were evaluated in light of several real-life datasets, and found to perform satisfactory.

Declaration

I hereby declare that the thesis entitled *A Study on Content Based Image Retrieval Techniques* submitted to the Department of Computer Science and Engineering, Tezpur University has been carried out by me and was not submitted to any other institution for award of any other degree.

Date: 4.8.08

Place: Tezpur

Pranabjyoti Dutta

Acknowledgment

It is my great pleasure to thank my thesis supervisor Prof. Malayananda Dutta for his help, encouragement and support throughout the course of this work. Prof. Dutta first introduced me to the interesting problem and motivated me to work on it, for which I am grateful to him. I specially thank Prof. Dhruba Kumar Bhattacharryya for his constant inspiration and immense help during the whole of my research tenure till submission of my thesis. I do hereby duly acknowledge the assistance and constant encouragement received from Prof. Jugal K Kalita Department of Computer Science, University of Colorado at Colorado Springs, USA, during my Ph. D. work. I would also like to thank Dr. Rajib Kumar Das, Calcutta University, who has helped me with his valuable suggestions from time to time while devising the various algorithms during my work. I also thank Prof. D. K. Saikia, who constantly encouraged me and provided the necessary facilities during the work and awaited eagerly, along with me, for the completion of the work I am very much thankful to those reviewers whose critical comments on three of my papers helped a lot in writing the thesis. I thank my friends and colleagues in Tezpur University for their help and encouragement. I also thank three of my student friends who have taken the pain to develop the prototype in java as their project work. I also convey my gratitude to the Internet for providing me the necessary information to carry out my research. Finally, I thank my family members for bearing with my whims and for shouldering my social responsibilities during the tenure, to make me free.

Pranabjyoti Dutta

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Motivation for the Research

Traditional image retrieval approach is based on manual image indexing where human indexers assign keywords to images. Relevant images can be retrieved by using the indexed keywords as queries. However, there are limitations of manual indexing. For example, it is very time consuming and expensive, especially when the size of the image collection is very large. In addition, different indexers may assign different keywords to the same image, or even the same indexers perform differently at various circumstances and times. For retrieval, users may not be aware of or agree with the indexed keywords or terms, making the retrieval results unsatisfactory. Advances in computer and multimedia technologies allow the production of digital images and large repositories of images. Therefore, designing automated image retrieval systems which can operate on a large scale is necessary. The goal is to create, manage, and query image databases in an efficient and effective manner.

Content Based Image Retrieval (CBIR) also known as query by image content is the application of computer vision to the image retrieval problem i.e. the problem of searching of digital images in large databases. CBIR, proposed in the early 1990s, is a technique to automatically index images by extracting their (low-level) visual features, such as color, texture, and shape, and the retrieval of images is based solely upon the indexed image features. Typically, images are represented as points in a high dimensional feature space. Then, a metric is used to measure *dis/similarity* between images on this space. Images *similar* to the query are retrieved.

1.2 The Aim and Objectives

The aim of this research work is to develop a Content Based Image Retrieval System based on the spatial layout of the color content of an image. For example, a colored image can be viewed as a distribution of colored pixels in a 2-D plane which is basically a projection of a 3D object. This distribution of colored pixels forms *color clusters* of arbitrary shape within the image. A color cluster can be viewed as a data set having a color and a position. An image is usually formed by a number of constituent objects. Each such object can be characterized by a single color cluster or a combination of several color clusters. Figure 1.1 shows an image with four color clusters marked by cluster numbers 1, 2, 3 and 4. In addition there are five distinct objects, viz., a tree constituted of cluster numbers 1 and 2, the background constituted of cluster number 3, the earth constituted of cluster number 4, the leaf constituted of cluster number 1 and the trunk of the tree constituted of cluster number 2. An index for the image can be automatically generated based on the above clusters and objects.

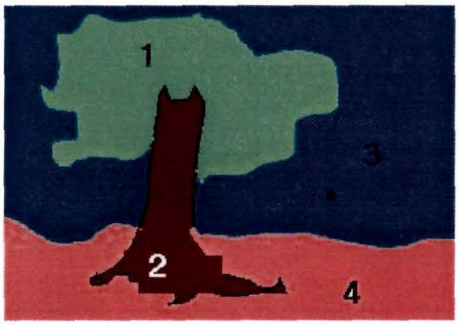


Figure 1.1. An example image having four clusters

The objectives of this research are as follows:

- Develop a fast clustering technique that can extract color clusters of any shape.
- Develop a prototype to identify the important color clusters or objects of an image.
- Develop a translation, scale and rotation invariant index for each objects or clusters identified. The dimensionality of the index has to be kept as small as it can be for an Image such that no *dimensionality reduction* module is necessary.
- Develop a index structure which will facilitate quicker search operation and
- Develop a matching engine for similarity search.

1.3 The Hypothesis and Novelty of the Work

1.3.1 The Hypothesis

The main hypothesis is to claim that proper segmentation of the color features of an image will help in identifying objects underlying the image. Once the color segments and the objects present in the image are identified, they can be indexed based on the spatial layout. When the image can be indexed by some indexing structure, searching will become easier.

1.3.2 The Originality of the Work

The originality of this thesis is the introduction of a new image segmentation/clustering approach (BOO-Clustering) of distribution of colored pixels within an image. Objects are identified from the color clusters present in the image by retaining only the interesting ones based on the percentage frequency of occurrence dynamically. Silhouette moments of second order are used for index calculation of the objects. These indices are stored in a variant of B-tree structure for indexing purpose. Appropriate matching engine has been devised for similarity retrieval of images.

1.4 Scope of our work

Our method of indexing and retrieval of images takes query by example approach i.e. an input query image is submitted to the system for index generation and retrieval process. The system generates all color clusters and all combinations of color clusters (that form the objects) present in the image automatically and silhouette moments are calculated. These set of silhouette moments are used for indexing and retrieval.

Occlusion, minimal difference in background and foreground colors as well as angle in which an image is acquired may lead to drastic change in shape of objects in the image. Under such circumstances silhouette moments based on color clusters may be inadequate for indexing and retrieval. Human computer interaction (HCI) may be necessary to gather knowledge about objects in the image. This requires further investigation and is beyond the scope of this thesis.

Image Database used for the experiments

As discussed in Scheme I and II, for our experiments we have used Cohn-Kanade Facial Expressions and other real world and synthetic image databases. They include *facial expressions, scenery, animals, cars, flowers* etc. For example, the facial expression database consists of nearly 5000 images of different persons and for each person there are nearly 30 different photographs having different facial expressions. All these images have been taken at a particular direction (facing directly) and at a fixed distance from the camera having the same background color. These facial images are of same aspect ratio. We have analyzed our proposals using standard measures of image retrieval viz. precision and recall (Section 2.5).

1.5 Thesis Outline

The reminder of this thesis is organized as follows:

Chapter 2 surveys the state-of-the-art image retrieval (CBIR) techniques and some Content Based Image Retrieval systems which are currently in use.

Chapter 3 reports the background and related concepts of the work.

Chapter 4 reports some segmentation/clustering algorithms currently in use.

Chapter 5 presents a clustering algorithm.

Chapter 6 presents the scheme I architecture and its retrieval performances.

Chapter 7 reports an enhanced version of scheme I reported in the preceding two chapters.

Chapter 8 gives the concluding remarks and also shows the future research direction

Chapter 2

Related Works

2.1 Introduction

Content-based image retrieval, a technique which uses visual contents to search images from large scale image databases according to user's interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines.

Before introducing the fundamental theory of content-based retrieval, we will take a brief look at its development. Early work on image retrieval can be traced back to the late 1970s. In 1979, a conference on Database Techniques for Pictorial Applications [ABD79] was held in Florence. Since then, the application potential of image database management techniques has attracted the attention of researchers [NSC79, NSC80, SKC81, SKC88]. Early techniques were not generally based on visual features but on the textual annotation of images. In other words, images were first annotated with text and then searched using a text-based approach from traditional database management systems. Comprehensive surveys of early *text-based image retrieval* methods can be found in [SKC92, HTN84]. Text-based image retrieval uses traditional database techniques to manage images. Through text descriptions, images can be organized by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard Boolean queries. However, since automatically generating descriptive texts for a wide spectrum of images is not feasible, most text-based image retrieval systems require manual annotation of images. Obviously, annotating images manually is a cumbersome and expensive task for large image databases, and is often subjective, context-sensitive and incomplete. As a result, it is difficult for the traditional text-based methods to support a variety of task-dependent queries.

In the early 1990s, as a result of advances in the Internet and new digital image sensor technologies, the volume of digital images produced by scientific, educational, medical, industrial, and other applications available to users increased dramatically. The difficulties faced by text-based retrieval became more and more severe. The efficient management of the rapidly expanding visual information became an urgent problem. This need formed the driving force behind the emergence of content-based image retrieval techniques. In 1992, the National Science Foundation of the United States organized a workshop on visual information management systems [RJP92] to identify new directions in image database management systems. It was widely recognized that a more efficient and intuitive way to represent and index visual information would be based on properties that are inherent in the images themselves. Researchers from the communities of computer vision, database management, human-computer interface, and information retrieval were attracted to this field. Since then, research on content-based image retrieval has developed rapidly [AEC93, JDC93, CFE94, YGH94, RJP92, RJA95, HJF95]. Since 1997, the number of research publications on the techniques of visual information extraction, organization, indexing, user query and interaction, and database management has increased enormously. Similarly, a large number of academic and commercial retrieval systems have been developed by universities, government organizations, companies, and hospitals. Comprehensive surveys of these techniques and systems can be found in [BFS95, YRT99, AMW00].

Content-based image retrieval, uses the visual contents of an image such as *color*, *shape*, *texture*, and *spatial layout* to represent and index the image. In typical content-based image retrieval systems, the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors. The similarities /distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. The indexing scheme provides an efficient way to search for the image database. Recent retrieval systems have incorporated users' relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results. In this chapter, we introduce these fundamental techniques for content-based image retrieval.

2.2 Image Content Descriptors

Generally speaking, image content may include both visual and semantic content. Visual content can be very general or domain specific. *General visual content* include color, texture, shape, spatial relationship, etc. *Domain specific visual content*, like human faces, is application dependent and may involve domain knowledge. *Semantic content* is obtained either by textual annotation or by complex inference procedures based on visual content.

A good visual content descriptor should be invariant to the accidental variance introduced by the imaging process (e.g., the variation of the illuminant of the scene). However, there is a tradeoff between the invariance and the discriminative power of visual features, since a very wide class of invariance loses the ability to discriminate between essential differences. Invariant description has been largely investigated in computer vision (like object recognition), but is relatively new in image retrieval [HBS00].

A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of *regions* or *objects* to describe the image content. To obtain the local visual descriptors, an image is often divided into parts first. The simplest way of dividing an image is to use a *partition*, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using *region segmentation* algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete *object segmentation* to obtain semantically meaningful objects (like ball, car, horse). Currently, automatic object segmentation for broad domains of general images is unlikely to succeed.

In this section, we will introduce some widely used techniques for extracting color, texture, shape and spatial relationship from images.

2.2.1 Color

Color is the most extensively used visual content for image retrieval [JDF90, JHS99, JHI97, MIA89, AKJ89, EMC98, GPR99, MSM95, MJS91, HJO95]. Its three-dimensional values make its discrimination potentiality superior to the single dimensional gray values of images. Before selecting an appropriate color description, color space must be determined first.

2.2.1.1 Color Space

Each pixel of the image can be represented as a point in a 3D color space. Commonly used color space for image retrieval include *RGB*, *Munsell*, *CIE L*a*b**, *CIE L*u*v**, *HSV* (or *HSL*, *HSB*), and *opponent color* space. There is no agreement on which is the best. However, one of the desirable characteristics of an appropriate color space for image retrieval is its *uniformity* [EMC98]. Uniformity means that two color pairs that are equal in similarity distance in a color space are perceived as equal by viewers. In other words, the measured proximity among the colors must be directly related to the psychological similarity among them.

RGB space is a widely used color space for image display. It is composed of three color components *red*, *green*, and *blue*. These components are called "*additive primaries*" since a color in RGB space is produced by adding them together. In contrast, CMY space is a color space primarily used for printing. The three color components are *cyan*, *magenta*, and *yellow*. These three components are called "*subtractive primaries*" since a color in CMY space is produced through light absorption. Both RGB and CMY space are device-dependent and perceptually non-uniform.

The CIE L*a*b* and CIE L*u*v* spaces are device independent and considered to be perceptually uniform. They consist of a luminance or *lightness* component (*L*) and two *chromatic* components *a* and *b* or *u* and *v*. CIE L*a*b* is designed to deal with subtractive colorant mixtures, while CIE L*u*v* is designed to deal with additive colorant mixtures. The transformation of RGB space to CIE L*u*v* or CIE L*a*b* space can be found in [AKJ89].

In HSV (or HSL, or HSB) space is widely used in computer graphics and is a more intuitive way of describing color. The three color components are *hue*, *saturation* (lightness) and *value* (*brightness*). The hue is invariant to the changes in illumination and camera direction and hence more suited to object retrieval. RGB coordinates can be easily translated to the HSV (or HLS, or HSB) coordinates by a simple formula [JDF90].

The opponent color space uses the opponent color axes (*R-G*, *2B-R-G*, *R+G+B*). This representation has the advantage of isolating the brightness information on the third axis. With this solution, the first two chromaticity axes, which are invariant to the changes in illumination intensity and shadows, can be down-sampled since humans are more sensitive to brightness than they are to chromatic information. In the following sections, we will introduce some commonly used color descriptors: the color histogram, color coherence vector, color correlogram, and color moments.

2.2.1.2 Color Moments

Color moments have been successfully used in many retrieval systems (like *QBIC* [MFH95, WNQ93]), especially when the image contains just the object. The *first order* (*mean*), the *second* (*variance*) and the *third order* (*skewness*) color moments have been proved to be efficient and effective in representing color distributions of images [MSM95]. Mathematically, the first three moments are defined as:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \tag{2-1}$$

$$\sigma_i = (\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^2)^{\frac{1}{2}} \tag{2-2}$$

$$s_i = (\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^3)^{\frac{1}{3}} \tag{2-3}$$

where *fij* is the value of the *i*-th color component of the image pixel *j*, and *N* is the number of pixels in the image.

Since only 9 (three moments for each of the three color components) numbers are used to represent the color content of each image, color moments are a very compact representation compared to other color features. Due to this compactness, it may also lower the discrimination power. Usually, color moments can be used as the first pass to narrow down the search space before other sophisticated color features are used for retrieval.

2.2.1.3 Color Histogram

The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing the global distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle.

Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in RGB space, or hue, saturation, and value in HSV space), a *histogram*, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases.

Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance in many applications. One way to reduce the number of bins is to use the opponent color space which enables the brightness of the histogram to be down sampled. Another way is to use clustering methods to determine the *K* best colors in a given space for a given set of images. Each of these best colors will be taken as a histogram bin. Since that clustering process takes the color distribution of images over the entire database into consideration, the likelihood of histogram bins in which no or very few pixels fall will be minimized. Another option is to use the bins that have the largest pixel numbers since a small number of histogram bins capture the majority of pixels of an image [YGH94] Such a reduction does not degrade the performance of histogram matching, but may even enhance it since small histogram bins are likely to be noisy.

When an image database contains a large number of images, histogram comparison will saturate the discrimination. To solve this problem, the *joint histogram* technique is introduced [GPR99]. In addition, color histogram does not take the spatial information of pixels into consideration, thus very different images can have similar color distributions. This problem becomes especially acute for large scale databases. To increase discrimination power, several improvements have been proposed to incorporate spatial information. A simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. As introduced above, the division can be as simple as a rectangular partition, or as complex as a region or even object segmentation. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

2.2.1.4 Color Coherence Vector

In [GPR96] a different way of incorporating spatial information into the color histogram, *color coherence vectors* (*CCV*), was proposed. Each histogram bin is partitioned into two types, i.e., coherent, if it belongs to a large uniformly-colored region, or incoherent, if it does not. Let $\alpha\alpha_i$ denote the number of coherent pixels in the *i*th color bin and β_i denote the number of incoherent pixels in an image. Then, the CCV of the image is defined as the vector <(α_1, β_1), (α_2, β_2), ..., (α_N, β_N)>. Note that <$\alpha_1+\beta_1$, $\alpha_2+\beta_2$, ..., $\alpha_N+\beta_N$> is the color histogram of the image.

Due to its additional spatial information, it has been shown that CCV provides better retrieval results than the color histogram, especially for those images which have either mostly uniform color or mostly texture regions. In addition, for both the color histogram and color coherence vector representation, the HSV color space provides better results than CIE L*u*v* and CIE L*a*b* space.

2.2.1.5 Color Correlogram

The *color correlogram* [JHI97] was proposed to characterize not only the color distributions of pixels, but also the spatial correlation of pairs of colors. The first and the second dimension of the three-dimensional histogram are the colors of any pixel pair and the third dimension is their spatial distance. A color correlogram is a table indexed by color pairs, where the *k*-th entry for (*i*, *j*) specifies the probability of finding a pixel of color *j* at a distance *k* from a pixel of color *i* in the image. Let *I* represent the entire set of image pixels and *Ic(i)* represent the set of pixels whose colors are *c(i)*. Then, the color correlogram is defined as:

$$\gamma_{i,j}^{(k)} = \Pr_{p_1 \in I_{c(i)}, p_2 \in I} [p_2 \in I_{c(j)} \,||\, p_1 - p_2 |= k] \qquad (2\text{-}4)$$

where *i, j* ε {*1, 2, ..., N*}, *k* ε {*1, 2, ..., d*}, and | *p1* − *p2* | is the distance between pixels *p1* and *p2*. If we consider all the possible combinations of color pairs the size of the color correlogram will be very large (*O(N^2d)*), therefore a simplified version of the feature called the *color autocorrelogram* is often used instead. The color autocorrelogram only captures the spatial correlation between identical colors and thus reduces the dimension to *O(Nd)*. Compared to the color histogram and CCV, the color auto-correlogram provides the best retrieval results, but is also the most computational expensive due to its high dimensionality.

2.2.1.6 Invariant Color Features

Color not only reflects the material of surface, but also varies considerably with the change of illumination, the orientation of the surface, and the viewing geometry of the camera. This variability must be taken into account. However, invariance to these environmental factors is not considered in most of the color features introduced above.

Invariant color representation has been introduced to content-based image retrieval recently. In [TGA00], a set of color invariants for object retrieval was derived based on the Schafer model of object reflection. In [GDF96], specular reflection, shape and illumination invariant representation based on blue ratio vector (r/b, g/b,1) is given. In [TGA99], a surface geometry invariant color feature is provided.

These invariant color features, when applied to image retrieval, may yield illumination, scene geometry and viewing geometry independent representation of color contents of images, but may also lead to some loss in discrimination power among images.

2.2.2 Texture

Texture refers to the visual patterns that have the properties of homogeneity that do not result from the presence of only a single color or intensity [JRS96]. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, fabric, etc. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment [RMH73]. Because of its importance and usefulness in Pattern Recognition and Computer Vision, there existed rich research results in the past three decades.

In the early 70's, Haralick et al. proposed the co-occurrence matrix representation of texture feature [RMH73]. This approach explored the gray level spatial dependence of texture. It first constructed a co-occurrence matrix based on the orientation and distance between image pixels and then extracted meaningful statistics from the matrix as the texture representation. Many other researchers followed the same line and further proposed enhanced versions. For example, Gotlieb and Kreyszig studied the statistics originally proposed in [RMH73] and experimentally found that *contrast*, *inverse deference moment* and *entropy* had the biggest discriminatory power [CCG90].

Motivated by the psychological studies in human visual perception of texture, Tamura et al. explored the texture representation from a different angle [HTM78]. They developed computational approximations to the visual texture properties found to be important in psychology studies. The six visual texture properties were *coarseness*, *contrast*, *directionality*, *linelikeness*, *regularity*, and *roughness*. One major distinction between the Tamura texture representation and the co-occurrence matrix representation is that all the texture properties in Tamura representation are visually meaningful whereas some of the texture properties used in co-occurrence matrix representation may not (for example, entropy). This characteristics makes the Tamura texture representation very attractive in Image Retrieval.

In early 90's after Wavelet transform was introduced and its theoretical framework established, many researchers began to study the use of Wavelet transform in texture representation [JRS94, TCC93, ALJ93, MHG94, AKJ92, KST94]. In [JRS94, JRS96], Smith and Chang used the statistics (mean and variance) extracted from the Wavelet sub-bands as the texture representation. To explore the middle-band characteristics, tree structured Wavelet transform was used by Chang and Kuo in [TCC93] to further improve the classification accuracy. Wavelet transform was also combined with other techniques to achieve better performance. Gross et al. used Wavelet transform, together with Kohonen maps, to perform texture analysis in [MHG94]. Thyagarajan et al. [KST94] and Kundu et al. [AKJ92] combined Wavelet transform with co-occurrence matrix to take advantage of both the statistics based and transform based texture analysis.

2.2.3 Shape

Shape can be defined as a "Spatial arrangement of something distinguished from its surroundings by its outline". Shape features of objects or regions have been used in many content-based image retrieval systems [JEG92, WIG90, HVJ91, DTS94]. Compared with color and texture features, shape features are usually described after images have been segmented into regions or objects. Since robust and accurate image segmentation is difficult to achieve, the use of shape features for image retrieval has been limited to special applications where objects or regions are readily available. The state-of-art methods for shape description can be categorized into either *boundary-based* (rectilinear shapes [HVJ91], polygonal approximation [EMA91], finite element models [SSA95], and Fourier-based shape descriptors [KAW90, HKT95, EPK77]) or *region-based* methods (statistical moments [MKH77, LYF94]). A good shape representation feature for an object should be invariant to translation, rotation and scaling. In this section, we briefly describe some of these shape features that have been commonly used in image retrieval applications.

The main idea of *Fourier Descriptor* is to use the Fourier transformed boundary as the shape feature. Some early work can be found in [CTZ72, EPK77]. To take into account the digitization noise in the image domain, Rui et al. proposed a modified Fourier Descriptor which is both robust to noise and invariant to geometric transformations [YRA96].

The main idea of *Moment Invariants* is to use region based moments, which are invariant to transformations, as the shape feature. In [MKH62], Hu identified seven such moments. Based on his work, many improved versions emerged. In [LYF94], based on the discrete versions of Green's theorem, Yang and Albregtsen proposed a fast method of computing moments in binary images. Motivated by the fact that most useful invariants were found by extensive experience and trial-and-error method. Kapur et al. developed algorithms to systematically generate and search for a given geometry's invariants [DKY95]. Gröss and Latecki developed an approach which preserved the qualitative differential geometry of the object boundary, even after an image was digitized [DKY95]. In [DCZ95, ZLD00], a framework of algebraic curves and invariants is proposed to represent complex objects in cluttered scene by parts or patches. Polynomial fitting is done to represent local geometric information, from which geometric invariants are used in object matching and recognition.

Some recent work in shape representation and matching include *Finite Element Method* (FEM) [APW96], Turning Functions [EML91], and Wavelet Descriptor [CGH96]. FEM defines a stiffness matrix, which describes how each point on the object is connected to other points. The eigen-vectors of the stiffness matrix are called modes and span a feature space. All the shapes are first mapped into this space and similarity is then computed based on the eigen-values. Along a similar line of Fourier Descriptor, Arkin et al. developed a Turning Function based method for computing both convex and concave polygons [EML91]. In [CGH96], Chuang and Kuo used Wavelet transform to describe object shape. It embraced the desirable properties such as multi-resolution representation, invariance, uniqueness, stability, and spatial localization. Barrow et al. first proposed the Chamfer matching technique, which compared two collections of shape fragements at a cost proportional to linear dimension, rather than area [HGB77].

In [BLS95], Li and Ma showed that the *Geometric Moments method* (region-based) and the *Fourier Descriptor* (boundary-based) were related by a simple linear transformation. In [BMM97], Babu et al. compared the performance of boundary based representations (Chain code, Fourier Descriptor, UNL Fourier Descriptor), region based representations (Moment Invariants, Zernike moments, Pseudo-Zernike moments), and combined representations (Moment Invariants and Fourier Descriptor, Moment Invariants and UNL Fourier Descriptor). Their experiments showed that the combined representations outperformed the simple representations.

2.2.4 Spatial Information

Regions or objects with similar color and texture properties can be easily distinguished by imposing spatial constraints. For instance, regions of blue sky and ocean may have similar color histograms, but their spatial locations in images are different. Therefore, the spatial location of regions (or objects) or the spatial relationship between multiple regions (or objects) in an image is very useful for searching images.

The most widely used representation of spatial relationship is the *2D strings* proposed by Chang *et al* [SKC87]. It is constructed by projecting images along the *x* and *y* directions. Two sets of symbols, *V* and *A*, are defined on the projection. Each symbol in *V* represents an object in the image. Each symbol in *A* represents a type of spatial relationship between objects. As its variant, the *2D G-string* [SKE88] cuts all the objects along their minimum bounding box and extends the spatial relationships into two sets of spatial operators. One defines local spatial relationships. The other defines the global spatial relationships, indicating that the projection of two objects are disjoint, adjoin or located at the same position. In addition, *2D C-string* [SYL90] is proposed to minimize the number of cutting objects. *2D-B string* [SYL92] represents an object by two symbols, standing for the beginning and ending boundary of the object. All these methods can facilitate three types of query. Type 0 query finds all images containing object *O1*, *O2*, ..., *On*. Type 1 finds all images containing objects that have certain relationship between each other, but the distance between them is insignificant. Type 2 finds all images that have certain distance relationship with each other.

In addition to the 2D string, *spatial quad-tree* [HST84], and *symbolic image* [VNG95] are also used for spatial information representation. However, searching images based on spatial relationships of regions remains a difficult research problem in content-based image retrieval, because reliable segmentation of objects or regions is often not feasible except in very limited applications. Although some systems simply divide the images into regular sub-blocks [MSM96], only limited success has been achieved with such spatial division schemes since most natural images are not spatially constrained to regular sub-blocks. To solve this problem, a method based on the *radon transform*, which exploits the spatial distribution of visual features without a sophisticated segmentation is proposed in [FGJ98, HWF98].

2.2.5 Segmentation

Segmentation is very important to Image Retrieval. Both the shape feature and the layout feature depend on good segmentation. Segmentation is the low-level operation for partitioning images by finding disjoint and homogeneous regions or, equivalently, by finding edges or boundaries. The homogeneous regions, or the edges, are supposed to correspond to actual objects, or parts of them, within the images. Thus, in a large number of applications in image processing and computer vision, segmentation plays a fundamental role as the first step before applying to images higher-level operations such as recognition, semantic interpretation, and representation. A detailed survey of segmentation methods has been reported in Chapter 4.

2.3 Similarity Measures and Indexing

2.3.1 Similarity/Distance Measures

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different *similarity/distance measures* will affect retrieval performances of an image retrieval system significantly. In this section, we will introduce some commonly used similarity measures. We denote *D(I, J)* as the distance measure between the query image *I* and the image *J* in the database; and *fi(I)* as the number of pixels in bin *i* of *I*.

Minkowski-Form Distance

If each dimension of image feature vector is independent of each other and is of equal importance, the *Minkowski-form distance Lp* is appropriate for calculating the distance between two images. This distance is defined as:

$$D(I,J) = (\sum_i | f_i(I) - f_i(J) |^p)^{1/p} \tag{2-5}$$

when p=1, 2, and ∞, $D(I, J)$ is the L_1, L_2 (also called Euclidean distance), and L_∞ distance respectively. Minkowski-form distance is the most widely used metric for image retrieval. For instance, MARS system [YRT97] used Euclidean distance to compute the similarity between texture features; Netra [WYB97, WYM99] used Euclidean distance for color and shape feature, and L_1 distance for texture feature; Blobworld [CCM99] used Euclidean distance for texture and shape feature. In addition, Voorhees and Poggio [HVT88] used L_∞ distance to compute the similarity between texture images. The *Histogram intersection* can be taken as a special case of

L_I distance, which is used by Swain and Ballard [MJS91] to compute the similarity between color images. The intersection of the two histograms of *I* and *J* is defined as:

$$S(I,J) = \frac{\sum_{i=1}^{N} \min(f_i(I), f_i(J))}{\sum_{i=1}^{N} f_i(J)} \qquad (2\text{-}6)$$

It has been shown that histogram intersection is fairly insensitive to changes in image resolution, histogram size, occlusion, depth, and viewing point.

Quadratic Form (QF) Distance

The Minkowski distance treats all bins of the feature histogram entirely independently and does not account for the fact that certain pairs of bins correspond to features which are perceptually more similar than other pairs. To solve this problem, *quadratic form distance* is introduced:

$$D(I,J) = \sqrt{(F_I - F_J)^T A (F_I - F_J)} \qquad (2\text{-}7)$$

where *A*=[*aij*] is a similarity matrix, and *aij* denotes the similarity between bin *i* and *j*. F_I and F_J are vectors that list all the entries in *fi*(*I*) and *fi*(*J*).

Quadratic form distance has been used in many retrieval systems [JHE95, WNQ93] for color histogram-based image retrieval. It has been shown that quadratic form distance can lead to perceptually more desirable results than Euclidean distance and histogram intersection method as it considers the cross similarity between colors.

Mahalanobis Distance

The *Mahalanobis distance* metric is appropriate when each dimension of image feature vector is dependent of each other and is of different importance. It is defined as:

$$D(I,J) = \sqrt{(F_I - F_J)^T C^{-1} (F_I - F_J)} \qquad (2\text{-}8)$$

where *C* is the covariance matrix of the feature vectors.

The Mahalanobis distance can be simplified if feature dimensions are independent. In this case, only a variance of each feature component, c_i, is needed.

$$D(I,J) = \sum_{i=1}^{N} (F_I - F_J)^2 / c_i \qquad (2\text{-}9)$$

Cosine measure

The Cosine measure is a very important similarity measure for points in the plane (and in higher dimensions). The cosine measure assigns a high similarity to points that are in the same direction from the origin, zero similarity to points that are perpendicular to one another, and negative similarity for those that are pointing in opposing directions to one another.

$$D(I,J)=\frac{\sum(IJ)}{(\sum I^2)(\sum J^2)} \tag{2-10}$$

2.3.2 Indexing Scheme

Another important issue in content-based image retrieval is effective indexing and fast searching of images based on visual features. Because the feature vectors of images tend to have high dimensionality and therefore are not well suited to traditional indexing structures, *dimension reduction* is usually used before setting up an efficient indexing scheme.

One of the techniques commonly used for dimension reduction is *principal component analysis* (*PCA*). It is an optimal technique that linearly maps input data to a coordinate space such that the axes are aligned to reflect the maximum variations in the data. The QBIC system uses PCA to reduce a 20-dimensional shape feature vector to two or three dimensions [MFH95] [WNQ93]. In addition to PCA, many researchers have used *Karhunen-Loeve* (*KL*) *transform* to reduce the dimensions of the feature space. Although the KL transform has some useful properties such as the ability to locate the most important sub-space, the feature properties that are important for identifying the pattern similarity may be destroyed during blind dimensionality reduction [WJK95]. Apart from PCA and KL transformation, *neural network* has also been demonstrated to be a useful tool for dimension reduction of features [JAC00].

After dimension reduction, the multi-dimensional data are indexed. A number of approaches have been proposed for this purpose, including *R-tree* (particularly, *R*-tree* [NBT90]), *linear quad-trees* [JVM99], *K-d-B tree* [JTR81] and *grid files* [JNH84]. Most of these multi-dimensional indexing methods have reasonable performance for a small number of dimensions (up to 20), but explore exponentially with the increasing of the dimensionality and eventually reduce to sequential searching. Furthermore, these indexing schemes assume that the underlying feature comparison is based on the Euclidean distance, which is not necessarily true for many image retrieval applications. One attempt to solve the indexing problems is to use hierarchical indexing scheme based on the *Self-Organization Map* (*SOM*) proposed in [HJF95].

2.4 User Interaction

For content-based image retrieval, user interaction with the retrieval system is crucial since flexible formation and modification of queries can only be obtained by involving the user in the retrieval procedure. User interfaces in image retrieval systems typically consist of a query formulation part and a result presentation part.

2.4.1 Query Specification

Specifying what kind of images a user wishes to retrieve from the database can be done in many ways. Commonly used query formations are: *category browsing*, *query by concept*, *query by sketch*, and *query by example*. Category browsing is to browse through the database according to the category of the image. For this purpose, images in the database are classified into different categories according to their semantic or visual content [AVM01]. Query by concept is to retrieve images according to the conceptual description associated with each image in the database. Query by sketch [GDF96] and query by example [JAA00] is to draw a sketch or provide an example image from which images with similar visual features will be extracted from the database.

Query by sketch allows user to draw a sketch of an image with a graphic editing tool provided either by the retrieval system or by some other software. Queries may be formed by drawing several objects with certain properties like color, texture, shape, sizes and locations. In most cases, a coarse sketch is sufficient, as the query can be refined based on retrieval results.

Query by example allows the user to formulate a query by providing an example image. The system converts the example image into an internal representation of features. Images stored in the database with similar features are then searched. Query by example can be further classified into query by external image example, if the query image is not in the database, and query by internal image example, if otherwise. For query by internal image, all relationships between images can be pre-computed. The main advantage of query by example is that the user is not required to provide an explicit description of the target, which is instead computed by the system. It is suitable for applications where the target is an image of the same object or set of objects under different viewing conditions. Most of the current systems provide this form of querying.

Query by group example allows user to select multiple images. The system will then find the images that best match the common characteristics of the group of examples. In this way, a target can be defined more precisely by specifying the relevant feature variations and removing irrelevant variations in the query. In addition, group properties can be refined by adding negative examples. Many recently developed systems provide both query by positive and negative examples.

2.4.2 Relevance Feedback

Human perception of image similarity is subjective, semantic, and task-dependent. Although content-based methods provide promising directions for image retrieval, generally, the retrieval results based on the similarities of pure visual features are not necessarily perceptually and semantically meaningful. In addition, each type of visual feature tends to capture only one aspect of image property and it is usually hard for a user to specify clearly how different aspects are combined. To address these problems, interactive *relevance feedback*, a technique in traditional text-based information retrieval systems, was introduced. With relevance feedback [YRT98] [TPM96] [YRA97] [JHS97], it is possible to establish the link between high-level concepts and low-level features.

Relevance feedback is a supervised active learning technique used to improve the effectiveness of information systems. The main idea is to use positive and negative examples from the user to improve system performance. For a given query, the system first retrieves a list of ranked images according to a predefined similarity metrics. Then, the user marks the retrieved images as relevant (positive examples) to the query or not relevant (negative examples). The system will refine the retrieval results based on the feedback and present a new list of images to the user. Hence, the key issue in relevance feedback is how to incorporate positive and negative examples to refine the query and/or to adjust the similarity measure.

2.5 Performance Evaluation

To evaluate the performance of retrieval system, two measurements, namely, *recall* and *precision* [AMW00], are borrowed from traditional information retrieval. For a query, let the retrieved number of images be R. R will contain relevant set of images r and irrelevant set of images i. i.e $R = r + i$. Database contains total N number of images and let n number of relevant images as for the query.

Precision is defined as a measure of the usefulness of the retrieved list of images; i.e. it is the percentage of retrieved images that are relevant.

$$\Pr ecision = \frac{\# of\ relevant\ images\ retrieved}{\# of\ images\ retrived} = \frac{r}{R} \tag{2-11}$$

Recall is defined as a measure of the completeness of the retrieved list of images, i.e. it is the percentage of all relevant images that are found by the search operation.

$$\operatorname{Re} call = \frac{\# of\ relevant\ images\ retrieved}{\# of\ relevant\ images\ in\ Collection} = \frac{r}{n} \tag{2-12}$$

To quantify the performance of retrieval systems normally *average precision recall (APR)* curves are used. APR curves are plotted as

$$\frac{1}{2}\sum_{i=1}^{N-1}(x_{i+1} - x_i)(y_{i+1} + y_i) \tag{2-13}$$

where (x_i,y_i) is the (recall, precision) pair when the number of retrieved images is i and N is the total number of top marches defines the *performance area.*

2.6 Some Content Based Image Retrieval Systems

Below we describe some Content Based Image Retrieval Systems which are in use

2.6.1 Amore (Advanced Multimedia Oriented Retrieval Engine)

Developer C & C Research Laboratories NEC USA, Inc.

URL http://www.ccrl.com/amore/.

References [SMK97], [SMK99].

Features The image is segmented into at most eight regions of homogeneous color, and downsized to 24×24 pixels. The regions in this picture are directly used for matching [KHY93].

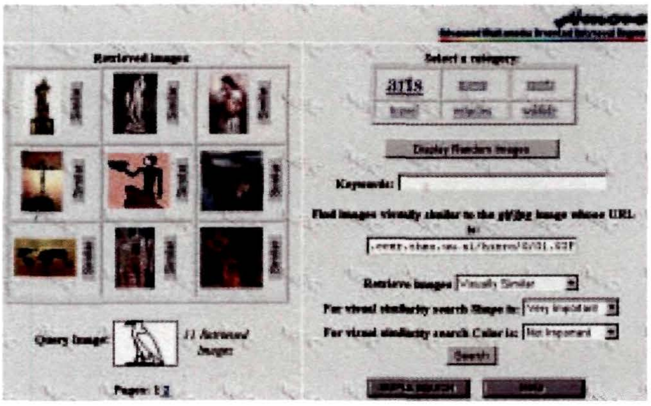
Querying The user first selects a category of images. An initial set of images can be selected at random or by keyword. Of these images, visually similar images can be retrieved. The query image can also be specified by its URL. In a research version of the system, sketching a query image was possible. The user can indicate the relative importance of color and shape.

Matching First a correspondence between regions in the query and target image is found Regions corresponding to the same regions in the other image are merged. The shape similarity between two regions is based on the number of pixels of overlap, a kind of template matching. The color similarity between two regions is the distance in HLS space between the uniform region colors [KHY93].

Indexing Indexing is performed by the COIR (Content-Oriented Image Retrieval) system, refer [KHS97].

Result presentation The retrieved images are shown as thumbnails, without an explicit order, see figure 2-1. In a research version of the system, result images were displayed as a scatter plot, with shape and color similarity values at the axes, or on a perspective wall [SMK97].

Applications Amore is used in the art retrieval system Arthur (Art Media and Text Hub and Retrieval System, http://www.isi.edu/cct/arthur/, developed at the Center for Cultural Technology within the Information Sciences Institute of the University of Southern California.



Retrieved images

Select a category:

arts

Display Random images

Keywords:

Find images visually similar to the gif/jpg image whose URL is:

Retrieve images: Visually Similar

For visual similarity search Shape is: Very Important

For visual similarity search Color is: Not Important

Search

Query Image:

11 Retrieved Images

Pages: 1 2

Figure 2.1: Amore result of similarity retrieval on shape.

2.6.2 BDLP (Berkeley Digital Library Project)

Developer University of California, Berkeley.

URL The homepage of project is at http://elib.cs.berkeley.edu/, a demo of retrieval from all photos in the collection is at http://elib.cs.berkeley.edu/photos/all.shtml.

References [CCV96].

Features There is a number of alphanumerical keys available for querying: the collection, key words, location, county, and photographer. The colors of each image are quantized into 13 colors bins. Six values are associated with each color bin: the percentage of the image with colors in that bin, and the number of 'very small', 'small', 'medium', 'large', and 'very large' dots of that color found.

Querying For content-based search, the user can select 13 colors, and indicate the amount ('any', 'partly', 'mostly') of that color in the picture. Also, for colored regions the user can indicate the size ('any', 'small', 'medium', 'large') and the quantity of regions with that color ('any', 'few', 'some', 'many').

Matching Image features are stored as text strings. For example, a picture of a sky with clouds might have a few large white regions, and a large amount of blue, and would have a feature text string "mostly blue large white few". Matching is done by substring matching, e.g. with query string "large white%".

Indexing All features are put into a relational database (the Informix Universal Server database management system).

Result presentation The retrieved photos are presented unordered, with id-number, photographer, and collection keys.

Applications The collections consist of 23195 images of plants, animals, people, and landscapes, 17000 images from the California Department of Water Resources, Corel Stock photos, and aerial photos of the Sacramento river delta region.

2.6.3 Blobworld

Developer Computer Science Division, University of California, Berkeley.

URL http://elib.cs.berkeley.edu/photos/blobworld/. A demo of Blobworld is available at http://elib.cs.berkeley.edu/photos/blobworld/start.html.

References [CCM00].

Features The features used for querying are the color, texture, location, and shape of regions (blobs) and of the background. The color is described by a histogram of 218 bins of the color coordinates in Lab-space. Texture is represented by mean contrast and anisotropy over the region, as the 2D coordinate (*contrast; contrast* × *anisotropy*). Shape is represented by (approximate) area, eccentricity, and orientation.

Querying The user first selects a category, which already limits the search space. In an initial image, the user selects a region (blob), and indicates the importance of the blob ('somewhat', 'very'). Next, the user indicates the importance of the blob's color, texture, location, and shape ('not', 'somewhat', 'very'). More than one regions can be used for querying.

Matching To match two color histograms h1 and h2, the quadratic form distance is used: $d(h_1, h_2) = (h_1 - h_2)^T A(h_1 - h_2)$, where $A = (a_{ij})$ is a symmetric matrix of weights representing the similarity between color bins i and j. The distance between two texture descriptors is the Euclidean distance between their coordinates in representation space. The distance between centroids is the Euclidean distance. The distances are combined into a single final distance.

Indexing Rather than actually computing the distances between the full color histogram vectors of length 218 as $d(h_1, h_2) = (h_1 - h_2)^T A(h_1 - h_2)$, singular value decomposition (SVD) is used to project the histogram vectors onto a lower-dimensional subspace. The resulting points are indexed by an R*-tree [NBH90].

Result presentation The retrieved images are ranked in linear order, and presented together with the segmented version showing the regions, see Figure 2.2.

Applications The demo on the web provides retrieval from a collection of 10000 Corel stock photos.

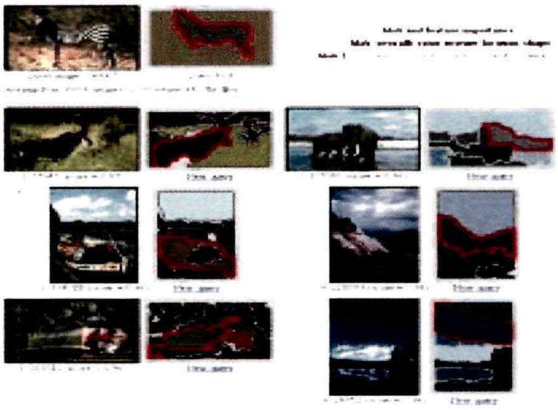


Figure 2.2: Blobworld query result.

2.6.4 **C-bird** (Content-Based Image Retrieval from Digital libraries)

Developer School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada.

URL http://jupiter.cs.sfu.ca/cbird/.

References [ZNL98], [ZNL99].

Features For each collected image, a feature descriptor and a layout descriptor are computed. A feature descriptor is a set of four vectors: a color vector, a most frequent color (MFC) vector, a most frequent orientation (MFO) vector, and a chromaticity vector. A 512-bin RGB histogram is stored in the color vector. The centroids of the regions associated with the 5 most frequent colors form the MFC vector and the centroids of regions of the 5 most frequent edge orientations form theMFO vector. The 36-dimensional chromaticity vector is computed as follows: first, a normalization of each RGB channel is made to obtain illumination invariance, then the 3D color histogram is replaced by a 2D chromaticity histogram. Treating this chromaticity histogram as an image, first a wavelet-based image reduction is applied, then the Discrete Cosine Transform coefficient matrix is built. The chromaticity vector is made of the 36 values of the upper left corner of the DCT matrix. For search by object model, some geometric data such as the area, the centroid and the eccentricity are computed from color regions associated with each of the MFCs. The layout descriptor contains a color layout vector and an edge layout vector. To construct these vectors the image is divided into 64 cells, and for each cell the most frequent colors and the number of edges for each orientation are determined. Also, for images at half and quarter resolution, a feature descriptor like the one described above is stored.

Querying The user is presented a grid of consecutive images from the database starting at a random position. To start a query by color histogram or color similarity with illumination invariance, one of the buttons under the selected query image is pressed (see Figure 2.3). For a query by color or texture layout, grids are presented for drawing color, texture density and edge orientation layout (see Figure 2.3). For a query by color percentage, 5 colors and their percentages are indicated by the user. For a query by object model, the user browses through a selection of query images and makes a choice.

Matching The distance between two chromaticity vectors in an illumination invariant color query is the L2 distance. Texture orientation histograms, as well as color histograms for the full image, are matched by histogram intersection.

The first step in a query by object model is a color localization: color regions for each MFC are extracted and for each region, some geometric data such as the area, the centroid and the eccentricity are computed. After selecting the images in the database that share a number of color regions with the query image, a number of vectors are produced by connecting the centroid of the first MFC region with the centroids of the other MFCs. Analyzing the length of these vectors and the angles between them, a hypothesis regarding the existence of an object at a certain scale and orientation (the difference of angles between centroids of the regions corresponding to the MFCs in the query and database image) is made. This hypothesis is tested in a second step by comparing the texture histogram for each pair of matching regions in the two images. The 2D texture histogram measures orientation (the gradient direction of the edge pixels) and edge separation from the grey level image. Finally, if there is sufficient similarity in their texture between the query object and the area in the database image where the supposed similar object was identified, a shape verification based on the Generalized Hough Transform is performed [DHB81].

Result presentation The user can choose the number of rows and columns of the displayed images grid. By clicking on a thumbnail image the user can see some color and texture characteristics of the image (color percentage and layout, texture layout).

Figure 2.3: C-bird query interface and query result.

2.6.5 Chabot

Developer Department of Computer Science, University of California, Berkeley, CA, USA.

URL http://http.cs.berkeley.edu/˜ginger/chabot.html. Chabot has evolved into Cypress (which, surprisingly, seems not to have inherited content based query capability). For a demo of Cypress, see http://elib.cs.berkeley.edu/photos/.

References [VEO95].

Features One of the early systems, Chabot aimed at combining text based descriptions with image analysis in retrieving images from a collection of photographs of the California Department of Water Resources. The system made use of an existing text description database of the collection, adding other types of textual information for querying such as the shooting date, the picture location, the perspective of the photo. For each image a color histogram containing only 20 bins is computed.

Querying The user is presented with a list of search criteria (such as keywords, photographer, film format, shooting date, perspective, location, and colors). The color criterion offers limited options for the user to choose from, such as 'mostly red' or 'some yellow'. The user has the possibility to define concepts, which are combinations of search criteria that the concept satisfies. For example, the concept of 'sunset' is defined as a combination of keyword ('sunset') and color ('mostly red' or 'mostly orange') criteria.

Matching To match a 'mostly …' color criterion, more than 50% of the pixels in an image must have the requested color. For the 'some …' color criterion, one or two of the 20 colors in the histogram must be qualified as the requested color.

Indexing The images and associated data are stored in the database POSTGRES, developed at the University of California, Berkely.

Result presentation Images are shown without specific order.

Applications The database contains 11643 images of California natural resources.

2.6.6 Circus (Content-based Image Retrieval and Consultation User-centered System)

URL http://lcavwww.epfl.ch/˜zpecenov/CIRCUS/Demo.html.

References [ZPF98], [ZPI97]

Features The color feature is a global histogram in Lab space. The texture features are the mean, standard

deviation, angular second moment, inverse difference moment, sum average, contrast, correlation, and sum variance, all derived from the co-occurrence matrix of gray values in the direction of $\pi/4$. These feature vectors are projected onto a lower dimensional space, derived from the 'feature by image' occurrence matrix $A = (a_{ij})$, with $a_{ij} = l_{ij}g_i$, where l_{ij} is the local weighting of feature i i in image jj, and g_i is the global weighting of feature i. This is analogous to the 'term by document' matrix used in Latent Semantic Indexing for text retrieval. Using singular value decomposition (SVD), an orthogonal base for features and images is computed, in which AA is expressed as a linear combination. Matrix A is then approximated by the first few terms of this sum. To speed up the computations, a fast approximation to the decomposition is computed using wavelet packets.

Querying From a query image, features are extracted and projected onto the lower-dimensional feature space. Alternatively, color queries can be specified by giving the percentages of each color in the desired image.

Matching The distance between two projected feature vectors is the cosine of the angle between the two vectors.

Relevance feedback Users can specify for each query a set of positive and negative examples. The new query consist of the intersection of features from the positive examples,minus the features from the negative examples.

Result presentation The images are shown in linear decreasing order of scoring value.

2.6.7 **Compass** (Computer Aided Search System)

Developer Centre for Scientific and Technological Research, Trento, Italy.

URL http://compass.itc.it/.

References [RBO00]

Features The color features used are the hue and luminance histograms, and the 2D hue and luminance coocurrence matrices. The texture feature is the histogram of the magnitude of the luminance gradient.

Querying The system works with query by example. However, rather than the usual single query image, the query consists of a set of images, see figure 2-4, which is then sent to possibly more than one image database server. An alternative is to browse through the database. For this purpose the database is clustered. Key images of all the clusters are projected using multidimensional scaling onto a line.

Matching The distance between two feature histograms is the L_1 distance. The distance between two images is a weighted sum of the individual feature distances. The cloud of query feature vectors in feature space is clustered into a number of query sets Q_iQi. The distance $d(I,Q_i)$ between a database image I and a query set of images Q_i, is the minimum over all images from the query set.

Relevance feedback The user can indicate which result images are relevant, and which are not. The relevant images are added to the query set. When a set of irrelavant images Q' 0 has been specified, the distance $d(I, Q, Q')$) between a database image I and a query set of images Q, given a set of irrelevant images Q' is given by $d(I, Q)/ (d(I,Q)/d(I,Q'))^{\gamma}$.

Result presentation The answers from the multiple servers are then merged and proposed to the user as a single result, shown in decreasing order of similarity score.

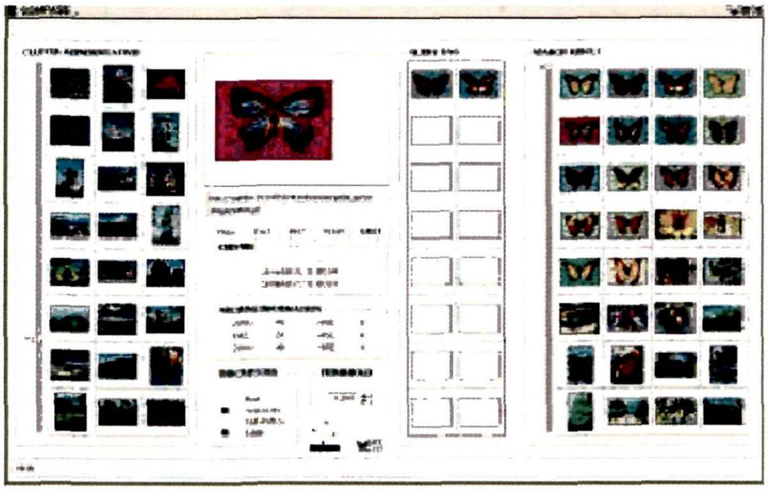


Figure 2.4: Compass interface.

2.6.8 Excalibur Visual RetrievalWare

The Visual RetrievalWare is a software developers kit for building applications for manipulating digital image files and their visual content. The toolkit contains C++ and Java API's for image processing, feature extraction, indexing and content-based retrieval. It also includes sample programs which might be used directly or can serve as templates for building more complex applications. One of these sample programs is the CST (Color, Shape, and Texture) demo.

Developer Excalibur Technologies.

URL http://vrw.excalib.com/. A demo is at http://vrw.excalib.com:8015/cst.

References [EVR00].

Features The CST demo allows queries by example based on HSV color histograms, relative orientation, curvature and contrast of lines in the image, and texture attributes, that measure the flow and roughness in the image.

Querying The user first defines the desired visual similarity by specifying the relative importance of the above image attributes, and then selects one of the displayed images as query, see figure 2-5.

Result presentation The images are shown without an explicit ordering.

Applications The software has been used in the Image Surfer system, which was used by the Yahoo! and Infoseek WWW search engines.

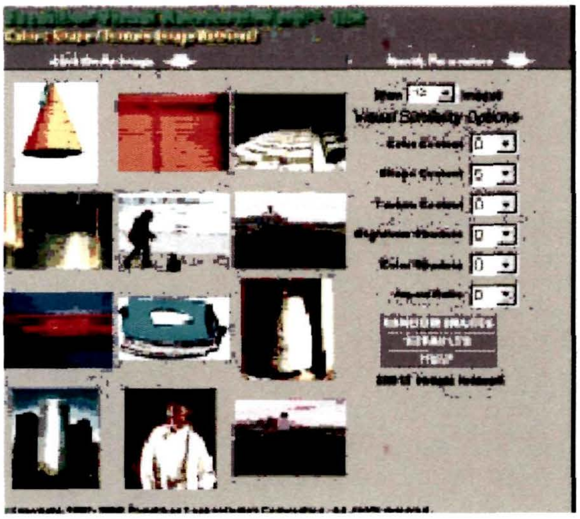


Figure 2.5: Excalibur: result of querying with upper left image based on shape alone.

2.6.9 FOCUS (Fast Object Color-based Query System)

Developer Department of Computer Science, University of Massachusetts, Amherst, MA.

URL http://wagga.cs.umass.edu/~mdas/color_proj.html. A demo of the system is available at http://cowarie.cs.umass.edu/~colordemo/mdas/demo1/phase0.html.

References [MDE97].

Querying The user can select as query one of the displayed template images, or create a new template by marking a sub-image which contains the region of interest.

Features Each image is divided in cells of 100×100 pixels and for each cell a color histogram in the HSV space, coarsely quantized along the saturation and value axes (64×10×10), is computed. The peaks of all local histograms are determined and combined in a list of unique peaks for the whole image by merging multiple copies of the same peak. Also, a frequency table is constructed which, for each color in the HSV space, gives the number of images that have a peak of that color.

The spatial relationships between colored regions are represented by means of a spatial proximity graph (SPG) constructed in two phases. First an intermediate SPG is generated, with one node corresponding to each color peak computed for the image cells. Two nodes in this graph are connected if their corresponding peaks are located in the same cell or are located in neighboring cells and have the same color. This graph is then simplified, by unifying all connected nodes of the same color in a single node, and stored using an adjacency matrix representation.

For the query image, a global color histogram is computed and color region relationships are determined at pixel level.

Matching The peaks of a query image are subjected to approximate range queries in the increasing order of their corresponding entries in the frequency table. From the resulting lists, the set of images which have peaks matching all query peaks are determined. For the images in this set, a matching score is computed as the sum of the L_1 distances between each query peak and the matched candidate peak.

To match the SPG of the query image with that of a candidate image, first the candidate SPG is reduced by removing any node whose corresponding peak does not match a query peak. Then it is checked if the query graph appears as a sub-graph in the candidate SPG.

Indexing The color peaks of the database images are stored in a B^+ tree [HFK91] sorted with hue as the primary key, followed by saturation and value.

Result presentation When the user submits a query by clicking on an image, the images are retrieved using the first phase of matching (the images displayed are the images that have all the colors of the query image). By clicking on the 'Refine Results' button, the retrieved images are subjected to the second phase of matching, where the spatial relationships of the matched color regions is analyzed in order to detect a query object in a candidate image.

Applications The database consists of 400 advertisements and 800 color natural images.

2.6.10 ImageScape

Developer Department of Computer Science, Leiden University, The Netherlands.

URL http://www.wi.leidenuniv.nl/home/lim/image.scape.html. A demo is available at http://ind134a.wi.leidenuniv.nl:2001/new2/imagesearch.demo.html.

References [MLK97], [JMB00].

Querying Using the sketch interface, the user can draw an outline of the desired image. For semantic querying, the user brings icons on a canvas that represent the objects/concepts he is looking for, at the desired position in the image. Examples of object/concept categories include human faces, stone or sand, water, sky, tree or grass, points and lines (see Figure 2.6).

Features Edge maps of the images collected by Web crawlers are obtained using the Sobel operator and a Gaussian blurring filter. A frequency histogram of the 3×3 binary pixel patterns occurring in the edge image, which is called trigram vector, is computed for all images. This vector is subjected to a dimensionality reduction using a band-pass filter. Various other features, used in object matching, are taken at pixel level: color, Laplacian, gradient magnitude, local binary patterns, invariant moments and Fourier descriptors.

Matching The first step of the object matching process uses the L_1 distance on the trigram vectors to retrieve the top 1% matches from the entire database. Among these, 20 matches are selected in a second step, a 20×20 template matching, using the most informative pixels to minimize the misdetection rate. These pixels are found as follows. For each object, a large set of positive and negative examples are used in finding the set of 256 pixels with the greatest discriminatory power, by maximizing the Kullback relative information combined with a Markov random field.

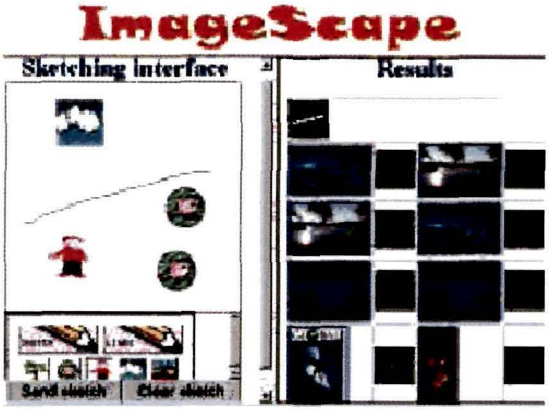


Figure 2.6: ImageScape query built with icons.

2.6.11 MARS (Multimedia Analysis and Retrieval System)

Developer Department of Computer Science, University of Illinois at Urbana-Champaign, further developed at Department of Information and Computer Science, University of California at Irvine, CA.

URL http://www-db.ics.uci.edu/pages/research/mars.shtml. Demos are available at location http://www-db.ics.uci.edu/pages/demos/index.shtml.

References [MOY97].

Features The system supports queries on combinations of low-level features (color, texture, shape) and textual ,descriptions. Color is represented using a 2D histogram over the HS coordinates of the HSV space. Texture is represented by two histograms, one measuring the coarseness and the other one the directionality of the image, and one scalar defining the contrast. In order to extract the color/texture layout, the image is divided into 5×5 sub-images. For each sub-image a color histogram is computed. For the texture of a sub-image, a vector based on wavelet coefficients is used. The object in an image is segmented out in two phases. First, a *k*-means clustering method in the color-texture space is applied, than the regions detected are grouped by an attraction based method. This consists of choosing a number of attractor regions and associating each region with the attractor that has the largest attraction to it. The attraction between two regions, *i* and *jj*, is defined as $F_{ij} = M_iM_j/d^2_{ij}$, where $M_{i,}$ M_j are the sizes of the two regions and d_{ij} is the Euclidean distance between the two regions in the spatial-color-texture space. In the MARS system, five attractors are used: one for each corner of the image (background attractors) and one in the center of the image (the objects attractor). This is consistent with the fact that their database consists of images of single objects. The shape of the boundary of the extracted object is represented by means of Fourier Descriptors (FD).

Querying Complex queries can be formulated using boolean operators. The desired features can be specified either by example (pointing an image database that has such a property) or direct (for example, by choosing colors from a palette or textures from an available set of patterns).

Matching The similarity distance between two color histograms is computed by histogram intersection. The similarity between two textures of the whole image is determined by a weighted sum of the Euclidean distance between contrasts and the histogram intersection distances of the other two components, after a normalization of the three similarities. For computing the texture similarity between two corresponding sub-images, the Euclidean distance between the vector representations is used. A weighted sum of the 5×5 color/texture similarities is used to compute the color/texture layout distance between two images. The similarity measure between two FD shape representations is a weighted sum of the standard deviations of $ratio(k) = M_2(k)/M_1(k)$ and $shift(k) = \theta_2(k) - \theta_1(k) - \Psi$, $k = -N_c, \ldots\ldots, N_c$, where $M_i(k)$ and $\theta_i(k)$ are the magnitude and the phase angle of the FD coefficients,   is the difference of the major axis orientations of the two shapes and Nc is the number of FD coefficients.

Each query has a query tree associated. In a query tree, the leaves represent the feature vectors (the terms of the boolean expression defining the query) while the internal nodes correspond to boolean operators or more complex terms indicating a query by object. Individual queries on each of the query terms are made. The tree is evaluated bottom-up: each internal node receives from each child a list of ranked images and combines these lists, after a normalization process, according to the weights on the parent-child links.

Indexing There is no information about indexing data structures used for queries. A new version of the system, WebMARS, is developed where the feature vectors are indexed using hybrid trees, which combine aspects of several indexing trees.

Result presentation Images are listed in order of decreasing similarity.

Relevance feedback Initially, the weights of the edges in the query tree are equal for the children of the same parent and their sum is 1. Based on the relevant images chosen by the user from the query result list, a tree re-weighting process takes place.

Applications The database consists of images of ancient African artifacts from the

Fowler Museum of Cultural History at UCLA.

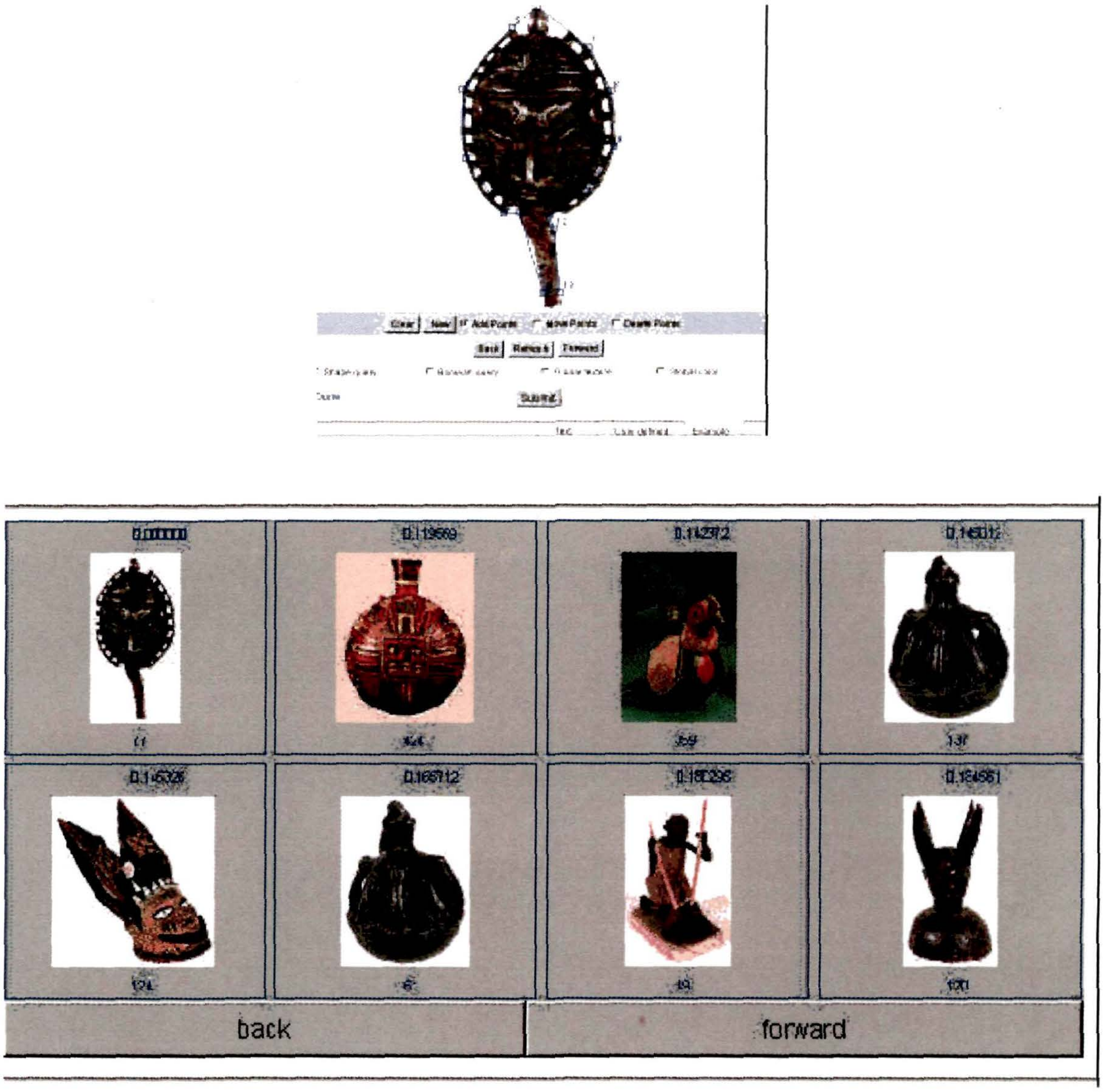


Figure 2.7: Mars shape query with drawn polygon.

2.6.12 NETRA

Developer Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA.

URL http://maya.ece.ucsb.edu/Netra/. A demo of the system is available at web address http://maya.ece.ucsb.edu/Netra/netra.html.

References [WYM97], [WYM99].

Features Images in the database are segmented into regions of homogeneous color. Of those regions, the following features are extracted: color, texture, shape, and spatial location. On the basis of a training set of images, the RGB color space is quantized, and represented by a color codebook of 256 colors, the centroids of the quantization cells. The colors of an image region are also quantized, giving a color feature vector fc = $(c_0,p_0, \ldots.., c_n,p_n)$, with c_i the index into the color code book, and pi the fraction of that color in the region, $p_0+ \ldots +p_n = 1$. The number n is the number of colors used to represent the region, which is different for each region. Texture is represented by a feature vector f_i t containing the normalized mean and standard deviation of a series of Gabor wavelet transforms of the image: $f_t = (\mu_{0,0}, \ldots, \mu_{s,k}, \sigma_{0,0}, \ldots, \sigma_{s,k})$,with s the number of scales, and k the number of directions.

There are three feature vectors used to represent the shape of regions. The first, f_K, is based on the curvature function of the contour, giving the curvature at each point on the contour. The second, f_R is based on the centroid distance function, giving at each contour point the distance to the centroid of the region. The third, $f_Z$is the complex coordinate function, representing each contour point as a complex number with real component equal to the x-coordinate, and the imaginary component equal to the y-coordinate. On 64 samples of each of these functions, the fast Fourier transform (FFT) is applied, of which the real (amplitude) component of the coefficients is used, the numbers $F_{-31}, \ldots, F_{32}$. The feature vectors are as follows:

$f_K = (|F_1|, \ldots, |F_{32}|)$, $f_R = (|F_1|, \ldots, |F_{32}|) / |F_0|$, $f_Z = (|F_{-31}|, \ldots, |F_{-1}|, |F_2, \ldots, |F_{32}||) / |F_1|$.

Querying There are 2,500 images from the Corel photo collection, organized in 25 categories, with 100 images in each category. You can select any one of them as the query image. All images in the database have been segmented into homogeneous regions. You can click on one of the regions and select one of the four image attribute color, spatial location, texture, and shape. Instead of using an image example, you can also directly specify the color and spatial location. The spatial location querying tool utilizes two bounding boxes to define the area of interest. The inner box is used to define the preferred area, and the box outside is used to constrain the objects to be within this area. Thus, if the object has any its bodies exceeding this outside box, they will not be considered.

Matching Consider two color feature vectors, f_c^A of region A, and f_c^B of region B. For each color c_i in f_c^A, the closest color $c^B{}_k$ in $f^B{}_c$ c is found, and the distance $d(c_i^A, f_c^B)$ is calculated as the weighted Euclidean distance in RGB space: $d(c_i^A, f_c^B) = |p_i^A - p_k^B|\ d(c_i^A, c_k^B)$. The distance between two texture feature vectors is the L_1 distance. The distance between two shape feature vectors is the Euclidean distance.

Indexing is based on the SS-tree [DWR96]. Color, texture, and shape are indexed separately. The first feature the user specifies is used to retrieve about 100 candidates. Then this feature and the possible other features together are used to order the retrieval result.

Result presentation The matched images are linearly ordered, see figure 2-8.

Applications An initial prototype of NETRA is used in ADL (see above) to search on texture.

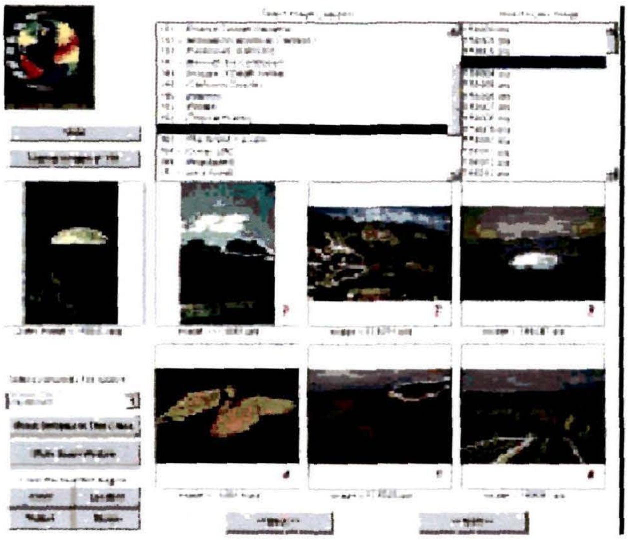


Figure 2.8: NETRA. Result of querying on shape with the complex description.

2.6.13 QBIC (Query By Image Content)

Developer IBM Almaden Research Center, San Jose, CA.

URL http://wwwqbic.almaden.ibm.com/. A demo version of the system is available at site http://wwwqbic.almaden.ibm.com/cgi-bin/stamps-demo.

References [WNR93].

Features Color features computed are: the 3D average color vector of an object or the whole image in RGB, YIQ, Lab, and Munsell color space and a 256-dimensional RGB color histogram. If x is an n-dimensional color histogram and $C = [c_1 c_2 \ldots c_n]$ is a $3 \times n$ matrix whose columns represent the RGB values of the n quantized colors, the average color vector x_{avg} is $C\ x$. The texture features used in QBIC are modified versions of the coarseness, contrast, and directionality features proposed by Tamura [HTM78].

The shape features consist of shape area, circularity, eccentricity, major axis orientation and a set of Algebraic moment invariants. The major axis orientation and the eccentricity are computed from the second order covariance matrix of the boundary pixels: the major axis orientation as the direction of the largest eigenvector and eccentricity as the ratio of the smallest eigen value to the largest one. For the database images, these shape features are extracted for all the object contours, semi-automatically computed in the database population step. In this process, the user enters an approximate object outline, which is automatically aligned with the nearby image edges, using the active contours technique. In this object detection step, the user can also associate text to the outlined objects.

The 18 algebraic moment invariants are the eigen values of the matrices $M_{[2,2]}$, $M_{[2,3]} \times M_{[3,2]}$, $M_{[3,3]}$, $M_{[3,4]} \times M_{[4,3]}$, $M_{[4,4]}$, $M_{[4,5]} \times M_{[5,4]}$, where the elements of $M_{[i,j]}$are scaled factors of the central moments.

QBIC also implemented a method of retrieving images based on a rough user sketch. For this purpose, images in the database are represented by a reduced binary map of edge points. This is obtained as follows: first, the color image is converted to a single band luminance; using a Canny edge detector, the binary edge image is computed and is next reduced to size 64 × 64. Finally this reduced image is thinned.

Querying QBIC allows queries based on example images, user-constructed sketches or/and selected color and texture patterns. In the last case, the user chooses colors or textures from a sampler. The percentage of a desired color in an image is adjusted by moving sliders.

Matching For the average color, the distance between a query object and database object is a weighted Euclidean distance, where the weights are the inverse standard deviation for each component over the samples in the database. In matching two color histograms, two distance measures are used: one low dimensional, easy to compute (the average color distance) and one much more computationally expensive (the quadratic histogram distance). The first one (which is computed for all the images in the database) acts as a filter, limiting the expensive matching computation to the small set of images retrieved by the first matching. The average color distance is $d^2_{\text{avg}}(x,y) = (x_{\text{avg}} - y_{\text{avg}})^t(x_{\text{avg}} - y_{\text{avg}})$. The histogram quadratic distance is given by $d^2_{\text{hist}}(x,y) = (x - y)^t A(x - y)$, where the symmetric color similarity matrix A is given by $a_{ij} = 1 - d_{ij}/d_{\max}$, with d_{ij} being the L_2 distance between the colors i and j j in the RGB space and $d_{\max} = max_{i,j}d_{ij}$. The texture distance is a weighted Euclidean distance, with the weighting factors being the inverse variances for each of the three texture components over the entire database. Two shapes are matched also by a similar weighted Euclidean distance between shape feature vectors.

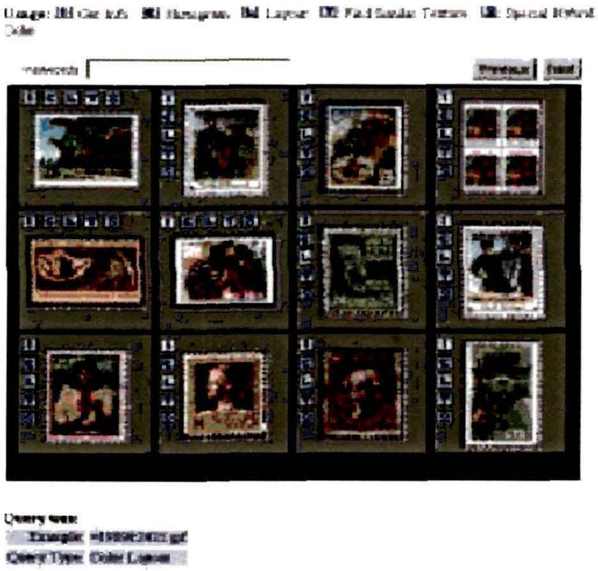
In a query by sketch, after reducing the binary sketch image drawn by the user to size 64×64 a correlation based matching is performed, a kind of template matching. This is done by partitioning the user sketch into 8×8 blocks of 8×8 pixels and finding the maximum correlation of each block of the sketch within a search area of 16×16 pixels in the image database (this is done by shifting the 8×8 block in the search area). This local correlation score is computed on the pixel level using logical operations. The matching score of a database image is the sum of the correlation scores of all local blocks.

Indexing QBIC was one of the first systems that applied multidimensional indexing to enhance the speed performance of the system. The average color and the texture features (both 3D vectors) are indexed using R*-trees. The 18-dimensional moment-based shape feature vector is first reduced using the KL transform and then indexed by using R*-trees.

Result presentation The best matches are presented in decreasing similarity order with (optionally) the matching score aside.

Relevance feedback Any retrieved image can be used as a seed for a subsequent query by example.

Applications At http://wwwqbic.almaden.ibm.com/tmdemo/ is a demonstration of the QBIC system as trademark server.



Query was:
Example: [illegible]
Query Type: Color Layout

Figure 2.9: QBIC.

2.6.14 VIR Image Engine

The VIR Image Engine is an extensible framework for building content based image retrieval systems.

Developer Virage Inc.

URL http://www.virage.com/products/vir-irw.html.

References [JBC96].

Features A basic concept is that of a *primitive*, which denotes a feature's type, computation and matching distance. Five abstract data types are defined: global values and histograms, local values and histograms, and graphs. The VIR Image Engine provides a set of general primitives, such as global color, local color, texture and shapes. Apart from these, various domain specific primitives can be created when developing

an application. When defining such a primitive, the developer supplies a function for computing the primitive's feature data from the raw image.

Querying and Result presentation The VIR Image Engine provides a set of GUI tools necessary for the development of a user interface. These include facilities for image insertion, image query, weight adjustment for re-query, inclusion of keywords, and support for several popular image file formats. Another available component, the query canvas, allows queries-by-sketch; it consists of a bitmap editor where the user can sketch a picture with drawing tools and color it using the colors from a palette. Also, the user can bring onto the canvas an image from an existing collection and modify it using the same drawing tools. Queries can be performed on various user-defined combinations of primitives.

Matching When defining a new primitive, a function for computing the similarity between two sets of feature data previously extracted must also be supplied by the developer. When comparing two images, for each primitive in the current query combination, a similarity score is computed using the distance function defined within the primitive. These individual scores are combined in an overall score using a set of weights in a way characteristic to the application. This score is then stored in a *score structure*, which contains also the individual similarity scores for each primitive. This allows a quick recomputation of the overall score for a new set of weights.

Relevance feedback A sort of relevance feedback is obtained by searching for matches to a fixed query image for different sets of weights.

Indexing The storage of feature data for all primitives is the responsibility of the application developer.

Applications The system was integrated into databases from Sybase, Object Design, and Objectivity, and added as a component to the Oracle DBMS. The AltaVista Photofinder and Illustra's Visual Intelligence system are two applications of Virage technology.

2.6.15 VisualSEEk

Developer Image and Advanced Television Lab, Columbia University, NY.

URL http://www.ctr.columbia.edu/VisualSEEk/.

References [JRS97].

Features In the database population step, each image is automatically decomposed into regions of equally dominant colors. For each region, feature properties and spatial properties are retained for the subsequent queries. A query consists of finding the images that contain the most similar arrangements of similar regions. The color region extraction uses the back-projection technique. The first step is the selection of a color set. This is a 166-dimensional binary vector c, which defines a selection of 166 colors in the HSV color space. From a given image I, another image B is generated by $B[x,y] = \max_j a[k,j]c[j]$, where $k \varepsilon \{0, \ldots, 165\}$is the index of the color of the pixel (x, y) in the image I and $a[k,j]$ is the similarity between two colors (with the indices k and j) in the HSV space. Next the image B is filtered and spatially localized color regions are extracted. Along with the color set used for back-projection, the region centroid, area (defined as the number of pixels in the region) and the width and height of the minimum bounding rectangle are also stored.

Querying To start a query, the user sketches a number of regions, positions and dimensions them on the grid (see Figure 2.10) and selects a color for each region. Also, the user can indicate boundaries for location and size and/or spatial relationships between regions. After the system returns the thumbnail images of the best matches, the user is allowed to search by example using the returned images.

Matching To find the matches of a query image with a single region, queries on color set, region absolute location, area and spatial extent are first done independently. The color set similarity is computed by $d(c_q, c_t) = (c_q - c_t)^t A(c_q - c_t)$, where c_q, c_t are two color sets and $A = (a[i,j])$ is the color similarity matrix. If the user has defined spatial boundaries for the query region, then its distance to a target region is 0 if the target region centroid falls inside this boundaries, and is given by the Euclidean distance between the centroids otherwise. The distance in area between two regions is the absolute value of the difference, while the distance between the minimum bounding rectangles, $(w_q,h_q)$and (w_t, h_t) of two regions is the L_2 metric. The results of these queries are intersected and from the obtained candidate set, the best matching images are taken by minimizing a total distance given by the weighted sum of the four distances mentioned.

If the query image consists of a number of regions, in absolute or relative location, then for each region positioned in absolute location, a query like that described above is made, and for regions positioned by relative location individual queries on all attributes except location are performed. For the intersection of all this query results, the relative spatial relations specified by the user are evaluated using 2D string representation [SKC87].

Indexing The color set distance can be written as $d(c_q,c_t) = \mu_q + \mu_t - 2c^{t}_{q}r_{t}$qrt, where $\mu_q = c^{t}_{q}Ac_q$, μ_t = $c_t^{t}Ac_t$ and r_t = Ac_t. For each target region, μ_t and $r_t[m]$, m ε {0, ..., 165}, are indexed individually. The centroids of the image regions are indexed using a quadtree. For the indexing of the minimum bounding rectangles, R-tree are used.

Result presentation The results of a query are displayed in decreasing similarity order. Under each retrieved image, the distance to the query image is indicated.

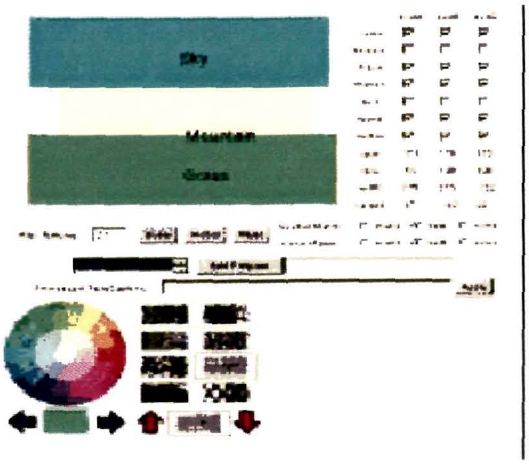


Figure 2.10: VisualSEEk query interface.

2.7 Discussion

In this chapter, we introduced some fundamental techniques for *content-based image retrieval*, including *visual content description, similarity/distance measures, indexing scheme, user interaction* and *system performance evaluation.* Our emphasis is on visual feature description techniques.

General visual features most widely used in content-based image retrieval are color, texture, shape, and spatial information. Color is usually represented by the color histogram, color correlogram, color coherence vector, and color moment under a certain color space. Texture can be represented by co-occurrence matrix, inverse deference moment, entropy, tamura feature, wavelet transform etc. Shape can be boundary-based and region-based and are represented by Fourier Descriptors and Moment invariants, polynomial fitting, Turning Function, Finite Element Method, Chain code, Zernike moments. The spatial relationship between regions or objects is usually represented by a 2D string. In addition, the general visual features on each pixel can be used to segment each image into homogenous regions or objects. Local features of these regions or objects can be extracted to facilitate region-based image retrieval.

There are various ways to calculate the similarity distances between visual features. In this chapter we have discussed some basic metrics; including the Minkowski-form distance, quadratic form distance, Mahalanobis distance, and Cosine Measure.

Efficient indexing of visual feature vectors is important for image retrieval. To set up an indexing scheme, dimensionality reduction is usually performed first to reduce the dimensionality of the visual feature vector. Commonly used dimension reduction methods are PCA, ICA, Karhunen-Loeve (KL) transform, and neural network methods. After dimension reduction, an indexing tree is built up. The most commonly used tree structures are R-tree, R*-tree, quad-tree, K-d-B tree, etc

Image retrieval systems rely heavily on user interaction. On the one hand, images to be retrieved are determined by the user's specification of the query. On the other hand, query results can be refined to include more relevant candidates through the relevance feedback of users. Updating the retrieval results based on the user's feedback can be achieved by updating the images, the feature models, the weights of features in similarity distance, and select different similarity measures.

Although content-based retrieval provides an intelligent and automatic solution for efficient searching of images, the majority of current techniques are based on low level features or current techniques are primarily based on low level features. In general, each of these low level features tends to capture only one aspect of an image property. Neither a single feature nor a combination of multiple features has explicit semantic meaning. In addition, the similarity measures between visual features do not necessarily match human perception. Users are interested in are semantically and perceptually similar images, the retrieval results of low-level feature based retrieval approaches are generally unsatisfactory and often unpredictable. Although relevance feedback provides a way of filling the gap between semantic searching and low-level data processing, this problem remains unsolved and more research is required.

Chapter 3

Background of the Work

3.1 Principles and Human Perception of Color

Color was first "discovered" in 1666 by Sir Isaac Newton as he observed that white light passing through a prism is separated into a continuous spectrum of colored light from violet to red. Since then, science has thoroughly investigated the physical and perceptual properties of color. Color is perceived is light, a form of electromagnetic radiation covering a narrow band of the electromagnetic spectrum between the wavelengths (λ) of 350nm to 780nm. Light originates from self-luminous sources, reflection from objects, or the transmission through a translucent medium. Figure 3.1 shows the distribution of wavelengths in the electromagnetic (EM) spectrum including the spectrum of visible light. Blue light occurs at wavelengths less than red light. Cosmic rays from outer space and radio and television signals form the extreme ends of the EM spectrum.

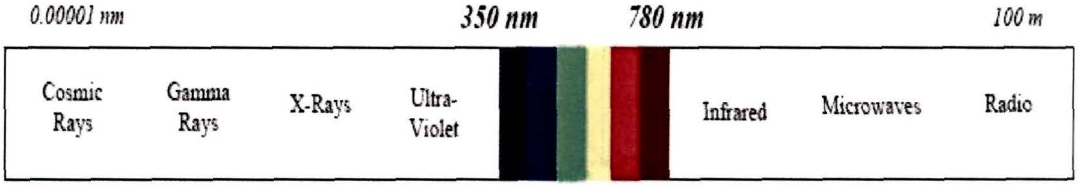


Figure 3.1: The electromagnetic spectrum

The human eye, through the receptors present in the retina called *rods* and *cones*, perceives color as linear combinations of three *primary colors*. These primary colors, red (R), green (G), and blue (B), have specific wavelength values of 700nm, 546.1nm, and 435.8nm,respectively.

Chromatic light is colored light. The basic terms used to describe chromatic light are *hue*, *saturation*, and *brightness*. Hue is used to describe the dominant wavelength or perceived color of an object. It is the "redness" of an apple or the "yellowness" of a banana. Saturation refers to purity of a hue or the amount of white light mixed with a hue or the distance a color is from a gray of equal intensity. Red is highly saturated with white light while pink is not. Brightness is the chromatic analogue of intensity for achromatic light and is practically difficult to measure. Hue and saturation are sometimes combined and referred to as *chromaticity*. Hence a color may be characterized by its brightness and chromaticity.

Given the response functions $f_r(\lambda)$, $f_g(\lambda)$, and $f_b(\lambda)$ for each of the primary colors, the following equation of the electromagnetic response for a wavelength λ is defined as

$$F(\lambda) = Xf_r(\lambda) + Yf_g(\lambda) + Zf_b(\lambda) \qquad (3\text{-}1)$$

The values (X,Y,Z) are called the *tristimulus values* for color F(λ) and denote the respective amounts or red, green, and blue necessary to form a color. Commonly, the tristimulus values are used to specify a color in terms of its trichromatic coefficients

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = \frac{Z}{X+Y+Z} \qquad (3\text{-}2)$$

Tristimulus values are, in general, normalized. Thus, the trichromatic coefficients are likewise normalized. While X, Y, and Z may all be equal to 1, the trichromatic coefficients are subject to the relation x+y+z=1. The primary colors and tristimulus color theory is the mechanism that allows televisions to display the colors we see. Cathode ray tubes (CRTs) have three channels of red, green, and blue. By varying the voltage of each channel and combing their outputs, each pixel on a television screen can output a large array of colors. Secondary colors are specified in terms of the primary colors. Magenta is form from equal amounts of red and blue light. Yellow is formed from equal amounts of red and green light. Cyan is formed from equal amounts of green and blue light. For pigments used for print, the primary and secondary color designations are reversed.

3.1.1 Color Model

The purpose of a color model (also called color space or color system) is to facilitate the specification of colors in some standard, generally accepted way specially in numerical form. In essence, a color model is a specification of the coordinate system and a subspace within that system where each color is represented by a single point. Most color models in use today are oriented either towards hardware (such as for color monitors and printers) or towards applications where color manipulation is a goal (such as creation of color graphics for animation). In terms of digital image processing, the hardware-oriented models most commonly used in practice are the RGB (red, green, blue) model for color monitors and also this model is widely used for digital images and computer graphics i.e. used by softwares. CMY (cyan, magenta, yellow) and CMYK (cyan, magenta, yellow, black) color models are used for color printing. HSI (hue, saturation, intensity), HLS (hue, lightness, saturation), HSV (hue, saturation, value) color models are used by various softwares for computer graphics and digital images, storage and retrieval, manipulation of images etc.

3.1.2 Properties of Color Spaces

The properties [JSS96, GPR96, MKM96] that are most important in color space processing are *uniformity, completeness* and *uniqueness*. A color space is said to be perceptually uniform if the distance between two points in the color space specify how the two colors are similar or dissimilar.

A uniform color space is one in which the metric distance between points in the color space corresponds to the perceptual distance (or similarity) of the points.
A complete color space contains points for all perceptually discernable colors.
A color space is unique if two distinct points in the color space represents two perceptually different colors.

3.1.3 RGB Color Model

Based on the *tristimulus theory* of vision, [Computer Graphics] our eyes perceive color through the stimulation of three visual pigments in the cones of the retina. These visual pigments have a peak sensitivity at wavelengths of about 630 nm (red), 530 nm (green), and 450 nm (blue). By comparing intensities in a light source the color of light is perceived. This theory of vision is the basis for displaying color output on a video monitor using the three color primaries, red, green, and blue, referred to as the RGB color model.

We can represent this model with the unit cube defined on R, G, and B axes, as shown in Figure 3.2. The origin represents black, and the vertex with coordinates (1,1,1) is white. Vertices of the cube on the axes represent the primary colors, and the remaining vertices represent the complementary color for each of the primary colors. The number of bits used to represent each pixel in RGB space is called the *pixel depth.* If in an RGB image each of the red, green, and blue channels are represented by eight bits then that RGB pixel will have a depth of 24 bits (3 image planes times the number of bits per plane).

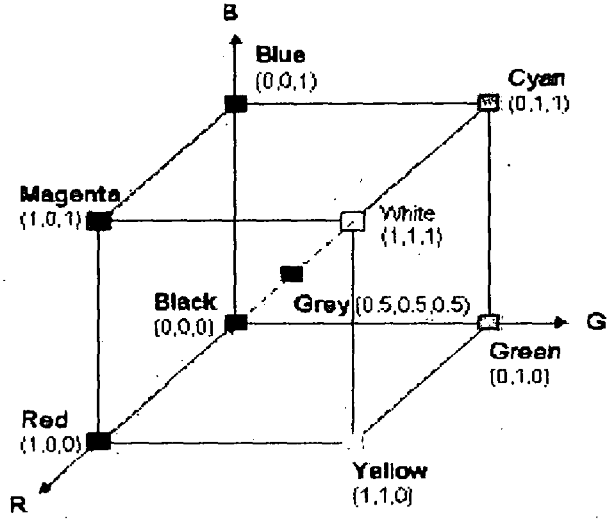


Figure 3.2: RGB color model

3.1.4 CMY Color Model

A color model defined with the primary colors cyan, magenta, and yellow (CMY) is used for describing color output to hard-copy devices. Unlike video monitors, hard-copy devices such as plotters produce a color picture by coating a paper with color pigments. We see the colors by reflected light, a subtractive process.

Cyan can be formed by adding green and blue light. Therefore, when light is reflected from cyan-colored ink, the reflected light must have no red component. That is, red light is absorbed or subtracted, by the ink. Similarly, magenta ink subtracts the green component from incident light, and yellow subtracts the blue component. In the CMY model, point (1,1,1) represents black, because all components of the incident light are subtracted. The origin represents white light. Equal amounts of each of the primary colors produce grays, along the main diagonal of the cube. A combination of cyan and magenta ink produces blue light, because the red and green components of the incident light are absorbed. Other color combinations are obtained by a similar subtractive process.

The printing process often used with the CMY model generates a color point with a collection of four ink dots, like an RGB monitor uses a collection of three phosphor dots. One dot is used for each of the primary colors (cyan, magenta, and yellow), and one dot is black. A black dot is included because the combination of cyan, magenta, and yellow inks typically produce dark gray instead of black.

The conversion from an RGB representation to a CMY representation can be done with the following matrix transformation

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (3\text{-}3)$$

where the white is represented in the RGB system as the unit column vector. Similarly, an CMY color representation is converted into an RGB representation with the following matrix transformation.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \qquad (3\text{-}4)$$

where black is represented as the unit column vector in the CMY system.

3.1.5 HSV Color Model

The HSB, CMY color models are hardware-oriented. By contrast Smith's HSV (hue, saturation, value) model [SMIT78] (also called the HSB model with B for brightness) is user-oriented, being based on the intuitive appeal of the artist's tint, shade and tone. The coordinate system is cylindrical and the subset of the space within which the model is defined is a hexcone, or six-sided pyramid as shown in the Fgure 3.3. The top of the hexcone corresponds to V=1, which contains the relatively bright colors.

Hue or H, is measured by the angle around the vertical axis, with red at 0^0, green at 120^0 and so on. Complementary colors in the HSV hexcone are 180^0 opposite to one another. The value of S is a ratio ranging from 0 on the center line (V axis) to 1 on the triangular sides of the hexcone.

The hexcone is one unit high in V, with the apex at the origin. The point at the apex is black and has a V coordinate of 0. At this point, the values of H and S are irrelevant. The point S=0 and

V=1 is white. Intermediate values of V for S=0 (on the center line) are grays. When S=0, the value of H is irrelevant. When S is not zero, H is relevant.

- *Hue*, the color type (such as red, blue, or yellow)
    - Ranges from 0-360 (but normalized to 0-100% in some application)
- *Saturation*, the "vibrancy" of the color:
    - Ranges from 0-100%
    - Also sometimes called the "purity" of color
    - The lower the saturation of a color, the more "grayness" is present and the more faded the color will appear, thus useful to define de-saturation as the qualitative inverse of saturation.

- *Value* the brightness of the color.
    - Ranges from 0-100%

Conversion from RGB to HSV

Let *MAX* equal the maximum of the (*R*, *G*, *B*) values, and *MIN* equal the minimum of those values.

$$H = \begin{cases} \text{undefined}, & \text{if } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases} \tag{3-5}$$

$$S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{otherwise} \end{cases} \tag{3-6}$$

$$V = MAX \tag{3-7}$$

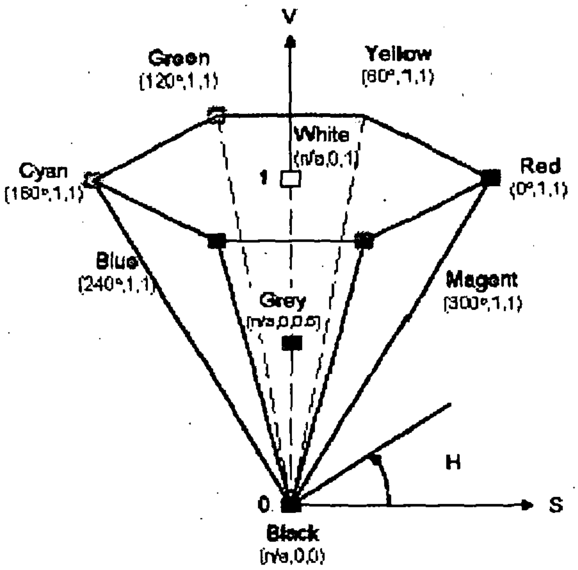


Figure 3.3: HSV Color Model.

3.1.6 HLS/HSI Color Model

The HLS (hue, lightness, saturation) also called HSL (hue, saturation, lightness/luminance) or HSI (hue, saturation, intensity) color model is defined in a double-hexcone subset of a cylindrical space as seen in figure 3.4. Hue is the angle around the vertical axis of the double hexcone, with red at 0^0. the colors occur around the perimeter in the same order as in the HSV color model. In fact, HLS model is a deformation of the HSV model, in which white is pulled upward to form the upper hexcone from V=1 plane. As with the single-hexcone model, the complement of any hue is located at 180^0 farther around the double hexcone, and saturation is measured radially from the vertical axis, from 0 on the axis to 1 on the surface. Lightness is 0 for black (at the lower tip of the double hexcone) to 1 for white (at the upper tip). The grays all have at S=0, but the maximally saturated hues are at S=1, L=0.5.

Conversion from RGB to HLS

$$H = \begin{Bmatrix} \theta & if B \leq G \\ 360 - \theta & if B > G \end{Bmatrix} \qquad (3\text{-}8)$$

with

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \tag{3-9}$$

The saturation component is given by

$$S = 1-\frac{3}{(R+G+B)}[\min(R,G,B)] \tag{3-10}$$

Finally the lightness component is given by

$$I = \frac{1}{3}(R+G+B) \tag{3-11}$$

3.1.7 CIE Uniform Chromaticity Scale Color Spaces

In an effort to define a perceptually uniform color space, the Commission Internationale de l'Eclairage (CIE) developed the L*-u*-v* (CIELUV) and the L*-a*-b* (CIELAB) color spaces based on their Uniform Chromaticity Scale (UCS) diagram (Entropy Hunt,. 1987; Berger-Shunn, 1994). While these color spaces are not strictly perceptually uniform, they are much closed to perceptual uniformity than most other color spaces. The first stage in the definitions of each color space is the linear transformation of the RGB color space to XYZ color space. Each transformation then defines a non-linear transformation from XYZ color space to their respective color space with respect to the chromaticity coordinates of a reference white illuminant with tristimulus values Xn, Yn, Zn.

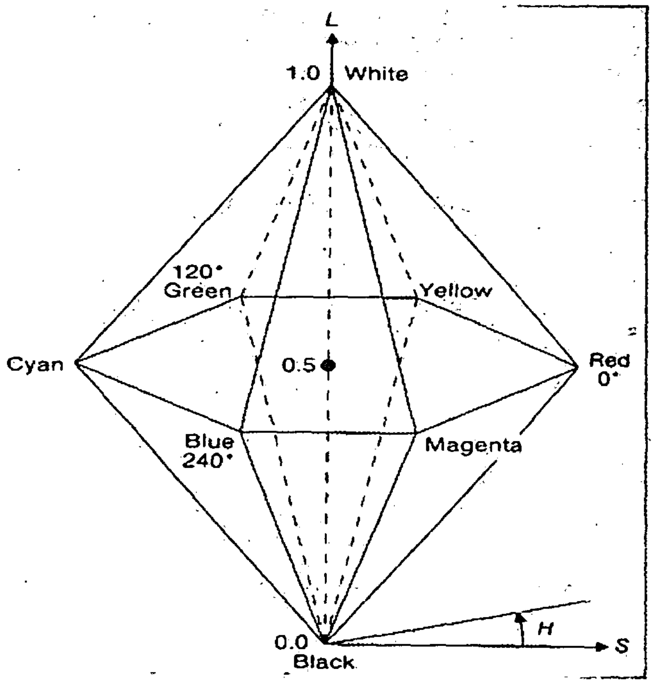


Figure 3.4: HLS Color Model.

The RGB→XYZ transformation given by the matrix

$$T_{RGB-XYZ} = \begin{bmatrix} 0.49 & 0.31 & 0.2 \\ 0.17697 & 0.8124 & 0.01063 \\ 0.0 & 0.01 & 0.99 \end{bmatrix} \qquad (3\text{-}12)$$

was developed in 1931 by CIE to counteract the presence of negative values in color matching functions based on the common RGB tristimulus system (Entropy Hunt, 1987).

The CIE UCS diagram is formed by plotting u' and v', where

$$u' = \frac{4X}{X+15Y+3Z} \qquad v' = \frac{9Y}{X+15Y+3Z} \qquad (3\text{-}13)$$

The CIELUV color space is a three-dimensional space composed of the following values

$$L^* = 116(Y/Y_n)^{1/3} - 16 \qquad \text{for} \qquad Y/Y_n > 0.008856 \qquad (3\text{-}14)$$

$$L^* = 903.3(Y/Y_n) \qquad \text{for} \qquad Y/Y_n \le 0.008856 \qquad (3\text{-}15)$$

$$u^* = 13L^*(u'-u'_n) \qquad (3\text{-}16)$$

$$v^* = 13L^*(v'-v'_n) \qquad (3\text{-}17)$$

The commonly used CIELAB color space is likewise a three-dimensional space determined by the following formula.

$$L^* = 116(Y/Y_n)^{1/3} - 16 \qquad \text{for} \qquad Y/Y_n > 0.008856 \qquad (3\text{-}18)$$

$$L^* = 903.3(Y/Y_n) \qquad \text{for} \qquad Y/Y_n \le 0.008856 \qquad (3\text{-}19)$$

$$a^* = 500[(X/X_n)^{1/3} - (Y/Y_n)^{1/3}] \qquad (3\text{-}20)$$

$$b^* = 200[(Y/Y_n)^{1/3} - (Z/Z_n)^{1/3}] \qquad (3\text{-}21)$$

The color space is cylindrical with L* forming the primary axis and varying from 0 to 100. The coefficients a* and b* may each vary between -80 and +80.

3.1.8 Discussion

All the color models discussed are complete i.e. the contain points for all perceptually discernable colors. But RGB and CMY color models are not perceptually uniform i.e. the matric distance between two points in the RGB and CMY color spaces does not suggest that the two colors are similar or dissimilar. Additionally, the three color color channels of the RGB and CMY color spaces do not vary consistently with one another with respect to brightness. Therefore, the pixels of the images in the image database and query examples must be transformed into an alternative color space that satisfies the properties of uniformity, completeness, and uniqueness.

The HIS, HSV, HSB, HLS family of color spaces share a similar characteristics. The main attraction of these family of color spaces is the separation of the chromaticity (hue and saturation) from luminance (intensity, brightness, and value). Also the transformation of these color spaces from the RGB color space is linear.

The CIE family of color spaces are also perceptually uniform and the conversion of the RGB color space to CIELAB or CIELUV color space is linear. But we have used the HSB model for our experiments as this color model is a built in java package. But to use CIELAB or CIELUV color model we shall have to write codes for it.

3.2 Entropy

The entropy of an image is a measure of the information content of the image. Given a vector **v** of numbers from a set $\{x_1, x_2, \ldots, x_n\}$ where the probability that $x_i \varepsilon v$ is $p_i = P(x_i)$, the entropy of **v** is given by the formula

$$H(v) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{3-22}$$

H(v) is a function of the probability distribution of some random variable and not a function of the actual values the variable may assume.

J. M. Zachary [JMZ00] has used the concept of entropy for index generation of colored images as it has the property of first-order joint probability density functions of colored histograms. Moreover the entropy value of an image is having less dimensions as that of the colored histograms. Thus the entropy of an image is calculated as

$$H(v) = -\sum_{i=1}^{M} v_i \log_2(v_i) \tag{3-23}$$

where v_i is the percentage of pixels in the image *I* which belong to the quantized color *i* and M is the number of different colors present in the image.

The entropy of an image H(v) = 0 implies that the image has all pixel values set to the same value; and H(v) is maximized when all possible colors in the color space of the image are equally represented. Intuitively, this means more information in an image is expressed if the image has more colors than in an image with fewer colors.

The concept of entropy has been utilized in our experiments to determine if quantization of colors is necessary for a given image. In Scheme I (*Chapter 5*) a fixed entropy value is kept. For images having entropy value more than this fixed value quantization is done. In Scheme III (*Chapter 7*) a dynamic entropy value is taken for quantization. This is so because the color content of natural images vary from image to image and hence use of a fixed entropy value to quantize the color content may not be justified.

3.3 Quantization

Color has been recognized as an important visual cue for image and scene analysis. Research work in color analysis has focused on color image formation, color quantization, human visual perception, image segmentation, color-based object recognition, and image database retrieval. The drastic reduction in the number of colors used to represent the color image content that effective and efficient computation of color indices usually demands is in general achieved by color space quantization, using a predefined color palette. Color quantization is sampling of three dimensional color spaces (such as RGB or Lab) which results in a discrete subset of colors known as a *color set* or *color codebook* or *palette*. It is extensively used for display, transfer, and storage of natural images in Internet-based applications, computer graphics, and animation. As a basic technique of color image processing, color quantization plays an important role in many aspects mentioned above.

Generally speaking, color image quantization is divided into four phases [PHC82]: (1) sampling the original image for color statistics; (2) choosing a color map based on the color statistics; (3) mapping original colors to their nearest neighbors in the color map; (4) quantizing and representing the original image. These four steps are inherently connected with each other in which step 1 and step 2 influence the final result to a great extent. A good color quantization technique must consider several factors such as the least distortion, complexity of algorithm, the characteristics of human visual system (HVS) and so forth. Currently, there is not a satisfactory solution to the problem of how to determine the number of colors to be used in the palette while considering human visual perception and the color variation in different image regions.

Palette or color set or color codebook design is to select a small number of representative colors from an image that has high color resolution to form a color set. Using this color set or palette, the high-resolution color image can be represented by replacing the original colors with a smaller set of color elements in the palette. In the past, many color quantization algorithms have been proposed such as median-cut algorithm [PHC82], uniform algorithm [PHC82], octree algorithm [GMP88], center-cut algorithm [JGX93] and clustering-based algorithms such as K-means, adaptive clustering algorithm [ISH00].

The basic idea of median-cut algorithm is to divide the colors of the original image into K boxes in the color space. Each box contains the same number of pixels of the original image. The average color value of each box is used as one of the K colors of the color palette. Colors with high pixel numbers can be included into one box; on the other hand, colors with low pixel numbers can not be represented well.

The uniform quantization algorithm divides the color space into subspaces directly, and chooses a group of colors with evenly distributed red, blue and green components. In this algorithm, the palette colors have no relationship with the colors of the original image. Although this algorithm is simple and fast, since not every image contains all the evenly distributed colors, the final result often differs largely from the original image.

The octree algorithm is based on the agglomerative idea which divides a predetermined subdivision of the RGB color space into levels of octants. Like the median-cut algorithm, this method loses color details. The center-cut algorithm repeatedly splits the color set whose bounding box has longest side until K sets are generated. The centers of K sets are used as palette colors. Clustering-based algorithms usually use the minimal distance as the metric for the clustering. The algorithm in [ISH00] uses a 3D-histogram which is fed into an adaptive algorithm to build the palette. A destined pixel-mapping algorithm is applied to classify pixels into their corresponding palette colors.

A few new or modified algorithms have been proposed in recent years. Zhao [ZYW00] proposed an improvement of K-means algorithm. Atsalakis divided an image into small windows and quantized the major colors of these windows [AAK02]. In [PHC821], an algorithm integrated with Gamma correction was proposed and proved to be efficient to improve the visual effect of quantized images.

Thus quantization of a colored image is performed mainly for two reasons: first, to reduce the color content of an image keeping the size of the image intact, and second, to reduce the size of the image thereby reducing the color content of the image. Quantization for the first case is performed mainly by dividing the color channel into an equidistant number of bins i.e. palettes, and the bin size is kept the same for all images. A particular color of a pixel falling within a certain bin is quantized by the average color of the bin. Scheme I (*Chapter 5*) and Scheme II (*Chapter 6*) uses this type of quantization method.

For quantization of an image as in the second case, a window of fixed size, say 3×3 or 5×5 pixels is considered. The average color or the color which occurs mostly in the window is mapped to a single pixel and thus the new image which is of reduced size than the original image is formed. Both these two cases of quantization can be termed as hard quantization because the bin size and the window size are predefined and they remain the same for all images for which quantization is to be done.

But to get a satisfactory solution to the problem of how to determine the number of colors to be used in the palette while considering human visual perception and the color variation in different image regions we have introduced a method for quantization of colored images known as *Adaptive Quantization* in Scheme III (*Chapter 7*).

3.4 Use of Moment

Moments have been successfully used in many retrieval systems (like *QBIC* [MFH95, WNQ93]). The *first order* (*mean*), the *second* (*variance*) and the *third order* (*skewness*) color moments have been proved to be efficient and effective in representing color distributions of images [MSM95]. Mathematically, the first three moments are defined as:

$$\mu_i = \frac{1}{N}\sum_{j=1}^{N} f_{ij} \tag{3-24}$$

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^2\right)^{\frac{1}{2}} \tag{3-25}$$

$$s_i = \left(\frac{1}{N}\sum_{j=1}^{N}(f_{ij} - \mu_i)^3\right)^{\frac{1}{3}} \tag{3-26}$$

where *fij* is the value of the *i*-th color component of the image pixel *j*, and *N* is the number of pixels in the image.

3.4.1 Silhouette Moments

Silhouette moments refer to moments calculated from a binary image. Binary images are Silhouette images whose intensity takes only two values, viz., 0 and 1 [RMK98]. The pixels on the object region are assigned a value 1, and that on the background region are assigned a value 0. In our experiment the output of the BOO-Clustering algorithm is a set of color clusters and objects based on positional distribution of colors. These color clusters and objects may have sharp or diffused boundaries depending on the type of image. The objects of interest in the image are formed by one or more color clusters from the set of color clusters of the image and it may consist of one or more colors. These color clusters and objects of interest need to be indexed separately using parameters for similarity search. A robust index is one which is translation, rotation, uniform scaling and non-uniform scaling (i.e. aspect ratio) invariant. To obtain such indices, we use silhouette moment [RMK98] of second order in our experiments.

A color cluster consists of a set of pixels denoted by S_q having a color C_p. An object consists of a set of pixels denoted by S_q having a single color or a combination of colors C_p that defines an area in the image with sharp or diffused boundaries. These image segments can be thought of as silhouette images. In our case, a pixel that form a cluster or an object is assigned a value 1, i.e., f(x,y) = 1 and other points are assigned a value 0, i.e., f(x,y) = 0.

For a silhouette image, the point (x_0, y_0) gives the geometric center of the image region. When moments are calculated by shifting the origin of the reference system to the intensity centroid of the image, they are called central moments. This transformation makes the moment computation independent of the position of the image reference system; hence translation invariant.

The *(p+q)*th order two-dimensional geometric moments [RMK98] are denoted by m_{pq} and can be expressed as

$$m_{pq} = \int_x \int_y x^p y^q f(x,y) dxdy \tag{3-27}$$

where $p,q = 0,1,2,3$ ...... is the order of the moment.

The central moment μ_{pq} [RMK98] which is translation invariant is defined by

$$\mu_{pq} = \int_x \int_y (x - x_0)^p (y - y_0)^q f(x,y) dxdy \tag{3-28}$$

$$\Rightarrow \mu_{pq} = \frac{1}{m \times n} \sum_x^m \sum_y^n (x - x_0)^p (y - y_0)^q f(x,y) \tag{3-29}$$

where m, n is the size of the image and p, q is the order of the moment. For silhouette moments f(x,y)=1 where there is a pixel else 0; to make the index scale invariant, let the image be scaled by a uniform factor k, then we have,

$$x' = kx \text{ and } y' = ky \tag{3-30}$$

$$\therefore dx'dy' = k^2 dxdy \tag{3-31}$$

Using equation (3-27)

$$m'_{pq} = \int_x \int_y (x')^p (y')^q f(x,y) dx'dy'$$

$$\Rightarrow m'_{pq} = \int_x \int_y (kx)^p (ky)^q f(x,y) k^2 dxdy$$

$$\Rightarrow m'_{pq} = k^{p+q+2} \int_x \int_y x^p y^q$$

$$\Rightarrow m'_{pq} = k^{p+q+2} m_{pq} \tag{3-32}$$

$$\Rightarrow m'_{00} = k^2 m_{00} \tag{3-33}$$

where m_{00} and m_{pq} are the geometric moments of order 0 and (p+q) and m'_{00} and m'_{pq} are scaled geometric moments of order 0 and (p+q).

Using equation (3-32) and (3-33)

$$\frac{m'_{pq}}{m'_{00}} = \frac{k^{p+q+2} m_{pq}}{k^2 m_{00}}$$

$$\Rightarrow \frac{m'_{pq}}{m'_{00} \times m'_{00}{}^{(\frac{p+q+2}{2}-1)}} = \frac{k^{p+q+2} m_{pq}}{k^2 \times m_{00} \times m'_{00}{}^{(\frac{p+q+2}{2}-1)}}$$

$$\Rightarrow \frac{m'_{pq}}{m'_{00}{}^{(\frac{p+q+2}{2})}} = \frac{k^{p+q+2} m_{pq}}{k^2 m_{00} k^{p+q} m_{00}{}^{(\frac{p+q+2}{2}-1)}}$$

$$\Rightarrow \frac{m'_{pq}}{m'_{00}{}^{(\frac{p+q+2}{2})}} = \frac{m_{pq}}{m_{00}{}^{(\frac{p+q+2}{2})}} \tag{3-34}$$

Hence equation (3-34) is independent of k. Now by replacing the geometric moment m_{pq} and m_{00} by central moment μ_{pq} and μ_{00} we can define

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(\frac{p+q+2}{2})}} \tag{3-35}$$

where η_{pq} is uniform scaling and translation invariant moment of order (p+q).

If an image is transformed with unequal scale factors k_1 and k_2 along x and y axes respectively then

$x' = k_1 x$ and $y' = k_2 y$

$\therefore dx'dy' = k_1 k_2 dxdy$

Using equation (3-27)

$$m'_{pq} = \int_x \int_y (x')^p (y')^q f(x,y) dx' dy'$$

$$\Rightarrow m'_{pq} = k_1^{(p+1)} k_2^{(q+1)} \int_x \int_y x^p y^q f(x,y) dxdy$$

$$\Rightarrow m'_{pq} = k_1^{(p+1)} k_2^{(q+1)} m_{pq}$$

Now,

$m'_{00} = k_1 k_2 m_{00}$ and

$m'_{20} = k_1^3 k_2 m_{20}$ and

$m'_{02} = k_1 k_2^3 m_{02}$

From the above equations we have

$$\frac{m'_{pq}}{m'_{00}} = \frac{k_1^{(p+1)} \times k_2^{(q+1)} \times m_{pq}}{k_1 \times k_2 \times m_{00}}$$

$$\Rightarrow \frac{m'_{pq} \times (m'_{00})^{(\frac{p+q+2}{2}+1)}}{m'_{00} \times (m'_{20})^{(\frac{p+1}{2})} \times (m'_{02})^{(\frac{q+1}{2})}} = \frac{(k_1 k_2 m_{00})^{(\frac{p+q+2}{2}+1)}}{(k_1^3 k_2 m_{20})^{(\frac{p+1}{2})} \times (k_1 k_2^3 m_{02})^{(\frac{q+1}{2})}} \times \frac{k_1^{(p+1)} k_2^{(q+1)} m_{pq}}{k_1 k_2 m_{00}}$$

$$= \frac{m_{00}^{(\frac{p+q+2}{2})} \times m_{pq} \times k_1^{(\frac{p+q+2}{2}+1+p)} \times k_2^{(\frac{p+q+2}{2}+1+q)}}{(k_1^3 k_2)^{(\frac{p+1}{2})} \times m_{20}^{(\frac{p+1}{2})} \times m_{02}^{(\frac{q+1}{2})} \times (k_1 k_2^3)^{(\frac{q+1}{2})}}$$

$$= \frac{m_{00}^{(\frac{p+q+2}{2})} \times m_{pq}}{m_{20}^{(\frac{p+1}{2})} \times m_{02}^{(\frac{q+1}{2})}}$$

where k_1 and k_2 are eliminated. Now by replacing the geometric moments m_{00}, m_{pq}, m_{20} and m_{02} by the respective central moments μ_{00}, μ_{pq}, μ_{20} and μ_{02} we can define η_{pq} as

$$\eta_{pq} = \frac{\mu_{00}^{(\frac{p+q+2}{2})} \times \mu_{pq}}{\mu_{20}^{(\frac{p+1}{2})} \times \mu_{02}^{(\frac{q+1}{2})}} \qquad (3\text{-}36)$$

which is aspect ratio or non uniform scaling and translation invariant moment.

A rotation of an image by an angle θ has an associated pixel coordinate transform given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (3\text{-}37)$$

$$x' = x\cos\theta - y\sin\theta \quad \text{and} \quad y' = x\sin\theta + y\cos\theta \qquad (3\text{-}38)$$

By using equation (3-27) with the condition

$\int\limits_{x}\int\limits_{y} f(x,y)dxdy$ = Constant we have

$$m_{20} = \int\limits_{x}\int\limits_{y} x^2 f(x,y)dxdy$$

$$m_{02} = \int\limits_{x}\int\limits_{y} y^2 f(x,y)dxdy \text{ and}$$

$$m_{11} = \int\limits_{x}\int\limits_{y} xyf(x,y)dxdy$$

Now,

$$m'_{20} = \int\limits_{x}\int\limits_{y} (x')^2 f(x,y)dx'dy'$$

$$\Rightarrow m'_{20} = \int\limits_{x}\int\limits_{y} (x\cos\theta - y\sin\theta)^2 f(x,y)dxdy$$

$$\Rightarrow m'_{20} = \int\limits_{x}\int\limits_{y} x^2 \cos^2\theta f(x,y)dx'dy' - \int\limits_{x}\int\limits_{y} xy2\sin\theta\cos\theta f(x,y)dxdy + \int\limits_{x}\int\limits_{y} y^2 \sin^2\theta f(x,y)dxdy$$

$$\Rightarrow m'_{20} = \cos^2\theta \times m_{20} - 2\sin\theta\cos\theta \times m_{11} + \sin^2\theta \times m_{02}$$

$$\Rightarrow m'_{20} = \frac{1+\cos 2\theta}{2} \times m_{20} - \sin 2\theta \times m_{11} + \frac{1-\cos 2\theta}{2} \times m_{02} \text{ and} \qquad (3\text{-}39)$$

$$m'_{02} = \int\limits_{x}\int\limits_{y} (y')^2 f(x,y)dx'dy'$$

$$\Rightarrow m'_{02} = \int\limits_{x}\int\limits_{y} (x\sin\theta + y\cos\theta)^2 f(x,y)dxdy$$

$$\Rightarrow m'_{02} = \int\limits_{x}\int\limits_{y} x^2 \sin^2\theta f(x,y)dx'dy' + \int\limits_{x}\int\limits_{y} 2xy\sin\theta\cos\theta f(x,y)dxdy + \int\limits_{x}\int\limits_{y} y^2\cos^2\theta f(x,y)dxdy$$

$$\Rightarrow m'_{02} = \sin^2\theta \times m_{20} + 2\sin\theta\cos\theta \times m_{11} + \cos^2\theta \times m_{02}$$

$$\Rightarrow m'_{02} = \frac{1-\cos 2\theta}{2} \times m_{20} + \sin 2\theta \times m_{11} + \frac{1+\cos 2\theta}{2} \times m_{02} \quad \text{and}$$

$$m'_{11} = \int\limits_{x}\int\limits_{y} x'y' f(x,y)dx'dy'$$

$$\Rightarrow m'_{11} = \int\limits_{x}\int\limits_{y} (x\cos\theta - y\sin\theta)(x\sin\theta + y\cos\theta) f(x,y)dxdy$$

$$\Rightarrow m'_{11} = \cos\theta\sin\theta \int\limits_{x}\int\limits_{y} x^2 f(x,y)dxdy + \cos^2\theta \int\limits_{x}\int\limits_{y} xyf(x,y) - \sin^2\theta \int\limits_{x}\int\limits_{y} xyf(x,y)dxdy - \sin\theta\cos\theta \int\limits_{x}\int\limits_{y} y^2 f(x,y)dxdy$$

$$\Rightarrow m'_{11} = \cos\theta\sin\theta \times m_{20} + \cos^2\theta \times m_{11} - \sin^2\theta \times m_{11} - \sin\theta\cos\theta \times m_{02}$$

$$\Rightarrow m'_{11} = \frac{\sin 2\theta}{2} \times m_{20} + \cos 2\theta \times m_{11} - \frac{\sin 2\theta}{2} \times m_{02} \qquad (3\text{-}40)$$

Now

$$m'_{20} + m'_{02} = \frac{1+\cos 2\theta}{2} \times m_{20} - \sin 2\theta \times m_{11} + \frac{1-\cos 2\theta}{2} \times m_{02} + \frac{1-\cos 2\theta}{2} \times m_{20} + \sin 2\theta \times m_{11} + \frac{1+\cos 2\theta}{2} \times m_{02}$$

$$\Rightarrow m'_{20} + m'_{02} = \frac{1+\cos 2\theta + 1 - \cos 2\theta}{2} \times m_{20} + \frac{1-\cos 2\theta + 1 + \cos 2\theta}{2} \times m_{02}$$

$$\Rightarrow m'_{20} + m'_{02} = m_{20} + m_{02} \qquad (3\text{-}41)$$

where m_{11}, m_{20} and m_{02} are geometric moments of order 2 and m'_{11}, m'_{20} and m'_{02} are the rotation invariant geometric moments of order 2

Equation (3-41) is rotation invariant. We define φ_1 and replacing the geometric moments m_{02} and m_{20} by translation and scale invariant moments η_{02} and η_{20} in equation (3-41) we get

$$\varphi_1 = \eta_{02} + \eta_{20} \qquad (3\text{-}42)$$

Similarly it can be shown that

$$\varphi_2 = (\eta_{02} + \eta_{20})^2 + 4\eta_{11}^2 \tag{3-43}$$

$$\varphi_3 = (\eta_{02} \times \eta_{20}) - \eta_{11}^2 \tag{3-44}$$

Thus, φ_1, φ_2 and φ_3 are 2nd order translation, scale and rotation invariant moments.

In our experiment, equations (3-42), (3-43), and (3-44) are utilized for indexing the color clusters and the objects of interest. Hence the indices of the objects of interest and color clusters consists of three and four parameters viz., <φ_1, φ_2, φ_3> and <*color of cluster*, φ_1, φ_2, φ_3> respectively.

Since only 9 (three moments for each of the three color components) numbers are used to represent the color content of each image, color moments are a very compact representation compared to other color features. Due to this compactness, it may also lower the discrimination power. Usually, color moments can be used as the first pass to narrow down the search space before other sophisticated color features are used for retrieval.

3.5 Distance

Cosine measure has been used in our experiments as because it is a very important similarity measure for points in the plane (and in higher dimensions). The cosine measure assigns a high similarity to points that are in the same direction from the origin, zero similarity to points that are perpendicular to one another, and negative similarity for those that are pointing in opposing directions to one another.

$$D(I,J) = \frac{\sum(IJ)}{(\sum I^2)(\sum J^2)} \tag{3-45}$$

3.6 Performance Evaluation

To evaluate the performance of retrieval systems, we have used two measurements, namely, *recall* and *precision* [AMW00], which have been discussed in Section 2.5. For each query, precision and recall are calculated; and for each precision recall pair, points are plotted which forms the Average Precision Recall (APR) curve.

The *area* enclosed by an APR curve and the axes as a performance metric, called *performance area* is used for performance evaluation. Performance area is defined as

$$\frac{1}{2}\sum_{i=1}^{N-1}(x_{i+1} - x_i)(y_{i+1} + y_i) \tag{3-46}$$

Where (x_i, y_i) is the (recall, precision) pair when the number of retrieved images is i and N is the total number of top matches.

Chapter 4

Segmentation

Segmentation is the low-level operation concerned with partitioning images by determining disjoint and homogeneous regions or, equivalently, by finding edges or boundaries. The homogeneous regions, or the edges, are supposed to correspond to actual objects, or parts of them, within the images. Thus, in a large number of applications in image processing and computer vision, segmentation plays a fundamental role as the first step before applying to images higher-level operations such as recognition, semantic interpretation, and representation. Until very recently, attention has been focused on segmentation of gray-level images since these have been the only kind of visual information that acquisition devices were able to take and computer resources to handle. Nowadays, color imagery has definitely supplanted monochromatic information and computation power is no longer a limitation in processing large volumes of data. The attention has accordingly been focused in recent years on algorithms for segmentation of color images and various techniques, often borrowed from the background of gray-level image segmentation, have been proposed. This paper provides a review of methods advanced in the past few years for segmentation of color images.

The desirable characteristics that a good image segmentation should exhibit with reference to gray-level images were clearly stated in [RMH85]: "*Regions of an image segmentation should be uniform and homogeneous with respect to some characteristics such as gray tone or texture. Region interiors should be simple and without many small holes. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged, and must be spatially accurate.*"

A more formal definition of segmentation, accounting for the principal requirements listed above, can be given in the following way [KSF81, NRP93, RJR95]: Let I denote an image and let H define a certain homogeneity predicate; then the segmentation of I is a partition P of I into a set of N regions R_n, $n = 1; \dots ;N$, such that: 1) $\bigcup_{n=1}^{N} R_n = I$ with $R_n \cap R_m \neq \emptyset$, $n \neq m$; 2) $H(R_n)$ = true for all n; and 3) $H(R_n \cup R_m)$ = false for all R_n and R_m adjacent. Condition 1) states that the partition has to cover the whole image; condition 2) states that each region has to be homogeneous with respect to the predicate H; and condition 3) states that the two adjacent region cannot be merged into a single region that satisfies the predicate H.

Segmentation is an extremely important operation in several applications of image processing and computer vision, since it represents the very first step of low-level processing of imagery. As mentioned above, the essential goal of segmentation is to decompose an image into parts which should be meaningful for certain applications [RMH85]. For instance, in digital libraries large collections of images and videos need to be catalogued, ordered, and stored in order to efficiently browse and retrieve visual information [SID96, SIS98]. Color and texture are the two most important low-level attributes used for content based retrieval of information in images and videos. Because of the complexity of the problem, segmentation with respect to both color and texture is often used for indexing and managing the data [MJS91].

In [LLS99, LLS01], an extensive bibliography is reviewed dealing with more recent color image segmentation approaches from which the most interesting ones are succinctly described next. In these surveys, the outstanding feature defining a segmentation is that the decomposition of an image into regions should be significant in relation to the application that is using such results.

These works classify segmentation techniques into three main categories

1. Feature-space based techniques.
    - Clustering.
    - Adaptive k-means clustering.
    - Histogram thresholding.
2. Image-domain based techniques.
    - Split-and-Merge.
    - Region-growing.
    - Graph-theoretical techniques.

  - Edge-based techniques.
  - Neural networks.
3. Physics-based techniques.

4.1 Feature-Space Based Techniques

Color is a constant property of the surface of each object within an image and each pixel of the color image can be mapped into a certain color space. Thus different objects present in the image will manifest themselves as clusters or clouds of points. The spreading of these points within each clusters is mainly determined by color variations due to shading effects and to the noise of the acquisition device. On the other hand, if instead of mapping pixels into color spaces, we build some *ad hoc* histograms upon color features, such as hue, for instance, it is likely that the objects will appear as peaks within these histograms.

Therefore, the problem of segmenting the objects of an image can be viewed as that of finding some clusters, according to the first strategy mentioned above, or as that of finding the peaks of some opportune histograms, according to the second strategy. These two approaches work within a certain feature space, which may be one of the color spaces and they generally neglect the spatial relationship among colors.

4.1.1 Clustering Techniques

Clustering can be broadly defined as a non supervised classification of objects in which one has to generate classes or partitions without any a priori knowledge. Analogous to the definition of segmentation given before in [NPP93], the problem of clustering can be precisely stated as, once given a certain number of patterns, determining the set of regions such that every pattern belongs to one of these regions and never to two adjacent regions at the same time. Classification of patterns into classes follows the general common sense principle that objects within a class should show a high degree of similarity while not across different classes, where they should exhibit very low affinity.

One among the commonest algorithms that have been proposed in the literature of cluster analysis is the k-mean clustering [SHP98], widely adopted in vector quantization and data compression. A fuzzy version of this is commonly used in a number of works referred to in [LLS99, LLS01], as well as the closely related approach of probabilistic clustering. A comparison between crisp and fuzzy versions of that algorithm can be found in [SRR95]. ISODATA is another algorithm often used for color space clustering [KTK99].

Another interesting and fruitful approach is the mean-shift algorithm reported in [DCM97, DCP99, DCP02]. Similarly to the problem of finding function extremes by gradient minimization, color clusters are found in this approach by computing the position in the feature space where the mean value within an image region shows the minimum variation in respect to other neighboring positions. Recently, mean-shift has also been extended to cope with the issue of tracking objects [DCP03].

Competitive learning based on the least-square criterion is employed in [TUM94], whereas the theory of connected components is adopted in [WWC97]. An original technique proposed in [NHC98] adopts the constrained gravitational clustering. Two points within a color space are modeled as two massive particles having an interaction according to the Newton's gravitational law. The net force on each particle determines the collapse of points into clusters whose number is governed by a given force-effective function. Yet, in [KUC94] color space is represented by way of a tree and clustering is achieved by simplification of that tree. This approach is an open door to the introduction of the closely related approach of graph theory, which will be separately reviewed due to its relevance.

Despite it is not included in any of the reviews referred to here, it is our strong believe that another important clustering strategy which recently has gained momentum is the probabilistic model-based approach to unsupervised learning that uses finite mixtures for the statistical modeling of data [AKJ88, AKJ00, GMD00]. Finite mixtures naturally model observations which are assumed to have been produced by one of a set of alternative unknown sources selected at random. Inferring the parameters of these sources and identifying which source produced each observation lead to a clustering of the set of observations. With this model-based (parametric) approach for clustering, opposed to heuristic methods like the aforementioned k-means of hierarchical growing methods [AKJ88], issues like the selection of the number of clusters or the assessments of the validity of a given model can be addressed in a more formal way.

The standard method used to fit finite mixture models is the Expectation-Maximization (EM) algorithm [DLR77, GMT97, GMD00] which converges to a Maximum Likelihood (ML) estimate of the mixture parameters. However, EM for finite mixture fitting is known to have several drawbacks, namely, it is local (greedy), sensitive to initialization and, for a certain type of mixtures, it may converge to the boundary of the parameter space leading to meaningless estimates, apart from the issue of selecting the number of components.

Among the pile of versions and heuristics used to implement the EM algorithm, the work in [MAF02] is outstanding for simultaneously dealing with all the problems mentioned before. An inference criterion is proposed that automatically selects the number of components, greatly unsensitizes EM to initialization, and avoids the finicking problem of reaching the boundaries of the parameter space. More recently, the same authors in [MHC04] propose the concept of feature saliency and introduce an EM algorithm that estimates it as a mixture-based clustering.

Other color image segmentation approaches that use EM are the early work in TYI98] and those of [BCGM98, CBGM02], where the EM process is driven both in color and texture, and is extensively applied to retrieve images from large and varied collections by means of their content. Finally, in [CPP00] instead of using the local iterative scheme, a deterministic annealing EM is proposed to provide a global optimal solution for the ML parameter estimation.

4.1.2 Adaptive k-means Clustering Techniques

A special classification has to be devoted to a class of segmentation algorithms that combines the idea of k-means clustering with the properties of local adaptivity to color regions and of spatial continuity. The aforementioned clustering techniques assign pixels to clusters only on the basis of their color and no further spatial constraints are imposed.

In order to include spatial constraints the work in [TNP92] proposes a generalization of the k-means clustering algorithm which considers the segmentation of gray-level images as a Maximum A Posteriori (MAP) probability estimation problem. The extension of this technique to color images is proposed in [MMC94]. The estimated segmentation is defined as the one that maximizes the posterior probability of the segmentation provided the observed data in the image.

By using Bayes's rule, it is the minimum of the product between the image prior and the conditional probability of the image given a certain segmentation. A Gibbs Random Field (GRF) is used in [SGD84, SZL95] as an image prior to model and enforce spatial homogeneity constraints. Conditional probability is modeled as a multivariate Gaussian distribution with a space-varying mean function.

The algorithm alternates from MAP estimation to local determination of class means, which are initially constant for each region and equal to k-means cluster centers. Interactively, the algorithm then updates those means by averaging them over a sliding window whose size progressively decreases, starting with global estimates and progressively adapting them to the local characteristics of each region.

This algorithm has been further extended in [ESA96], where color image segmentation and edge linking are combined, applying a split &merge strategy to enforce edge consistency. Besides, in [JLR97, JLR98] the algorithm described in [MMC94] is modified to accept in the former a new color space and metric, which is claimed to provide physically more coherent segmentations, and derivative priors combining both region-based and edge-based statistics in the latter.

4.1.3 Histogram Thresholding Techniques

Histogram thresholding is among the most popular techniques for segmenting gray-level images and several strategies have been proposed [KSF81, RMH85, NPP93]. In fact, peaks and valleys in one-dimensional histograms can be easily identified as objects and backgrounds in gray-level images. In the case of color images, things are a little bit more complex since one has to identify different parts of a scene by combining peaks and valleys in three histograms or by partitioning a whole 3D histogram. A common problem with histogram is that noise often gives rise to spurious peaks and thus to segmentation ambiguities. To prevent this, some smoothing provisions are usually adopted.

Usually, pixel color is distributed into three histograms which are independently restricted by thresholds, e.g., by maximizing the within-group variance and combining the three results with a predicate logic function afterwards [MCM98]. In [LSM98] a watershed scheme is adopted to segment either 2D (chromaticity) or 3D histogram from a color image. Histograms are coarsened through convolution with a spherical window to avoid over segmentation. In [DCT95] only hue information is exploited and, therefore, it is suggested a circular histogram thresholding since hue is an angular attribute. Histogram smoothing is achieved by means of a scale-space filter. Other works use the whole HSI color space despite segmentation is undergone through only one coordinate, either hue or intensity. In [CSA97] a fast segmentation algorithm is suggested which resorts to a pre-clustered chromaticity plane after quantization of the HSV space represented into orthogonal Cartesian coordinates.

The work in [KSI96] singles out faces from color images by defining appropriate domains corresponding to skin-like regions within the HSV space. Robustness against changes in illumination and shadows is obtained by disregarding the luminance (V). In [GYM98] an entropy-based thresholding which assumes that patterns in the feature space are generated by two distinct sources, called modes and valleys. First, patterns are classified in either categories by using entropy thresholding and then the number of modes in the feature space is computed employing a modified Akaike's information criterion.

An alternative way of smoothing histograms and achieving better segmentations is by means of fitting a family of curves or density functions to shape observations. Thus, the distribution of the chrominance of the objects in a scene is modeled in [ESA95] as a Gaussian PDF allowing this way an adaptive setting of object-class thresholds. In [LJL94] an adaptive threshold function for both RGB and HSI spaces is devised by using B-splines. Another manner of smoothing hue histograms is suggested in [LLS98] by working with the low-low band of the wavelet transform of the image to be segmented.

4.2. Image-Domain Based Techniques

Almost all the segmentation algorithms of the previous section exclusively operate in some feature spaces. Thus, the regions (segments) they return are expected to be homogeneous with respect to the characteristics represented in these spaces; however, there is no guarantee at all that these regions also show spatial compactness, which is a second desirable property in segmentation applications beside homogeneity. In fact, cluster analysis and histogram thresholding account in no way for the spatial locations of pixels; the description they provide is global and it does not exploit the important fact that points of a same object are usually spatially close due to surface coherence [RJR95]. On the other hand, if pixels are clustered exclusively on the basis of their spatial relationships, the end result is likely to be with regions spatially well connected but with no guarantee that these regions are also homogeneous in a certain feature space.

In the literature of segmentation of gray-level images, a great many techniques have been suggested that try to satisfy both feature-space homogeneity and spatial compactness at the same time [RMH85, KSF81]. The latter is ensured either by subdividing and merging or by progressively growing image regions, while the former is adopted as a criterion to direct these two processes [RMH85, KSF81, ARA82, RJR95]. According to the strategy preferred for spatial grouping, these algorithms are usually divided into split-and-merge and region growing techniques; this distinction may also be extended to the corresponding algorithms for color image segmentation which will be analyzed in the following sections.

4.2.1 Split-and-merge Technique

A common characteristic of these methods is that they start with an initial inhomogeneous partition of the image (usually the initial segment is the image itself) and they keep performing splitting until homogeneous partitions are obtained. A common data structure used to implement this procedure is the quadtree representation [RJR95, STB92] which is a multiresolution scheme. After the splitting phase, there usually exist many small and fragmented regions which have to be somehow connected. The merging phase accomplishes this task by associating neighboring regions and guaranteeing that homogeneity requirements are met until maximally connected segments can be produced. The region adjacency graph (RAG) is the data structure commonly adopted in the merging phase [RJR95, STB92]. In many algorithms, smoothness and continuity of color regions are enforced with the adoption of a Markov Random Field (MRF) [GRC83, SGD84] which basically is a stochastic process characterized by the following property: the conditional probability of a particular pixel taking in a certain value is only a function of the neighboring pixels, not of the entire image. Besides, the *Hammersley Clifford theorem* establishes the equivalence between MRF's and Gibbs distributions [SGD84].

Panjwani and Healey [DKP95] model color texture in RGB components by means of a Gaussian Markov Random Field (GMRF) which embeds the spatial interaction within each of the three color planes as well as the interaction between different color planes. In the splitting phase, the image is recursively partitioned into square regions until each of them contains a single texture described by a color GMRF model. This phase is followed by an agglomerative clustering phase which consists of a conservative merging and of a stepwise optimal merging process.

Liu and Yang [JLY94] define instead an MRF on the quadtree structure representing a color image and use the above mentioned equivalence with a Gibbs distribution. With a relaxation process [ARA82] they control both splitting and merging of blocks in order to minimize the energy in the Gibbs distribution; this is shown to converge to a MAP estimate of the segmentation.

Numerous variations in the split-and-merge strategies have been investigated. In [MCH97] a k-means algorithm is used for both classifying the pixels in the splitting phase and grouping pattern classes in the merging phase. In [VAC97] the splitting is initially performed by segmenting the luminance and then refined by checking the chrominance homogeneity of the obtained regions; the merging is based on an ad hoc cost function. In [SJH98] the splitting is operated with the watershed transform [HDC77] of the gradient image of the luminance component simplified by a morphological gray-scale opening [RJR95, AKJ89, RCG92]; the merging step is realized with a *Kohonen's self-organizing map* (SOM) [STB92]. Shafarenko et al. [LSM97] apply instead the watershed transform to the $L^*u^*v^*$ gradient of images and merge the patches of the watershed mosaic according to their color contrast until a termination criterion is met. A similar splitting approach is adopted in [KSC94] whereas the merging phase is performed by iteratively processing the RAG constructed upon the resulting over segmented regions. Also Round et al. [AJR97] employ a split-and-merge strategy for segmentation of skin cancers; the splitting phase is based on a quad-tree representation of the image and the following conservative merging is performed with a RAG.

Gevers et al. [TGV94, TGA97] believe that split-and-merge algorithms based on a quadtree structure are not able to adjust their tessellation to the underlying structure of the image data because of the rigid rectilinear nature of the quadtree structure; therefore, they suggest replacing it with an incremental Delaunay triangulation [STB92]. A further alternative possibility is to use Voronoi diagrams [STB92] as proposed by Schettini et al. [RSM94] and by Itoh and Matsuda [SII96].

Broadly speaking, we can fit within the class of split-and-merge techniques also some algorithms based upon differential equations and pyramidal data structures. At the first glance, they do not appear to belong to this category since the strategies they adopt to achieve segmentation are rather different from those reviewed so far; but a more careful look into them will bring to light an underlying split-and-merge idea.

The usefulness of pyramidal representation of images for segmentation was pointed out by Burt et. al [PJB81] about two decades ago and ever since a number of methods to segment images by working with pyramids have appeared. It is well-known that pyramids are data structures in which images can be represented at different resolutions (fine-to-coarse) by means of tapering layers recursively obtained by averaging and down sampling their respective underlying layers

[RJR95] (the finest layer at the bottom of a pyramid is the image itself). Thus, father-son relationships can be naturally introduced between adjacent layers of pyramids; segmentation can be achieved with a pyramid-linking process [PJB81] based on a tree data structure where the values of the fathers at a certain high layer are propagated down to the sons of the lowest level. The construction of a pyramid can be regarded as a splitting phase while the subsequent linking process can be seen as a merging phase. Recently, Lozano and Laget [VLB96] have suggested fractional pyramids for segmentation of color images and Ziliani and Jensen [FZB98] have proposed a modified version of the linking approach of [PJB81].

4.2.2 Region Growing Techniques

An homogeneous region of an image may be obtained through a growth process which, starting from a pre-selected seed, progressively agglomerates points around it satisfying a certain homogeneity criterion; the growth process stops when no more points can be added to the region. The region growing techniques are mainly aimed at processing single regions; nevertheless, by combining different and subsequent growth processes, one may agglomerate in regions all the points of an image, obtaining this way its segmentation. After a region growing procedure, there might exist some very small regions or there could be two or more neighboring regions grown at different times exhibiting similar attributes. A common post-processing provision consists therefore in a merging phase that eliminates such instances by generating broader regions.

The region growing can be considered a sequential clustering or classification process [ARA82]; thus the dependence of the results on the order according to which the image points are processed has to be accounted for. The main advantage offered by this kind of techniques is that the regions obtained are certainly spatially connected and rather compact. As for the clustering techniques of Section 4.1.1, where a similar problem arises in the feature space, also for the region growing techniques one is faced with the problem of choosing suitable seed points and an adequate homogeneity criterion.

Tremeau and Borel [ATN97] suggest several different homogeneity criteria operating in RGB coordinates. In a first phase, they generate a certain number of connected regions with a growing process and, in a second phase, they merge all the regions having similar color distributions; after the second phase, the regions have therefore homogeneous colors but they may be disconnected. Kanai [YKI98] develops a segmentation algorithm which resorts to both color and intensity information. The markers (seeds) are extracted from intensity via morphological open-close operations and from color through quantization of the HSV space; joint markers are defined as the sets comprising both kinds of markers. A region growing process based on a watershed algorithm starts from these joint markers. A region merging process eventually reduces the number of segmented regions.

In [BCM97], the initial seeds are generated by retaining the significant local minima of the magnitude of the color image gradient; however, with this algorithm the two following situations might arise: 1) there is more than one seed per region; 2) small objects do not have any seed. The authors devise a procedure for obtaining markers having a one-to-one correspondence with the image regions. The region growing is performed with a watershed-like algorithm proposed by the authors and working on the original color image instead of on a gradient image.

Deng et al. [YDB99] determine a limited number of color classes within an image through color quantization and propose a criterion for "good" segmentation based on them. The application of this criterion within local windows and at multiple scales generates *J-images* in which high and low values respectively correspond to possible region boundaries and to region centers. A region growing method is adopted where the seeds are the valleys of the J-images; the resulting over segmentation is finally removed with a merging phase.

Rehrmann and Priese [VRL98] suggest using a special hexagonal topology in a hierarchical region growing algorithm which results indipendent of the starting point and of the order of processing. Ikonomakis et al. [NIK98] develop an algorithm to segment both gray-scale and videophone-type color images; the procedure is a standard region growing process followed by region merging. Color homogeneity is tested with measurements in the HSI space.

If one define a cluster as a collection of touching pixels that have almost the same color while the change in color is gradual," the fuzzy nature of the segmentation problem can be emphasized. Moghaddamzadeh and Bourbakis [AMN94, AMN97] have adopted this outlook of the problem and advanced two algorithms working in RGB coordinates to implement a region growing strategy for both fine and coarse segmentation of color images. A fuzzy approach for region growing segmentation is adopted also in [TCP96] whose algorithm is based upon several linguistic rules defining relationships among hue, chroma, and intensity. Colantoni and Laget [PBC97] compare the results of four different algorithms obtained by the various combinations of region growing and watershed transform in a pre-segmentation step and in the actual segmentation algorithm. Images are represented in $L^*a^*b^*$ coordinates and handled by means of RAG's and contour graphs.

4.2.3 Graph Theoretical Techniques

Another interesting approach is the one based on graph theory. The goal here is to partition a graph describing the whole image into a set of connected components that correspond to image regions. There are at least two ways of doing so. On the one hand, there are splitting methods that partition a graph by removing superuous edges. On the other hand, *region-growing* methods join components as a function of the attributes of nodes and edges. Next, some of these graph-partitioning approaches are briefly described.

The most efficient graph-based algorithms use fixed thresholds and purely local measures to find regions. For example, the approach in [CTZ71] is based on breaking large edges in a Minimum Spanning Tree (MST) of the graph. A more recent method [ZWR93] is based on the computation of the minimum cut in the graph representing an image. The cut criterion is designed to minimize the similarity between regions that are being split. This approach captures non local properties of the image but requires more than nearly linear time, in contrast with the more efficient methods described bellow that are based upon local information. Other refinements of such methods can be found in [JSJ97, JSS98], where a normalized version of the minimum cut is computed. For a wider review on this sort of approaches, refer to the works in [UEG97, POF98].

In [RUG97] a measure of local variability is employed to decide which edges to remove. Local measures just rely upon the nearest neighbors of points and are not enough to get a reasonable glimpse of the whole image variability since they do not capture non local properties. This issue is specifically tackled in [PFF98]. Another graph-theoretical work in [JPW98] presents computationally "inexpensive" algorithms for probability simulation and simulated annealing, such as Hastings's and generalized Metropolis's algorithms. To reduce the computational burden, a hierarchical approximation is proposed minimizing at each step a cost function on the space of all possible partitions. Some other methods use more sophisticated models such as MRF [SGD84], but they tend to be quite inefficient in terms of computational time.

It is important to state that numerous works take advantage of MSTs as a mean to reduce the inherent algorithmic complexity. In [TVA93] vertexes connected by the smallest weight edge are melted by an iterative process. At the end, the MST formed at each step is further split by removing edges bearing the highest weight while generating a hierarchy of partitions. In [YXE97] a MST is built up using the Kruskal's algorithm to find a partition minimizing a cost function afterwards. This is accomplished with a dynamic approach and diverse heuristics to further reduce the algorithm complexity.

The approach in [PFF98] is even more drastic, combining both region-growing and Kruskal's routine. Despite in [JSJ97] it is argued that in order to capture non local image properties a segmentation algorithm should start with large image regions and split them afterwards, rather than starting with small image regions and then merging them, [PFF98] proves that a region-merging algorithm can as well produce segmentations from non local image properties by a bottom-up scheme.

4.2.4 Edge Based Methods

Segmentation may also be obtained by detecting the edges among regions as it was extensively investigated for gray-level images [NPP93], from where it is well-known that edges can be found by using functions approximating the gradient or the Laplacian of an image, which are of course scalar functions. The problem encountered in color images is that of finding a counterpart of gradient functions for color images. This can be basically defined at least in two ways, namely, by embedding in a single measure the variations of all the three color channels, or by computing the gradient of each single channel and combining them accordingly to a given criteria afterwards.

The first approach requires some basic concepts of differential geometry such as the first fundamental form. Its eigenvectors provide the direction of maximal and minimal change while its eigen values provide the corresponding rates of change. The chromatic edge detectors in [MCA97] for vector-valued functions is based upon this metric. Numerous examples of the latter approach are given instead in [LLS99, LLS01], e.g., different combinations of gradients of hue, saturation, and intensity computed in HSI coordinates, or finding clusters in the RGB space and computing edges as the transitions from one cluster to another.

A truly original algorithm for boundary detection is proposed in [WYB97]. They use a kind of predictive coding model to identify the direction of change in color and texture at any point and at a given scale. This gives rise to an edge flow which, through propagation, converges to the image boundaries. There are several arguments in favor of hue as the most important color attribute for segmentation. In particular, the work in [FPC94] demonstrates that, if the integrated white condition holds, hue is invariant to certain kinds of highlights, shadings, and shadows. Edge detection is then achieved by finding the zero crossings of the convolution over the hue image with a suitable Laplacian function, as in the gray-level case. Nevertheless, the poor behavior of hue near small values should be taken into account in that situation. Neural networks in the form of Kohonen's self-organizing maps can also be used for contour segmentation, as reviewed in [LLS99, LLS01].

The framework for object segmentation based on color snakes or active contours, originally proposed in [MKA87], can also fit within the context of edge-based techniques. The classical snakes approach consists in deforming an initial contour towards the boundary of the detected object. Deformation is obtained by minimizing a global energy such that its local minimum is attained at the boundary of the object. Formulation of active contours for vector-valued images and, therefore, for color images, due to Sapiro [GSV96, GSC97], is based on a Riemannian metric which captures information from all image components. Instead, color-invariant snakes that use color-invariant gradient information to drive the deformation process are proposed in [TGS98]. In this way, snakes return region boundaries pretty insensitive to disturbances due to shadowing, shadows, and highlights. Papers in [GSV96, GSC97] likewise show a close relation existing between active contours for color images and other algorithms based on frameworks such as partial differential equations, anisotropic diffusion, and variational approaches to image segmentation.

4.2.5 Neural Network Method

Finally, in [LLS99, LLS01] it is cited the class of image segmentation techniques adopting a classification based on neural networks. It is well-founded that neural networks are structures made up of a large number of elementary processors massively interconnected performing simple functions each. Despite their complexity, neural networks offer two important properties in pattern recognition tasks, namely, high degree of parallelism, which allows very fast computational times and makes them suitable for real-time applications, and good robustness to disturbances, which provides reliable estimates.

Another interesting feature is that, in the case of image segmentation, neural networks permit accounting for spatial information. On the other hand, in most kind of networks the final number of segments within an image must be known beforehand and run a preliminary learning phase to train the network to recognize patterns. Usually segments are derived with some a priori knowledge about the problem or in a preprocessing stage.

A number of algorithms were already proposed in [NPP93] for segmenting gray-level images by means of neural networks. What is new in the reviews in [LLS99, LLS01] is that the discussion on neural-network based techniques is just offered in the field of segmentation of color images.

The authors in [PCD97] present two algorithms based on the idea of regarding segmentation as the problem of minimizing a suitable energy function for a Hopfield network. The first algorithm consists of three different networks, each dedicated to a color feature, combining the results afterwards. The second algorithm consists instead of a single network which classifies image pixels into the classes obtained by a preliminary histogram analysis in the color space. Other slightly different versions are also cited in [LLS99, LLS01], e.g., one using a pre-classification algorithm to spot out some regions of interest in biomedical images and another one applying an active-region segmentation algorithm.

It is important to state that this kind of techniques is optimal whenever the specific classification problem is well understood and the number of possible classes is beforehand known. This is quite the case both in medical applications and in the issue of human face localization by means of color segmentation.

An example for the latter is the work in [HAR98], where a retinally connected neural network examines small windows of an image and decides whether each window contains a face. The system arbitrates between multiple networks to improve the performance over a single network and a bootstrap algorithm is employed for training. False detections are added into the training set as training progresses in order to eliminate the need of a manual selection of negative training samples, which must be chosen in order to span the entire space of non face images.

In [HCF00] a neural network-based scheme for human face detection and eye localization in color images under an unconstrained scene is presented. A Self-growing Probabilistic Decision-based Neural Network (SPDNN) is used to learn the conditional distribution for each color class. The paper demonstrates a successful application of SPDNN to face detection and eye localization on a populated database as well as a good processing speed.

Regarding the field of medical images, a swell of algorithms has also been proposed dealing with the segmentation of color images. For instance, an algorithm for medical stained images is presented in [HON94], where three are the possible classes represented by three different colors. They suggest a three-layered neural network as the input layer and the three desired classes as the output layer.

In this regard, it is very common in neural networks the adoption of three layers since this structure is capable of implementing arbitrarily complex decision surfaces composed of intersecting hyper-planes in the pattern space. Classical also are the learning phases obtained with a back propagation routine as in [SNK97]. Similarly, the paper in [NFF94] uses two three-layered neural networks along with the learning through back propagation to separate cells from background in medical images.

In [MSR99] an unsupervised approach using Hopfield neural network is presented for the segmentation of color images of stained liver tissues. As in [PCD97], the segmentation problem is formulated then as the minimization of an energy function, with the addition of some conditions to reach a status close to the global minimum in a pre-specified time of convergence.

Recently, an efficient and accurate tool for segmenting color images has been proposed in [DGM02] grounded on a cluster-based approach to train very large feed-forward neural networks. This paper shows a great potential in applications where the accuracy is the major factor, specially in the area of medical imaging, where segmentations must provide the highest possible precision.

4.3. Physics Based Methods

All the algorithms examined so far are certainly prone to segmentation errors if the objects portrayed in the color images are affected by highlights, shadowing, and shadows. These phenomena cause the appearance of color of uniformly colored surfaces to change more or less drastically, whence those algorithms are very likely to return over segmented regions. The only way to overcome this drawback is to analyze how light interact with colored materials and to introduce models of this physical interaction in the segmentation algorithms. This motivates the name of physics based techniques given to them. The mathematical tools they use do not significantly differ from those adopted by the algorithms of the previous section; the major difference with respect to those is the underlying physical model accounting for the reflections properties of colored matter.

Colored materials may be divided into three main categories: *optically inhomogeneous dielectrics*, *optically homogeneous dielectrics*, and *metals*. A milestone in the field of physics based segmentation was laid by Shafer in [SAS85] where he introduces the *dichromatic reflection model* for inhomogeneous dielectrics. This model is defined by $\zeta(\lambda; g) = \zeta_s(\lambda; g) + \zeta_b(\lambda; g) = m_s(g)c_s(\lambda) + m_b(g)c_b(\lambda)$ and states that the total radiance $\zeta(\lambda; g)$ of the light reflected by an inhomogeneous dielectric is given by the sum of two independent parts: the radiance $\zeta_s(\lambda; g)$ of the light reflected by the object's surface and the radiance $\zeta_b(\lambda; g)$ of the light reflected from the underlying object's bulk. Symbol g denotes dependence on geometric parameters while λ is the wavelength. Moreover, the dichromatic reflection model states that each of the previous components can be split into a pure geometric coefficient $m(g)$ independent of wavelength and into a relative spectral power distribution $c(\lambda)$ that depends on wavelength but not on geometry. Shafer proves that in a color space such as the RGB the dichromatic reflection model simply reads $C_\zeta = m_sC_s + m_bC_b$, where C_ζ is the color (pixel value) measured, ms and mb are the magnitudes of reflection at the considered point, and C_s and C_b are the colors of interface and body reflection of the material. This model may effectively explain some particular shapes of clusters in the color space. Based upon this model, Klinker et al. [GJK90] set up an algorithm (using either a split or a region-growing strategy) which makes some optical hypotheses relating objects' colors, shading, and highlights and try to justify with them
the cluster shapes. The main limitation of this technique is that it can be applied only to inhomogeneous dielectrics.

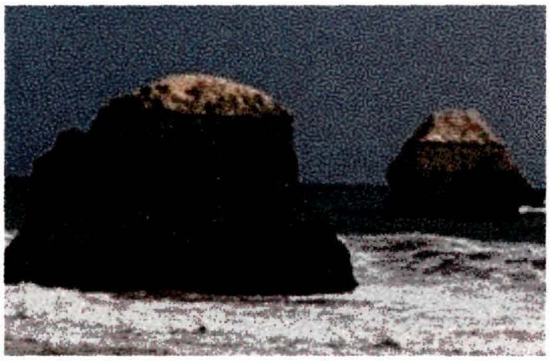


Figure 4.1(a). Natural Image

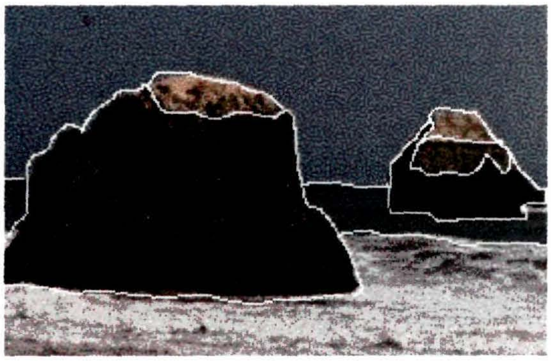


Figure 4.1(b). Natural Image (Segmented)

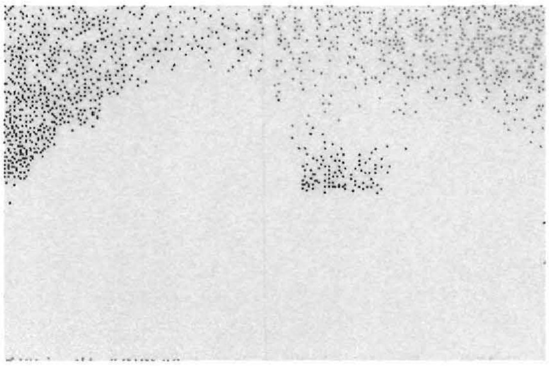


Figure 4.1(c). Cluster 1

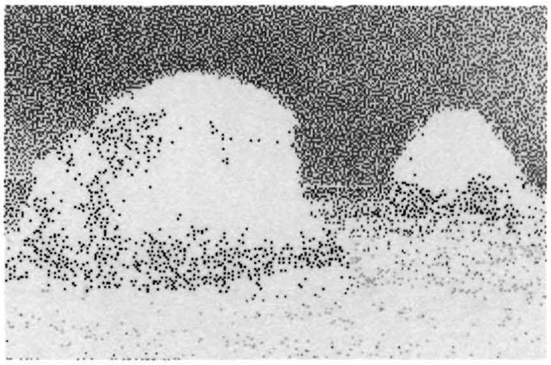


Figure 4.1(d). Cluster 2

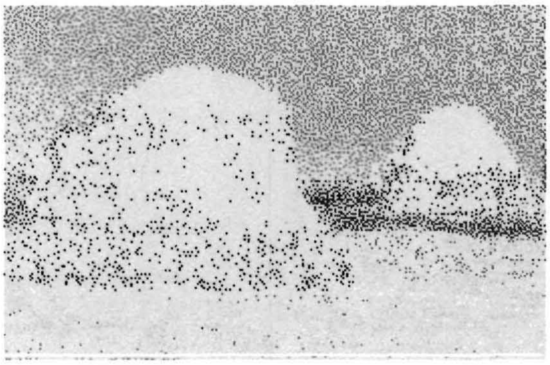


Figure 4.1(e). Cluster 3

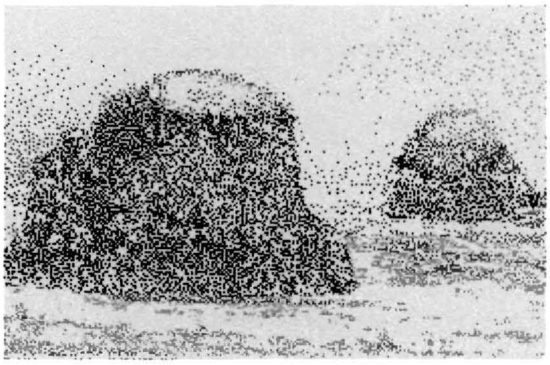


Figure 4.1(f). Cluster 4

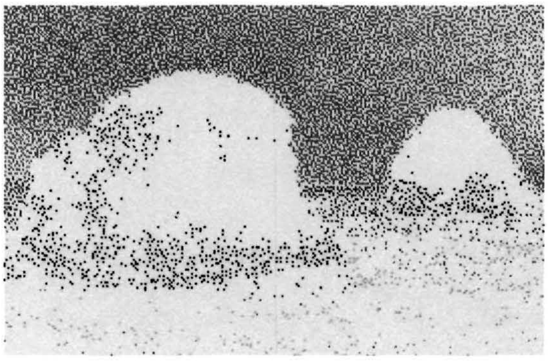


Figure 4.1(g). Cluster 5

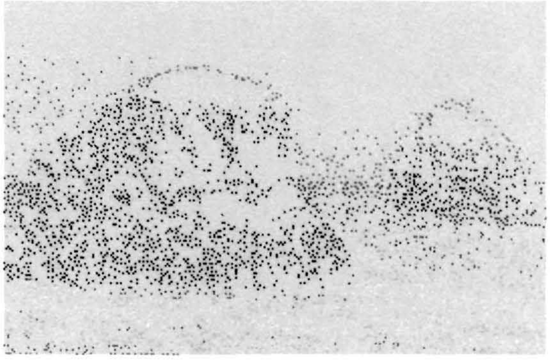


Figure 4.1(h). Cluster 6

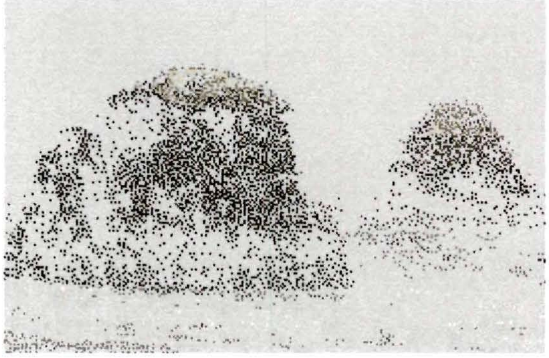


Figure 4.1(i). Cluster 7

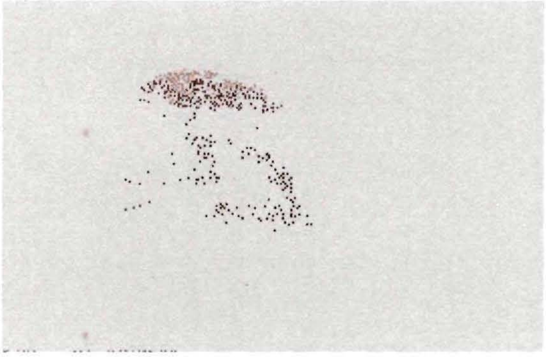


Figure 4.1(j). Cluster 8

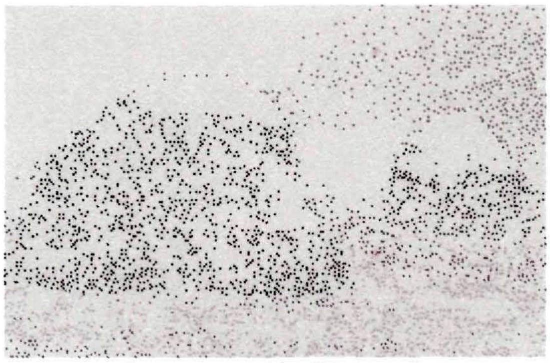


Figure 4.1(k). Cluster 9

4.4 Discussion

From the above discussion we have come to know that segmentation is the process of partitioning an image into disjoint and homogeneous regions or by finding edges or boundaries between different color regions. The homogeneous regions or the edges correspond to actual objects or parts of objects within the image. The regions should be spatially well connected i.e. having spatial relationship and homogeneous in a certain feature space such as color or texture. The edges define a sharp boundary between objects or parts of objects. Adjacent regions of a segmentation should have significantly different values.

A natural image is constituted of many different color distribution that forms segments. This distribution of color overlaps or diffuses with each other and cannot be demarcated by a fine edge or boundary separating them. In this case, one of the property of segmentation, i.e. disjoint color regions cannot be achieved. Another important property of segmentation is connectedness of homogeneous (say same colored pixels) regions. In this case let a particular colored pixels be concentrated in two different places of the image quiet a distant apart such that they can not be put in the same segment due to a distance factor. These two anomalies are taken care of in the methods discussed in Image-Domain Based Techniques. But none of those techniques operate in linear time.

Figure 4.1(a) depicts a natural image and Figure 4.1(b) depicts a segmented image of image Figure 4.1(a). Figure 4.1(c) to Figure 4.1(k) depicts the clusters that forms the objects in the image of Figure 4.1(a). It can be clearly seen that the color clusters that forms the objects does not have a sharp boundary between color regions demarcating an edge to form a segment. To get a smooth boundary between segments a smoothing algorithm has to be run which may detoriate the segment formation. Hence it will be less erroneous if we choose color clusters that forms the objects or parts of the objects. In our example image Figure 4.1(a), the combination of clusters Cluster 4, Cluster 6, Cluster7 and Cluster 8 forms the *hill object* of the example image. This hill object does not have a sharp boundary demarcating it from the background.

We have developed a sliding window based technique which has been discussed in the next chapter to identify color clusters of any shape within the images and it operates in linear time.

Chapter 5

BOO-Clustering technique for Spatial Data

This chapter presents a new three dimensional numeric data clustering technique. This technique of data clustering is applied to images to find out the color clusters of any shape present in an image efficiently. The technique exploits the Density/Neighborhood concept in identifying the clusters in a single pass. The choice of the name of the algorithm as "*BOO-Clustering*" has been discussed in section 5.4.

5.1 Spatial Data Clustering Using Density/Neighborhood Approach

Considering the color content and position of each pixel, an image can be characterized by a set of color clusters and objects of interest of arbitrary shapes. To identify the color clusters in an image, any density-based clustering technique can be applied.

In density based clustering, a *cluster* is defined as a connected dense component, growing in any direction where density leads. Several useful clustering techniques [RMK98], [CCM99], [WYJ97], [CBT03], [ABH01], [YDB01], [DCP97] have been proposed based on this concept. However, most of these algorithms are found to be costly in terms of execution time. We call our clustering technique BOO-Clustering, which is a variant of a color segmentation technique, operating in *linear time*.

In the next section we discuss the BOO-Clustering algorithm in details.

5.2 The BOO-Clustering Algorithm

As discussed already, we have seen that a colored image consists of distribution of colored pixels. This distribution is sometimes compact forming a segment and sometimes diffused i.e. not compact and distributed through out the image. In the first case, the color segments have a definite shape and a boundary. But in the second case the color clusters have a definite shape but a boundary can not be given and hence such type of distribution can be called *fuzzy distribution of colors. Both these two type of distributions of color clusters has to be identified by the* clustering algorithm. GDBSCAN [JSM00] (Generalized Density Based Spatial Clustering of Applications of Noise) and BOO-Clustering (BOOMERANG Clustering) algorithm which have been developed in house, detects these two types of clusters. GDBSCAN is an established algorithm for finding clusters within a data set having three dimensions. Our colored image is having a three dimensional data set i.e. each colored pixel is identified by its position (x, y value) and color. Hence GDBSCAN can be applied to image data; but it has some drawbacks and hence a fast and less complicated three dimensional data clustering technique had to be developed and we have named it as "*BOO-Clustering*" spatial data clustering technique.

To determine a pixel that belongs to a particular cluster, both the algorithms look into the neighborhood of that pixel. The neighborhood is generally defined by a distance parameter which is normally given a fixed value i.e. a threshold. In GDBSCAN, the neighborhood may consist of both clustered and not clustered pixels. But in BOO-Clustering the neighborhood consist of only clustered pixels. Clustered pixels are those pixels which have been marked as belonging to a particular cluster and not clustered pixels are those pixels which are yet to be marked as clustered. That is where the difference between the two clustering algorithms lies.

Also in another modified version of the BOO-Clustering algorithm the input distance parameter need not be given at the time of clustering process. This version of the algorithm always produces a constant number of clusters and returns the same distance parameter value irrespective of number of execution of the algorithm on the same data i.e. an image.

On experimentation, the BOO-Clustering algorithm has been found to be less complicated, fast and robust in respect of detection of color clusters of any shape within the image. GDBSCAN is also robust in respect of color cluster detection. It has also been found that with different input distance parameter to both the clustering algorithms produces nearly the same number and same shaped color clusters. The present version of BOO-Clustering algorithm works on three dimensional numeric data and hence it can be applied to various other applications. The GDBSCAN algorithm is not discussed here as it is an already established algorithm. The next section discusses the BOO-Clustering algorithm in details.

5.3 Definitions Needed for BOO-Clustering Technique

The description of the BOO-Clustering technique requires the following definitions:

Definition 1: *Unclassified Pixels* (u) are those pixels which are not yet clustered. *Classified pixels* (c) are those pixels which belong to a particular color cluster.

Definition 2: *Target Pixel (t)* is a pixel which is in hand to be classified.

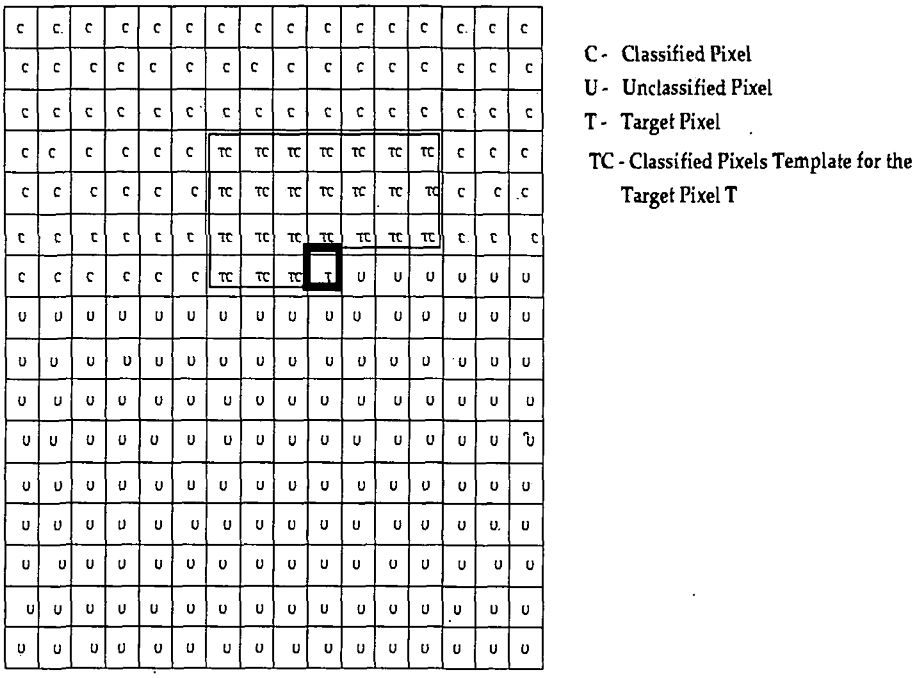
Definition 3: A template (*T*) of a target pixel *t* at the position (*x*, *y*) is a set of pixels which are already classified and are at a neighborhood distance *d* from *t*. The mathematical model is defined in the next section. A sample template (*T*) is shown in Figure 5.2.

Definition 4: *Pixel Threshold* (*PT*) is the minimum number of classified pixels required in a cluster so that it does not become a *Noisy Cluster*. Experimentally, it has been found that a cluster with fewer than the number of pixels in the template does not carry useful meaning. Hence, such noisy clusters are discarded.

5.4 BOO-Clustering: A Novel Clustering Technique

Figure 5.1 shows an image where each block represents a pixel. Here we do not show a color in each block, but some blocks are marked classified (C) i.e. belongs to a particular cluster, some are marked unclassified (U) i.e. not yet clustered. One block is marked as a target pixel (t) which is in hand to be clustered. Some are marked as pixels from a template (TC) a set of pixels which are already classified and are at a neighborhood distance *d* from t. The process of clustering starts from position (0,0) of the image and successively proceeds till the last pixel is classified. After completion of clustering, each pixel is assigned a *Cluster Id.* BOO-Clustering assigns a Cluster Id to a target pixel (t) by searching in the template (T) of the target pixel for similar colored pixels. The template (T) of the target pixel (t) consists of a set of pixels (TC), which are already classified, i.e., pixels which have already been assigned a Cluster Id. Three cases may arise after the search operation. First, not a single pixel is found which is of the same color as that of the target pixel. In this case, a new Cluster Id is assigned to the target pixel (t). It implies that there is no single cluster with the same color as that of the target pixel nearby. Second, if some or all pixels have the same color as that of the target pixel and those pixels have a common Cluster Id, then it assigns that common Cluster Id to the target pixel (t). This implies that there is a single cluster with the same color as that of the target pixel nearby. Third, if some or all pixels have the same color as that of the target pixel, but bear different Cluster Ids, it implies that there is more than one cluster with the same color as that of the target pixel. But, as they appear in the template of the target pixel, they should belong to the same cluster. Hence, in such a situation the algorithm assigns any one of the found Cluster Ids to the target pixel and merges those found color clusters of the template with different Cluster Ids to one cluster. While merging, the algorithm traverse back re-clustering successively, pixel by pixel, till it reaches the first pixel or does not find any cluster in the template of the backtracking pixels. After assigning a Cluster Id (new or already assigned) to the target pixel, the algorithm takes the next pixel as the target pixel and starts the process of clustering.

The BOO-Clustering algorithm proceeds sequentially from the first pixel till it encounters the last pixel. In the step of execution, if the third case arises, the algorithm backtracks to the first pixel for merging of clusters (re-clustering) and then comes back to the position where it has left and starts clustering from the immediate next pixel. This is the heart of the algorithm which works like a BOOMERANG. Hence the name BOO-Clustering.



C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
C	C	C	C	C	C	TC	TC	TC	TC	TC	TC	TC	C	C	C
C	C	C	C	C	C	TC	TC	TC	TC	TC	TC	TC	C	C	C
C	C	C	C	C	C	TC	TC	TC	TC	TC	TC	TC	C	C	C
C	C	C	C	C	C	TC	TC	TC	T	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

C - Classified Pixel
U - Unclassified Pixel
T - Target Pixel
TC - Classified Pixels Template for the Target Pixel T

Figure 5.1. A Sample Template Shown in an Example Image.

5.5 Data Structure/Symbols Used for BOO-Clustering Algorithm

Image(m, n)	: Array to hold spatial color data of an input color image where *m* is the number of rows & *n* is the number of columns in the array.
Cluster(m, n)	: Array to hold the cluster id of the pixels in the *Image*.
d	: Number that defines the size of the *Template*.

Template (*T*) : $T = \{P((x\text{-}i), (y\text{-}j))$ if $x\text{-}i>=0$ and $y\text{-}j>=0$ ($i=1,2,3,..d; j=0,1,2,....d$) $P((x+i), (y\text{-}j))$ if $x+i<=n$ and $y\text{-}j>=0$ ($i=0,1,2,..d; j=1,2,3...d$)$\}$ defines the size of the template for a *Target_pixel P.*

total_clusters : Variable that holds the total clusters detected.

```
Algorithm BOO-Clustering(Image(m,n), d)
begin
    total_clusters = 0;
    for row = 1 to m
    begin
            for col = 1 to n
            begin
                if (row ==1 ) and (col == 1) Cluster(row, col) = 1;
                else Search_in_template(row, col, Image(row,col));
                endif
            end;
    end;
    return(total_clusters);
end;

Search_in_template(row,col, Target_pixel)
begin
        Search in Template for Pixels having the same color value as that of Target Pixel;
        Case 1: Not a single pixel found;
                assign cluster(row,col) with a new Cluster id;
                total_clusters = total_clusters + 1;
        Case 2: Some pixels found and they all have same Cluster id;
                assign cluster(row,col) with the found Cluster id;
        Case 3: Some pixels found and they have different Cluster ids;
                assign to the cluster (row,col) with the smallest Cluster ids;
                replace the other Cluster ids having the same pixel values as that of the target
                pixel, cluster(row,col), with the smallest Cluster id;
end;
```

5.6 Complexity Analysis

The original BOO-Clustering algorithm is of order $\theta(N^2)$. To make it linear, instead of backtracking, any one of the Cluster Id's is assigned to the target pixel and a link of Cluster Ids is maintained as a list for those Cluster Ids whose pixel values are the same as that of the target pixel but have different Cluster Ids. Once the clustering is over, the algorithm scans the list to reassign a common Cluster Id to those clusters which are connected. Hence the time complexity of the algorithm is $\theta(N \times 2(d^2 + d))$ where d is a number that defines the size of the template.

Here, d is usually very small as compared to N and hence can be neglected. Also, the time taken for scanning the list is smaller compared to N. Thus, the complexity of BOO-Clustering is $\theta(N)$.

If d is large, then $2(d^2 + d)$ part of the complexity $\theta(N \times 2(d^2 + d))$ cannot be neglected and the complexity of the algorithm does no longer remains linear. d can go up to m or n whichever is the bigger; the complexity becomes $\theta(N \times 2(m^2 + m))$, i.e., $\theta(N \times N)$, i.e., $\theta(N^2)$. To avoid this problem if the third case arises; the algorithm assigns the smallest Cluster Id of the found Cluster Ids to the target pixel and assigns that smallest Cluster Id to those pixels having the same color as that of the target pixel and bearing different Cluster Ids. Thus the problem of back tracking or maintaining a list of connected cluster ids for the purpose of re-clustering has been removed. Hence the complexity of BOO-Clustering algorithm becomes $\theta(N)$ after this modification.

5.7 Performance Study

The proposed algorithm was implemented in a P-VI machine with 256 MB RAM in java platform. The effectiveness of the algorithm was tested over a downloaded dataset on facial and other images. Figure 5.2 depicts some clusters and objects detected by this algorithm of a facial image.

The proposed clustering algorithm was compared with some of its popular counterparts, and the comparison results are reported in Table 5.1 below. From the table, it can be easily seen that the proposed BOO-Clustering algorithm is equally good with Fuzzy k-means from the complexity point of view. However, it can detect clusters of all shapes, whereas Fuzzy k-means has limitations in this regard.

Table 5.1: Comparison of BOO-Clustering with its counterparts

Algorithm	Shapes	Outliers/Noise	Order Dependency	Complexity
GDBSCAN	All Shapes	Robust	Independent	Θ(NlogN)
Fuzzy k means	Not all	Robust	Independent	Θ(N)
BOO-Clustering	All Shapes	Robust	Independent	Θ(N)

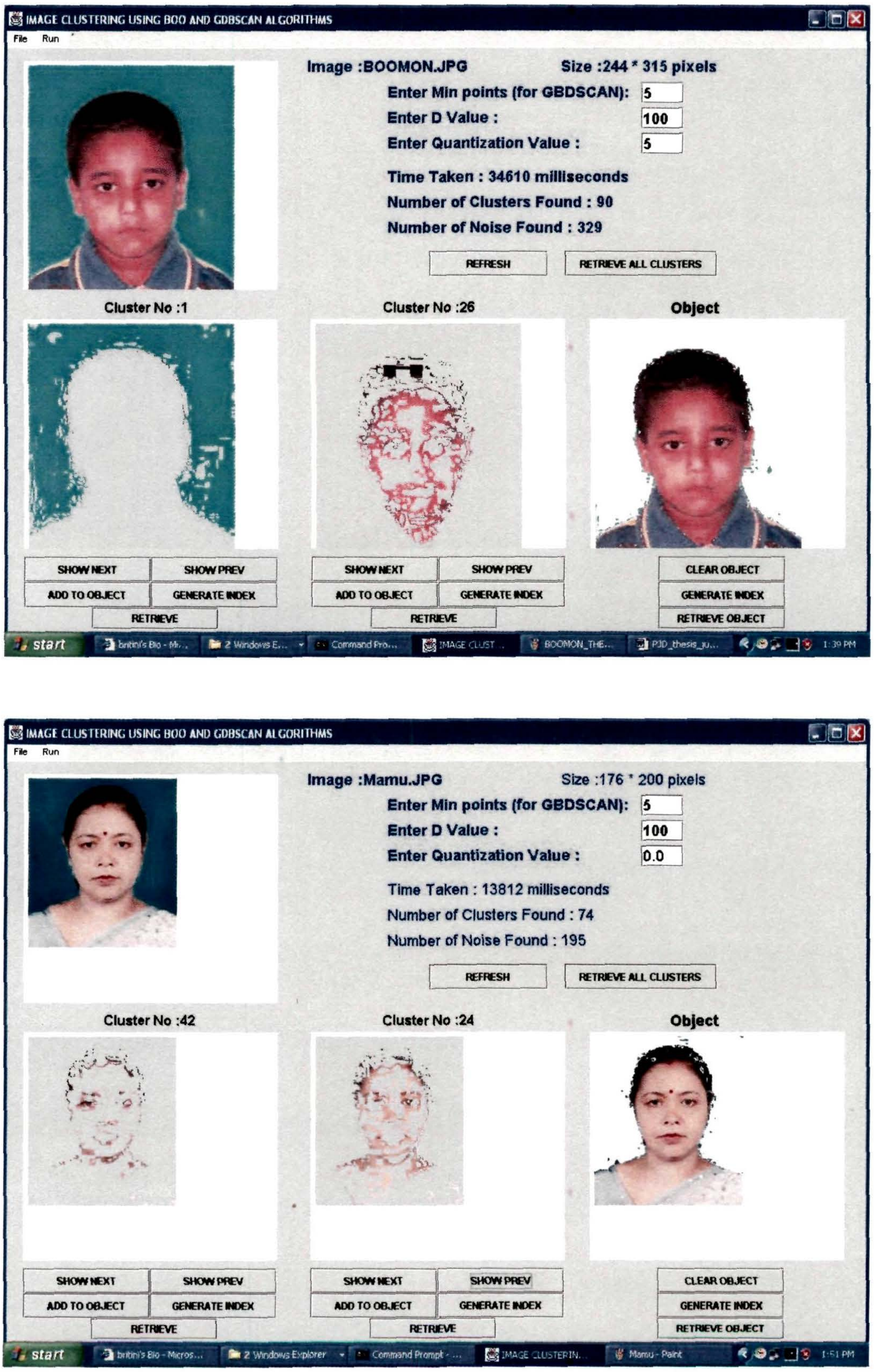


IMAGE CLUSTERING USING BOO AND GDBSCAN ALGORITHMS

Image :BOOMON.JPG Size :244 * 315 pixels

Enter Min points (for GBDSCAN): 5

Enter D Value : 100

Enter Quantization Value : 5

Time Taken : 34610 milliseconds

Number of Clusters Found : 90

Number of Noise Found : 329

Cluster No :1

Cluster No :26

Object

IMAGE CLUSTERING USING BOO AND GDBSCAN ALGORITHMS

Image :Mamu.JPG Size :176 * 200 pixels

Enter Min points (for GBDSCAN): 5

Enter D Value : 100

Enter Quantization Value : 0.0

Time Taken : 13812 milliseconds

Number of Clusters Found : 74

Number of Noise Found : 195

Cluster No :42

Cluster No :24

Object

Figure 5.2. Clusters and objects generated by BOO-Clustering algorithm for an input facial image.

5.8 Discussion

In this chapter we have discussed a new three dimensional numeric data clustering technique that have been utilized for finding the color clusters present in an image. The algorithm detects color clusters of any shape present in the image. The salient features of the clustering algorithm reported are:

- The algorithm operates in linear time and detects the color clusters present in the image in a single pass.
- The algorithm is designed based on Density/Neighborhood approach using a sliding window concept.
- Based on the clusters and objects identified by the algorithm, various features that describe an image such as shape, color etc. can be extracted from the image which may be useful in the retrieval process.

Next successive chapters are dedicated to the various schemes proposed for Content Based Image Retrieval techniques based on this clustering technique.

Chapter 6

Scheme I

This scheme presents an efficient spatial indexing technique based on Silhouette moments that makes the index robust subject to the three basic transformations for CBIR. Spatial index is generated based upon a fast and robust clustering technique, which can recognize color clusters of any shape. The new clustering technique has been found to be efficient in terms of time complexity and cluster quality than many of its counterparts. A matching engine has been devised to retrieve images from the image database, which has the capacity for global and regional similarity search.

6.1 Architecture of Scheme I

Scheme II works in four steps. In *Step 1*, it accepts the input query image, quantizes the image and produces the color clusters present in the image. In the *2nd step* the user identifies the color clusters and the combination of color clusters that forms the objects in an interactive manner. In *Step 3*, translation, aspect ratio, scale and rotation invariant silhouette moments are calculated to generate the spatial cluster indices and the spatial object indices of the selected clusters and objects. In *Step 4*, query results are given based on inferences made by a matching engine.

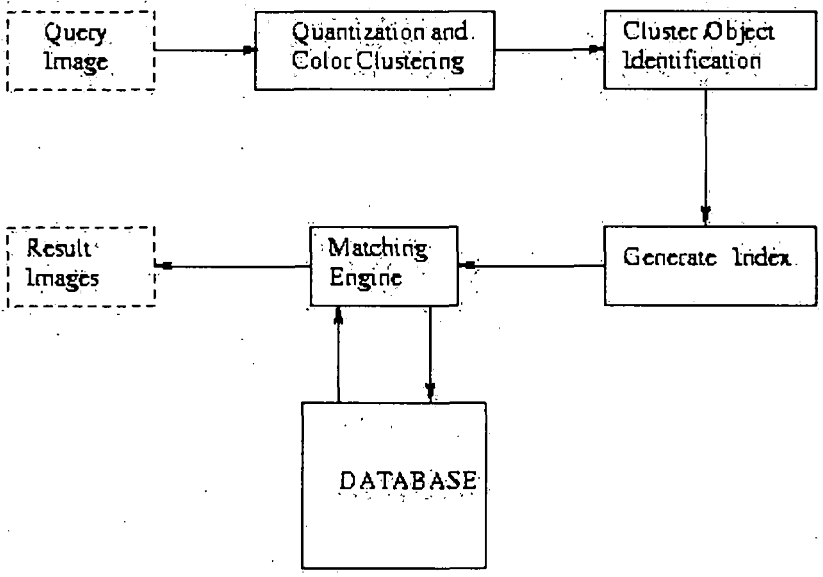
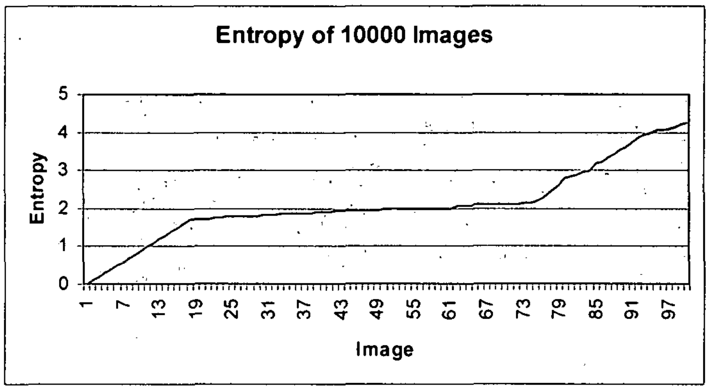

Query Image
Quantization and Color Clustering
Cluster Object Identification
Generate Index
Matching Engine
Result Images
DATABASE

Figure 7.1. Architecture of Scheme II

6.2 Entropy Calculation

As discussed in *Chapter 3* Section 3.3, entropy is calculated for the given input image and this entropy value is used for quantizing an image. An average entropy for a large collection of images (around 10,000 images downloaded from the web) has been calculated and plotted in Figure 6.2. From the entropy plot it has been found that the average entropy of images is mostly 2. Hence after exhaustive experimentation we can infer that those images whose entropy value is



less than or equal to 2 contains less colors otherwise it contains more colors and needs quantization.

6.3 Quantization

Quantization of the color space is necessary to reduce the dimensionality of the index that characterizes an image at the cost of the quality of the index. The proposed scheme quantizes the color contents of the query image over HSV color space. It basically attempts to quantize the *Hue* component of each color pixel by balancing the *visual fidelity* and the *dimensionality* of the resulting quantization. The Human Visual System (HVS) discerns the changes in the *hue* component by much smaller gaps than changes in the *Saturation* and *Value*. Based on experimentation it has been observed that partitioning the buckets with an equidistant interval of 5 is more justified. Hence the quantization parameter $q=5$ is kept the same for all images. Thus the total number of buckets over the *hue*-axis is 20. The images that have *entropy value* greater than 2 are quantized with this partition.

After quantization of those images that needs it and those images that does not need quantization have to go through the process of cluster generation and subsequently spatial object index generation module. In this scheme we have used the power of BOO-Clustering algorithm for cluster generation.

6.4 Clustering

To retrieve the color clusters present in the input image both the BOO-Clustering algorithm as discussed in *Chapter 5* and GDBSCAN [JSM00] (Generalized Density Based Spatial Clustering of Applications with Noise) has been employed. The only input parameter to BOO-Clustering algorithm is a distance parameter d that defines the neighborhood distance of the template. Where as, the GDBSCAN algorithm takes two input parameters, neighborhood distance d parameter and min-points, the minimum number of points to be present in the neighborhood defined by d parameter. The clustering results of both the algorithms depend on the choice of the input parameters.

The value of the d parameter was kept fixed for all images. But as the distribution of colored pixels vary from image to image, a fixed d value may not produce distinct and meaningful color clusters for different images by BOO-Clustering algorithm. Hence to get distinct and meaningful color clusters both the BOO-Clustering and GDBSCAN algorithms are executed in a trial and error method for different input parameters.

6.5 Detection of color clusters and objects of interest in an image

The output of the BOO-Clustering algorithm and GDBSCAN algorithm is a set of color clusters of an image. An object is formed by a single color cluster or by combination of a set of color clusters. In an image the color clusters or some or all combination of color clusters that forms object within the image may not be of importance for index generation purpose. An interactive method is the only effective way for detection of proper color clusters and objects of interest of an image for index generation. Hence in this scheme, those clusters and objects of an image are chosen which are of importance for index generation for the image manually.

For example Figure 1.1 depicts an image having four different color clusters denoted by color^1 the leaf color cluster of the tree, color^2 the trunk color cluster of the tree, color^3 the background color cluster, and color^4 the soil color cluster. Each of these color clusters can be represented as objects. The other distinct object of the image is combination of the color^1 the leaf color cluster of the tree and color^2 the trunk color cluster of the tree. Thus there are five distinct objects in the image which are of importance for index generation for the image. The other combination of color clusters such as color^3 the background color cluster and color^4 the soil color cluster are not of very much importance for index generation. Hence to select the objects of interests in an image for index generation only an interactive method is an appropriate one at the cost of time complexity.

6.6 The Spatial Cluster Index and Object Index

As discussed in *Chapter 3* the translation, aspect ratio, scale and rotation invariant silhouette moments [RMK98] of second order are given by

$$\varphi_1 = \eta_{20} + \eta_{02} \qquad (6\text{-}1)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}{}^2 \qquad (6\text{-}2)$$

$$\varphi_3 = \eta_{20}\eta_{02} - \eta_{11}{}^2 \qquad (6\text{-}3)$$

In this scheme equations (6-1), (6-2), and (6-3) are utilized for index generation of the color clusters and the objects generated. The indices for the color cluster or objects of interest consists of four parameters viz., *<color of the cluster, φ_1, φ_2, φ_3>* termed as *spatial cluster index* and *<average color of the clusters, φ_1, φ_2, φ_3>* is termed as *spatial object index* respectively. Hence, a search on spatial cluster index of all the clusters present in the input image represents a global similarity search of the image. A search on spatial object index, or one or a few spatial cluster indices of the input image represents regional similarity search of the image. The spatial cluster indices and spatial object indices are stored in a spatial Cluster Object Tree as depicted in Figure 6.2. Here the φ_1, φ_2, φ_3 are the translation, aspect ratio, scale and rotation invariant silhouette moments. For two or more similar clusters/objects with different colors will have the same φ_1, φ_2, φ_3 values.

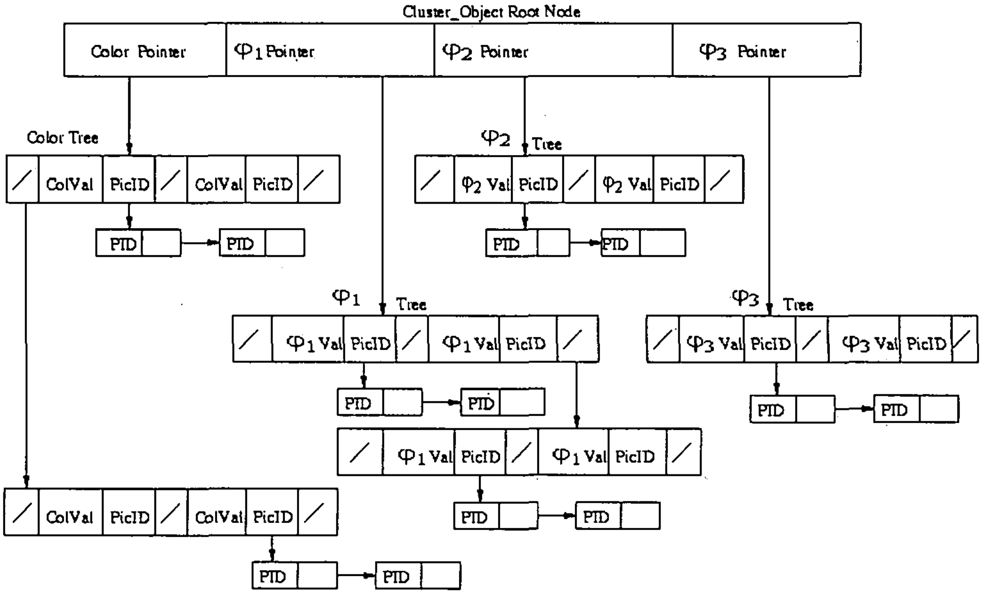


Figure 6.2. Spatial cluster-object tree

6.7 Robustness & Compactness of the Index

To establish an index to be robust, it has to be robust subject to the three basic transformations, i.e. translation, rotation, uniform and non-uniform scaling. The clusters produced by BOO-Clustering are more reasonable than GDBSCAN [JSM00] due to the following reasons:

1. The time complexity of BOO-Clustering is $\theta(N)$ as compared to $\theta(N \times \log N)$ of GDBSCAN when applied to image data.
2. Both GDBSCAN and BOO-Clustering algorithms can identify color clusters of arbitrary shapes present in the image viz. concave and convex. But the drawback of GDBSCAN is that it depends on two parameters viz. d and min-points; where as BOO-Clustering depends only on one parameter viz. d. Hence keeping d parameter to a fixed value we shall always get a fixed set of color clusters for a particular image in case of BOO-Clustering algorithm. But in case of GDBSCAN, keeping d parameter fixed but by altering the second parameter min-point we shall get different set of color clusters.

The three parameters viz., φ_1, φ_2, φ_3 are already been proved to be robust to the three basic transformations [RMK98]. The other parameter average color of the cluster or the object is also robust to the three basic transformations because the average color of the cluster or object of an image always remains the same even the image is tempered with the three basic transformations. Moreover if the color of the image is tempered, the other three components of the index will not change. Thus the four dimensional index is robust in all respect.

As regards compactness of the index, both BOO-Clustering and GDBSCAN algorithm produces color clusters present in the image eliminating the noisy clusters. From these color clusters objects of interest are extracted. For each color clusters and object of interest a four-tuple index is calculated. Hence the size of the index varies from image to image depending on the color clusters and objects of interest present in the image. But index generated by the George *et al.* [GPI03] method is more compact than ours because their method generates only one set of twelve chromatic moments for each image and which is fixed for any image. Keeping in view of the retrieval efficiency, ours is better than our counterparts. This is because our method can perform global search as well as region or object level search, which is a lacuna of George *et al.* method that it cannot search at *object level.*

6.8 Database Organization

The output of the BOO-Clustering algorithm and GDBSCAN is a set of color clusters from which the objects of interest are separated out from the image in an interactive manner. The spatial cluster indices and object indices are stored in a spatial cluster-object tree. The spatial cluster object tree is a variant of B-tree (Figure 6.2). The root node termed as Cluster_Object Root Node, of the tree points to k independent tree structures, where k is the number of parameters in the index. If a new parameter is to be accommodated (i.e., for a k+1 dimensional index), the root node has to be updated by insertion of a new pointer and accordingly an associated tree structure will have to be generated. Each of the parameter trees will maintain the parameter key value (i.e., say, *Cluster/Object Color* value, φ_1 value, φ_2 value, φ_3 value), along with a pointer to a list of Image IDs (i.e., PIDs).

6.9 Matching Engine for Scheme I

The matching engine has the facility for searching images in the image database based on the spatial cluster and object indices using the cluster-object tree (Figure 6.2). Figure 1.1 depicts a query image with four clusters having four different colors; color^1, color^2, color^3, and color^4. For each cluster, the query image will have one cluster index. Let these cluster indices be denoted by <color^1, φ_1^1, φ_2^1, φ_3^1>, <color^2, φ_1^2, φ_2^2, φ_3^2>, <color^3, φ_1^3, φ_2^3, φ_3^3> and <color^4, φ_1^4, φ_2^4, φ_3^4>. For global search, the matching engine searches for all those images that have all the four cluster indices, as that of the query image, present in image database.

For object level search, at first the objects present in the query image are determined from the set of clusters present in the query image. The objects may consist of either a single color cluster or a combination of color clusters. For example, in Figure 1.1 the query image has four color clusters (1, 2, 3 and 4). Cluster 1 and 2 forms the object "tree"; color cluster 3 is the "background" object; color cluster 4 is the "soil" object. Also color cluster 1 is the "leaf" object of the tree and color cluster 2 is the "trunk" object of the tree. Thus there are all total five objects in the query image as depicted in Figure 1.1 and hence there will be five object indices in the query image. Let these object indices be denoted by <Leaf_object^1, φ_1^1, φ_2^1, φ_3^1>, <Trunk_ object^2, φ_1^2, φ_2^2, φ_3^2>, <Background_ object^3, φ_1^3, φ_2^3, φ_3^3>, <Soil_ object^4, φ_1^4, φ_2^4, φ_3^4> and <Tree_ object^5, φ_1^5, φ_2^5, φ_3^5>. Here Tree_ object^5 value is the average color value of Leaf_object^1 and Trunk_object^2.

For object level search, one of the query may be "*Find all the images from the image database that have the object 'tree'*". For this query the matching engine will use the index <Tree_ object^5, φ_1^5, φ_2^5, φ_3^5>. At first it will search in the color tree and will fetch those Picture Ids from the database that matches the stored color value with the Tree_ object^5 color value. Next it will take up φ_1^5 value and search will be done on the φ_1 tree and will fetch another Picture Id list for matching stored φ_1 value with φ_1^5 value. Similarly the matching engine will fetch another two Picture Id lists for φ_2^5 and φ_3^5 value by searching in the φ_2 and φ_3 tree respectively. Next all the four Picture Id lists are concatenated to form a single Picture Id list, which is the final list of image retrieved.

The matching engine also has the facility for searching in the image database for color clusters and objects using both the conjunction 'AND' and disjunction 'OR'. For example, the query image as depicted in Figure 1.1 may have a query that "*Find all images that have color cluster (1 AND 2) OR 3; i.e. find all those images which have either both the (Leaf Objects 1 AND Trunk Object 2) OR Background Object 3*". For this query, the indices < Leaf_object^1, φ_1^1, φ_2^1, φ_3^1>, <Trunk_ object^2, φ_1^2, φ_2^2, φ_3^2> and < Background_ object^3, φ_1^3, φ_2^3, φ_3^3> of the query image will be utilized and the search will initiate from Leaf object index. At first, it will search in the color tree and will fetch those Picture Ids from the database that matches Leaf_object^1 color value. Next it will take up φ_1^1 value and search will be done in the φ_1 tree and will fetch another Picture Id list that matches φ_1^1 value. Similarly, the matching engine will fetch another two Picture Id lists for φ_2^1 and φ_3^1 value by searching in the φ_2 and φ_3 tree respectively. After that all the four Picture Id lists will be concatenated to form a single Picture Id List (PID^1) which matches the first Leaf Object index < Leaf_object^1, φ_1^1, φ_2^1, φ_3^1> of the query. Thus another two Picture Id lists (PID^2 and PID^3) for the indices <Trunk_ object^2, φ_1^2, φ_2^2, φ_3^2> and < Background_ object^3, φ_1^3, φ_2^3, φ_3^3> will be generated. Now, according to the query the Picture Id lists PID^1, PID^2 and PID^3 will be combined together according to (PID^1 AND PID^2) OR PID^3 to produce the final set of Picture Id list that corresponds to the images in the image database which satisfy the given query. This type of query can also be applied to object level search and combination of object and cluster level search.

The matching engine also has the facility of searching images in the database based on the parameters of interest and also on objects/clusters of interest. Both the conjunction AND and disjunction OR can be used both on objects and parameters. For example a query can be "*Search for all images having any (color value) FOR (φ_1 value AND φ_2 value AND same φ_3 value) for objects (Leaf Object AND Trunk Object) OR Background Object. i.e. Search for images that contains the Tree Object and Background Object of any color*". For this query, the indices < Leaf_object^1, φ_1^1, φ_2^1, φ_3^1>, <Trunk_ object^2, φ_1^2, φ_2^2, φ_3^2> and < Background_ object^3, φ_1^3, φ_2^3, φ_3^3> of the query image will be utilized and the search will initiate from Leaf object index. The matching engine will start the search operation from Leaf object. First it will fetch a list of images by searching in the φ_1 tree. Similarly it will fetch two lists of images for φ_2, and φ_3 and finally these three lists of images will be merged to a single list (PID1) according to the given criteria φ_1 AND φ_2 AND φ_3. This list (PID1) will contain all images that have the Tree object of any color and having similar shape. Similarly, another two lists (PID2) and (PID3) will be generated by the matching engine for the Trunk object and the Background object. Finally these three lists will be concatenated according to the given, criteria; Leaf object AND Trunk object OR Background object; to form a single list. This final list of images is the desired set of images of the said query. The Matching Algorithm has already been discussed in *Chapter 5*.

6.10 Complexity Analysis and Comparison

There are two phases of computations involved in querying an image database. First, calculation of the index for the query image and second, comparison of the generated index with the stored indices of the images in the image database and subsequently retrieval of the similar images from the image database. We have compared this Scheme with the Paschos et. al. Chromatic Moment [GPI03] based technique.

6.10.1 Retrieval Time by Chromatic Moments:

1. Computation of Chromatic Moments is $O(M \times Q^2 \times (2 \times \mu \times \alpha))$
   where M = No. of Chromatic Moments = 12,
   Q = No of histogram bins,
   μ, α = 2 multiplication and one addition per histogram bin.
2. Histogram computation time is O(N)
   where N = Total numbers of pixel in the image
3. Time taken by reading the chromaticity moments of each stored image is $O(S \times M \times \rho)$
   Where S = Total number of images in the image base,
   ρ = Time for reading the number from the disk
4. Computing distance to each stored image is $O(S \times M \times (\alpha + d))$
   Where d = is the time taken to compute an absolute value

Hence the total retrieval time is O((1) + (2) + (3) + (4))

Which is $O(M \times Q^2 \times (2 \times \mu \times \alpha)) + O(N) + O(S \times M \times \rho) + O(S \times M \times (\alpha + d))$

Which is $O(M \times Q^2 + N + S \times M + S \times M)$

Which is $(M \times (Q^2 + 2 \times S) + N)$

Which is $(Q^2 + 2S + N)$

6.10.2 Retrieval Time by BOO-Clustering and GDBSCAN techniques:

1. Time taken for Cluster generation by BOO-Clustering technique is O(N)
2. Time taken for Cluster generation by GDBSCAN technique is O(N × log N)
3. Time taken for Index calculation for an object having C_i no of pixels =
   Time taken for calculation of central moment for μ_{00}, μ_{20}, μ_{02}, μ_{11} + Time taken for calculation of aspect ratio invariant moment for $\eta_{20}, \eta_{02}, \eta_{11}$ + Time taken for calculation of rotation, translation, scale and aspect ratio invariant moments φ_1, φ_2, φ_3

is $O(4 \times C_i + \psi + \nu)$

4. Time taken for calculation for O number of objects

   is $O(4 \times (\sum_{i=1}^{O} C_i) + \psi + \nu)$

   which is $O(4 \times\psi + \nu)$ is O(4N)

5. Retrieval time is O × log S is log S [O is ignored because of its smaller size]
6. Total retrieval time taken by BOO-Clustering technique = (1) + (4) + (5)

   is N + 4N + log S

   is 5N + log S
7. Total retrieval time taken by GDBSCAN technique = (2) + (4) + (5)

   is O(N×log N + 4N + log S) is O(N log N + log S)

Here, $Q^2 < 5N$ and $Q^2 < N\times\log N+4N$ but $2S >> \log S$. If the number of images in the image database is smaller in size the Chromatic Moments index has time advantage over the BOO-Clustering and GDBSCAN techniques at the cost of precision and recall. As the number of images in the image database increases in size, both the BOO-Clustering and GDBSCAN techniques have time advantage over chromatic moments with a good precision and recall.

6.11 Experimental Results

To test the technique, we used a downloaded database ((a) Cohn-Kanade Facial Expression Database; http://www.pitt.edu/~jeffcohon, (b) ftp://ftp.eecs.umich.edu.groups /ai/dberwick/essbthm.zip and other images) consisting of 5000 real world and synthetic images divided into 100 similar groups such as *facial expressions*, *scenery*, *animals*, *cars*, *flowers* and *space*. Implementation was carried out for chromatic moment based technique along with the proposed cluster and object based techniques using BOO-Clustering method and GDBSCAN method in HSB color space. All the methods have been implemented in Java platform in an Intel Pentium IV machine. For any input image, indices are generated for the color clusters and objects of interest in an interactive manner for both the BOO-Clustering and GDBSCAN method. Similarity search is done on clusters of interest or objects of interest using a matching engine. We have performed a two way test to establish the supremacy of the BOO-Clustering method over the Chromatic Moment and GDBSCAN methods. They are:

1. Retrieval time comparison of BOO-Clustering method vs GDBSCAN method and Chromatic Moment method.

2. Retrieval effectiveness of the three methods for global and regional search.

6.11.1 Retrieval time comparison of BOO-Clustering vs GDBSCAN and Chromatic Moment methods

In this experiment we have tried to plot the average time taken by the three methods vs the number of images in the image database for global and regional search. In global search, all the clusters generated by the GDBSCAN and BOO-Clustering method are taken into account for the search operation; and in regional search, some significant clusters or objects produced by GDBSCAN and BOO-Clustering method are taken into account for the search operation. To test the retrieval time comparison, we have taken 100 images from the image database and the average retrieval time for the 100 images for both the global and regional search has been plotted for BOO-Clustering, GDBSCAN, Chromatic Moment method for different database size (Figure 7.3 and 7.4). It can be clearly seen that as the size of the image database increases the BOO-Clustering method has time advantage over the GDBSCAN and Chromatic Moment method.

6.11.2 Retrieval effectiveness of the three methods for Global and Regional Search

Retrieval effectiveness is calculated by the *average precision recall* (APR) curve. In our experiment, the APR was plotted for the best 200 queries out of the calculated 500 queries taken at random over the downloaded image database. . Numerically, the *area* [RKS99] enclosed by an APR curve and the axes as a performance metric, called *performance area*, is defined as

$$\frac{1}{2}\sum_{i=1}^{N-1}(x_{i+1} - x_i)(y_{i+1} + y_i) \qquad (5\text{-}5)$$

where (x_i,y_i) is the *(recall, precision)* pair when the number of retrieved images is i and N is the total number of top matches.

The proposed method was compared with the Chromatic Moment based technique [RKS99] along with the GDBSCAN [JSM00] method for both global and regional search. In global search, all the clusters produced by BOO-Clustering and GDBSCAN for a query image are taken into account for the search operation for both the techniques. Figure 6.5 reflects the query result of global search of the three techniques. Based on experimental results, it is found that the performance areas for the curves are: 387 (BOO-Clustering based retrieval), 321 (GDBSCAN based retrieval) and 271 (Chromatic Moment based retrieval). Hence the improvement produced by the proposed BOO-Clustering based method over the GDBSCAN and Chromatic Moment based technique are: 20.56% and 42.8% respectively.

The proposed technique was also tested based on some specific clusters and objects (formed by combination of some selected clusters) of the query image along with its counterparts. For fine-tuning the search operation we have used AND/OR conjunction between the clusters and objects of some of the query images, i.e. if for a query image, five significant clusters (C1, C2, C3, C4, C5) are chosen for the search operation, then the search can be given as: "Search for all the images in the image database where cluster C1 AND C2 AND C3 OR C4 AND C5 matches with the query image". Figure 7.6 reflects the query result of regional search of the two techniques. Based on experimental results, it is found that the performance areas for the curves are: 309 (BOO-Clustering based retrieval) and 258 (GDBSCAN based retrieval). Hence the improvement produced by the proposed BOO-Clustering based method over the GDBSCAN based technique for regional search is 19.77%.

6.12 Discussion

An improved content-based indexing scheme has been presented in this scheme. The scheme generates a compact, 4-dimensional transformation invariant index for the color clusters and objects of interest produced by a robust data clustering technique of any color image. The indices produced help in global and regional similarity search of images. The indices generated are stored in a spatial tree structure for faster retrieval of images. The proposed scheme can be found to be superior in comparison to its counterparts [GPI03]. The main drawback of the scheme is that, the clusters generated by BOO-Clustering and GDBSCAN algorithms are to be selected in an interactive manner for object detection and subsequently index generation for all images while creating the spatial cluster and object index database. Hence, the scheme suffers from execution time point of view while creating the spatial cluster index and object index database. To overcome it, next chapter presents an enhanced version of the scheme.

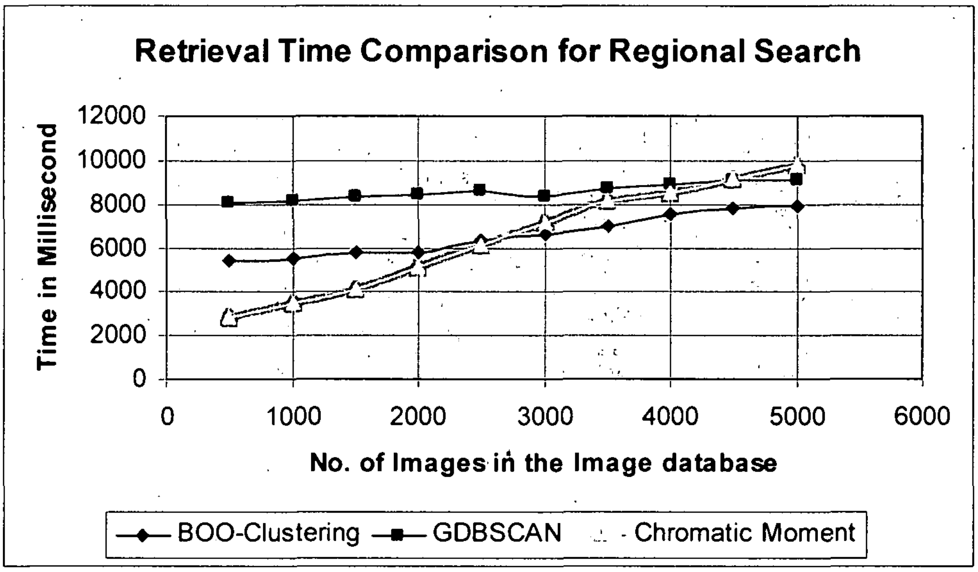


Figure 6.3. Retrieval time comparison for Regional Search.

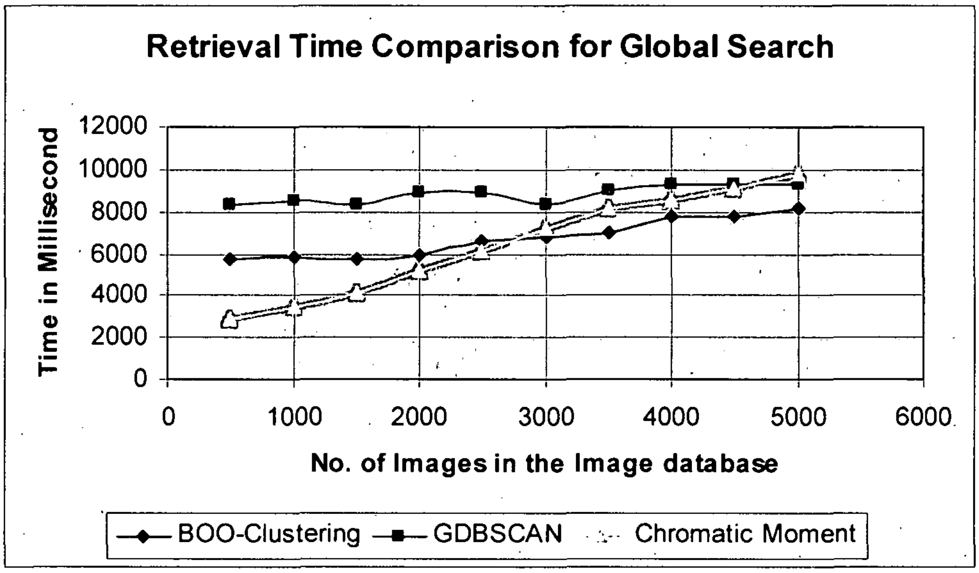


Figure 6.4. Retrieval time comparison for Global Search.

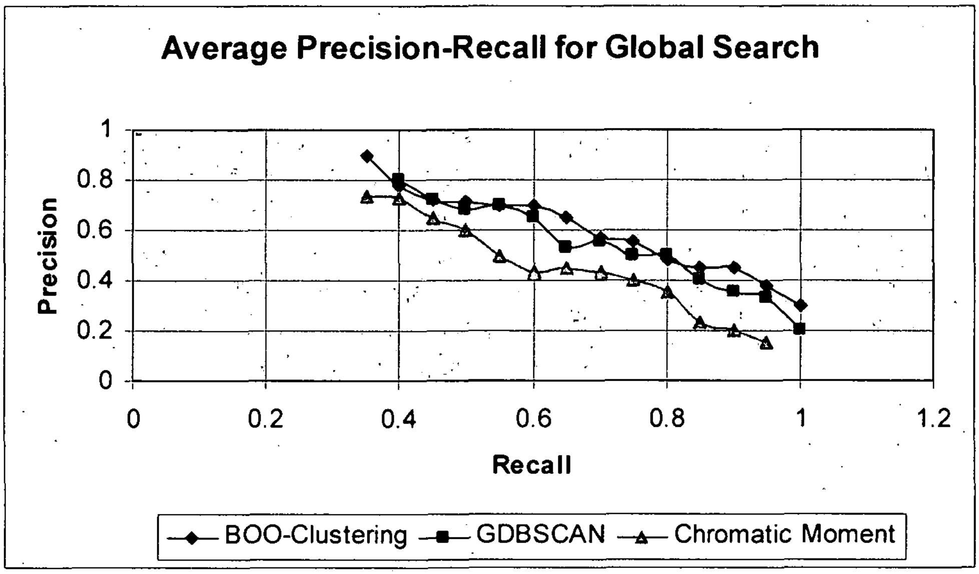


Figure 6.5. Precision Recall for BOO-Clustering vs GDBSCAN and Chromatic Moment based techniques for Global search.

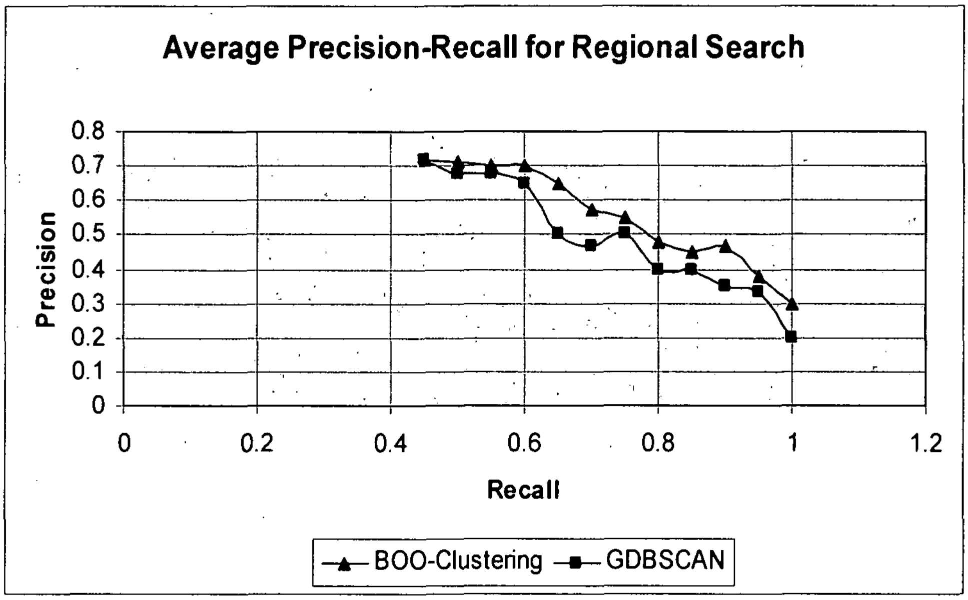
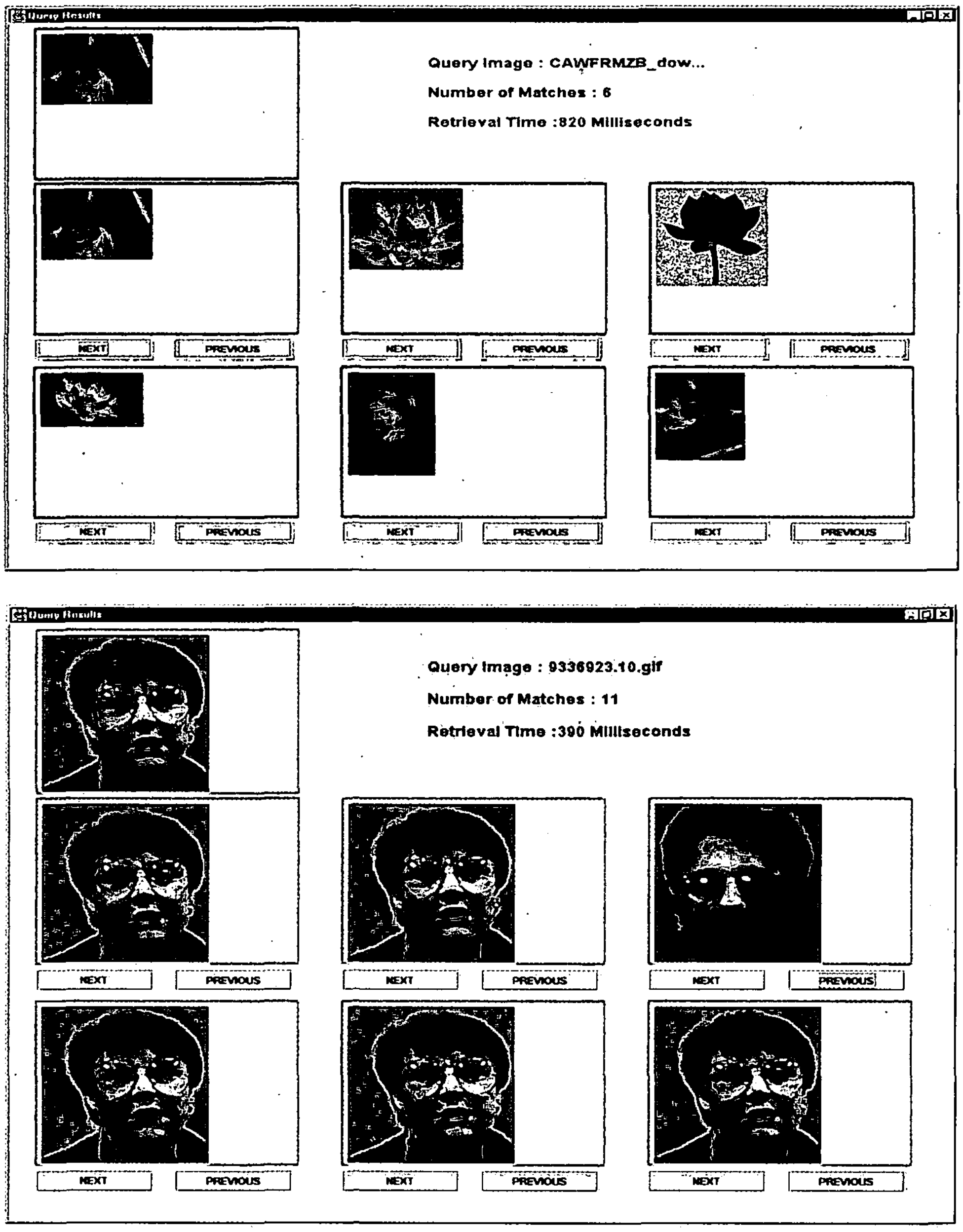


Figure 6.6. Precision Recall for BOO-Clustering vs GDBSCAN based techniques for Regional search.



Query Image : CAWFRMZB_dow...

Number of Matches : 6

Retrieval Time :820 Milliseconds

Query Image : 9336923.10.gif

Number of Matches : 11

Retrieval Time :390 Milliseconds

Figure 6.7. Some retrieval results of Scheme I.

Chapter 7

Scheme II

This scheme presents a better spatial indexing technique than Scheme I based on Silhouette moments that makes the index robust subject to the three basic transformations for Content Based Image Retrieval.

This scheme is an enhancement of those schemes reported in the previous two chapters. Though this scheme also exploits the silhouette moments, however, it is based on color quantization, color cluster generation, followed by objects of interest generation. All these processes involved are fully automated i.e. no input parameter is necessary to achieve the goal of index generation. Thus for a particular image the spatial index generated always remains the same every time it is processed. All the indices generated for a particular image may not be of importance for retrieval purpose. Hence those indices are kept in the database which are important. This is done by retaining only the interesting ones based on the percentage frequency of occurrence dynamically. The indices are stored in two tree structures termed as database_frequent and database_all which facilitates in speeding up of the retrieval performance. A modified matching engine has been devised to retrieve images from the image database which have the capacity for global and regional or object level similarity search. Based on experimentation, the performance of this Scheme has been found to be better than Scheme I. Next, we describe the architecture of the scheme.

7.1 Architecture of Scheme II

Scheme II works in *six* steps. In *Step 1* it accepts the input query image and quantizes the image over the hue component in HSV color space. In *Step 2*, it generates all the color clusters present in the image using modified BOO-Clustering algorithm. In *Step 3,* detection of useful color clusters and objects of interest in an image are done. In *Step 4,* indices are generated for all combinations of the color clusters and objects of interest present in the image. In *Step 5*, the query results are given based on inferences made by a matching engine. In *Step 6*, all those indices of objects are stored in the database, which are interesting based on the percentage frequency of occurrence dynamically (Learning Objects of Interest). The architecture of the proposed scheme is depicted in Figure 7.1. Next, we describe each of the modules presented in the architecture in brief.

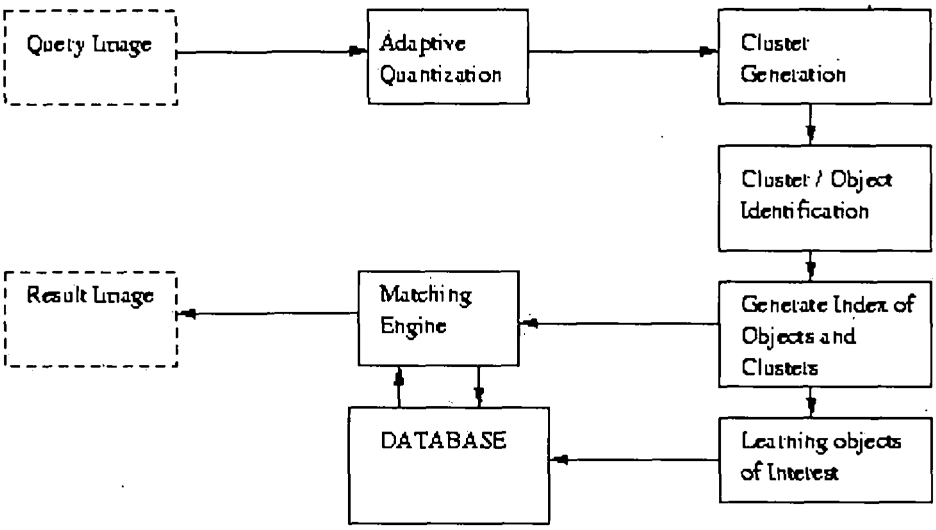

Query Image
Adaptive Quantization
Cluster Generation
Cluster / Object Identification
Result Image
Matching Engine
Generate Index of Objects and Clusters
DATABASE
Learning objects of Interest

Figure 7.1: Architecture of Scheme II.

7.2 Adaptive Quantization

As reported in *Chapter 3* the quantization of a color image is performed for two perspective: 1) to reduce the size of the image and thereby to reduce the color content, 2) to reduce the color content keeping the size of the image intact. In both the cases a static quantization (non-realistic) is resulted if 1) a fixed window size say 3×3 or 5×5 pixels is considered and 2) a fixed bin size or equidistant bin size is used for all images.

However, in reality it is not so simple because the distribution of colors within an image varies from image to image. The color of the target pixel in an image depends on the color of the pixels surrounding the target pixel. Hence, in such a situation quantization by a fixed bin size or a fixed window size may lead to detoriation of the quality of the quantized image. In such a situation, a variable window size or bin size is necessary for quantizing an image whose size may differ from image to image. This type of quantization where variable sized window or bin is used may be termed *adaptive quantization* leading to less detoriation of the quantized image.

In our experiment, we estimate the size of the window for each image based on the distribution of color within the image. To achieve this goal, we use entropy for estimation of the window size. Entropy is a measure of the variation in the color content of the image [JMZ00]. It is calculated for a colored image as: $H(v) = -\sum v_i \log(v_i)$, where v_i is the percentage of pixels in the image which belongs to color i. $H(v) = 0$ implies that all the pixels of the image are of same color. $H(v)$ is maximized when all possible colors in the color space of the image are equally represented.

The number of pixels surrounding any target pixel is 8, 16, 24, ... depending on the neighborhood distance d_n = 1, 2, 3, ... respectively which defines the window size. We have proposed an algorithm (discussed next), which calculates this neighborhood distance d_n automatically. To achieve this, our algorithm calculates the local entropy e_w of the pixels present in the window defined by d_n and compares it with the average entropy E of the whole image. If the local entropy is less than the average entropy, then the neighborhood distance d_n is used for the quantization of the whole image. Conceptually, it calculates the minimum size of the window where the density of similar colored pixels is more than that of the whole image, i.e., entropy e_w is less than E. Thus, the window size may vary from image to image leading to quantization of a target pixel based on the distribution of pixels in the image and distribution of the pixels in the window.

To find the smallest window size, defined by d_n, to be used for quantization, the algorithm scans from the 1st pixel till it reaches the last pixel. The algorithm first calculates the average entropy E of the whole image. d_n, which defines the smallest size of the window, is assigned with m or n whichever is bigger. A local window size defined by d_w is assigned with a initial value (m-1) or (n-1) whichever is bigger. A local entropy e_w is calculated for the pixels within a window defined by d_w of a target pixel. If e_w<=E and d_w< d_n, then d_n is assigned with d_w; the algorithm goes to the next pixel and the process is repeated by assigning d_w=(m-1) or (n-1). Else if d_w>= d_n, the algorithm goes to the next pixel and process is repeated by assigning d_n=(m-1) or (n-1). If e_w>E and d_w< d_n, then the value of d_w is decremented by 1 and the process is repeated with the decremented value of d_w . Else if d_w >= d_n, then the algorithm goes to the next pixel and the process is repeated with d_w=(m-1) or (n-1) whichever is bigger.

Thus, the algorithm calculates the smallest window size defined by the distance parameter d_n for which entropy value is more than the entropy of the whole image, i.e., it finds the smallest window size in which color variation of the similar colored pixels present in the window is more than the color variation of the whole image. This window size is used for the quantization process.

In the next phase, the process of quantization starts from the 1st pixel till it reaches the last pixel using the window defined by the quantization parameter d_n calculated in the 1st phase of the algorithm. For quantization of a target pixel, the algorithm calculates the local entropy e_l of the pixels in the window defined by d_n and compares it with the average entropy value E of the image. If $e_l<E$, quantization of the target pixel is not necessary, otherwise, the average color of the pixels present in window is assigned to the target pixel. Thus, a better quantization of the images has been achieved without much distortion at the cost of time complexity θ(N×m) or θ(N×n).

7.2.1 Algorithm for estimation of neighborhood distance d_n for adaptive quantization

```
Step 1: Calculate the average entropy $E$ for the whole image.
Step 2: $d_n = m$ or $n$ which ever is bigger.
Step 3: For each pixel in the image
Step 4: $d_w=(m-1)$ or $(n-1)$ which ever is bigger.
        Step 4.1: Calculate local entropy $e_w$ of the local window defined by $d_w$.
        Step 4.2: If ($e_w <= E$) then
                        If ($d_w < d_n$) then
                              $d_n = d_w$
                              go to step 3 for the next pixel.
                        Else if ($d_w >= d_n$)
                              go to step 3 for the next pixel.
                        Endif
                  Else if ($e_w > E$) then
                        If  ($d_w < d_n$) then
                              $d_w = d_w - 1$.
                              go to step 4.1.
                        else if ($d_w >= d_n$)
                              go to step 3 for the next pixel.
                        Endif
                  Endif
```

7.2.2 Algorithm for Adaptive Quantization

```
Step 1: For each pixel in the image
        Step 1.1: Calculate the local entropy $e_l$ defined by the distance parameter $d_n$.
        Step 1.2: If ($e_l$ >= $E$ ) then
                        Assign the average color value of the pixels in the local window defined
                        by $d_n$ to the target pixel.
                        Go to step 1 for the next pixel.
                  Else
                        Go to step 1 for the next pixel.
                  Endif
```

7.3 Cluster Generation

Once the image has been quantized, it has to pass through the clustering phase in the next step. In this scheme we have used the BOO-Clustering algorithm that has been discussed in *Chapter 5.* The only input to the algorithm is the neighborhood d parameter that defines the size of the template (sliding window). We have used the d value as user input, but this has a drawback. Choice of the d parameter may vary from user to user for the same image leading to different set of cluster generation and ultimately a different set of indices.

Hence in this scheme a new method has been devised for automatic calculation of d parameter and the value of this d parameter remains the same every time the image is processed for clustering using the BOO-Clustering algorithm and we term it as Modified BOO-Clustering Algorithm.

7.3.1 Modified BOO-Clustering Algorithm: A Heuristic Approach

To calculate the distance parameter d that determines the template and to find color clusters automatically, the modified BOO-Clustering algorithm is executed repeatedly taking d value from 1 to *m* or *n* (size of the image) whichever is greater. While clustering with a smaller d value, the number of clusters produced is more than the number of clusters produced by taking a bigger d value because the template represented by a smaller d value is smaller than the template represented by a bigger d value. Thus, a smaller template detects only the thickly populated clusters and a bigger template detects sparsely distributed clusters, i.e., bigger clusters which will encapsulate smaller clusters. From a certain d value up, the number of clusters produced by the modified BOO-Clustering algorithm remains the same. Hence, the process stops when it finds the number of clusters produced by two successive d values are equal. Thus this algorithm calculates the d value automatically and the d value remains the same irrespective of number of times the image is processed for a particular image. The clusters produced by the modified BOO-Clustering algorithm for the final d value are used in the next step for cluster and objects of interest selection. Thus a better clustering result is achieved which remains the same irrespective of user input d value at the cost of higher complexity θ(N×m) or θ(N×n) of the modified BOO-Clustering algorithm.

Data Structure/Symbols Used for Modified BOO-Clustering Algorithm

Image(m, n) : Array to hold spatial color data of an input color image where *m* is the number of rows & *n* is the number of columns in the array.

d : Number that defines the size of the *Template*.

total_cluster_A: Variable that holds the total clusters detected.

total_cluster_B: Variable that holds the total clusters detected for incremented value of *d*.

mn : Variable to hold the size of row or column which ever is bigger.

Modified BOO-Clustering Algorithm(*Image*(*m*,*n*))

```
begin
        if (m>n) then
                mn = m;
        else
                mn = n;
        endif;
        for d = 1 to mn
        begin
                total_cluster_A = BOO-Clustering(Image(m,n), d);
                total_cluster_B = BOO-Clustering(Image(m,n), d+1);
                if (total_cluster_A = total_cluster_B) then
                        return(total_cluster_A, d);
                        stop;
                endif;
        end;
end;
```

7.4 Detection of Useful Color Clusters and Objects of Interest in an Image

In Scheme I clusters and objects are to be selected in an interactive method for index generation. This may sometimes lead to inappropriate index generation as because the choice of color clusters to object formation may differ from user to user. One method is to take all possible combination of color clusters forming the objects. But this also has problem of exponential growth of the number of objects.

To alleviate the problems we use a procedure of reduction in the number of the different combinations used for the objects. This is done by retaining only the interesting ones based on the percentage frequency of occurrence dynamically. This is an enhancement to the Scheme I and II.

This is performed by taking a *compound object index* for indexing purpose. The compound object index is represented by a single object index for the whole object and a set of color cluster indices that forms the object. This is because an object is formed by a set of color clusters. For example, in Figure 1.1, the tree object is constituted of the Leaf color cluster and the Trunk color cluster. Hence the compound object index of this tree object is represented by the object index <φ^T_1, φ^T_2, φ^T_3> which carries the spatial information of both the Leaf color cluster and Trunk color cluster taken together. The other part of the index is the set of color cluster indices and are defined by <*color of the trunk,* φ^t_1, φ^t_2, φ^t_3> and <*color of the leaf,* φ^l_1, φ^l_2, φ^l_3>. The parameters <φ^t_1, φ^t_2, φ^t_3> and <φ^l_1, φ^l_2, φ^l_3> defines the spatial information of the Trunk color cluster and Leaf color cluster respectively.

Thus in general, the compound object index is represented by a single object index <φ^o_1, φ^o_2, φ^o_3> which carries the spatial information of the whole object and a set of color cluster indices <*color of the cluster,* φ^c_1, φ^c_2, φ^c_3> associated with the object. Here translation, rotation, scale invariant silhouette moments of second order (equations 6-1, 6-2, 6-3) are used for index generation. Hence in this scheme, an object having one cluster, the compound object index will have one object index and a single color cluster index associated with it. Similarly, an object having three color clusters the compound index will have one object index and three color cluster indices. For two or more compound indices having equal object indices values for the parameters <φ^o_1, φ^o_2, φ^o_3> but different set of color clusters indices, implies that the object carry the same spatial information but they are formed by different set of color clusters or different colored clusters or the clusters are of different shapes and orientation. The object index <φ^o_1, φ^o_2, φ^o_3> carries the spatial information of the object but it does not carry the color information of the object as the object may be formed by one or many color clusters. But the color cluster indices <*color of the cluster,* φ^c_1, φ^c_2, φ^c_3> carries the spatial as well as the color information of each color cluster that it represents.

7.4.1 Learning Objects of Interest

In this step to learn the objects of interest discarding the others, two database structures: *database_frequent* and *database_all* which are a variant of B-tree have been used. The structure of both the database is similar. *database_all* contains the indices of all the objects for all the images in the image database. *database_frequent* contains the indices of all those objects of the images which occur in *database_all* by some percentage which is defined by some threshold value. The structure of both the database is depicted in Figure 8.2.

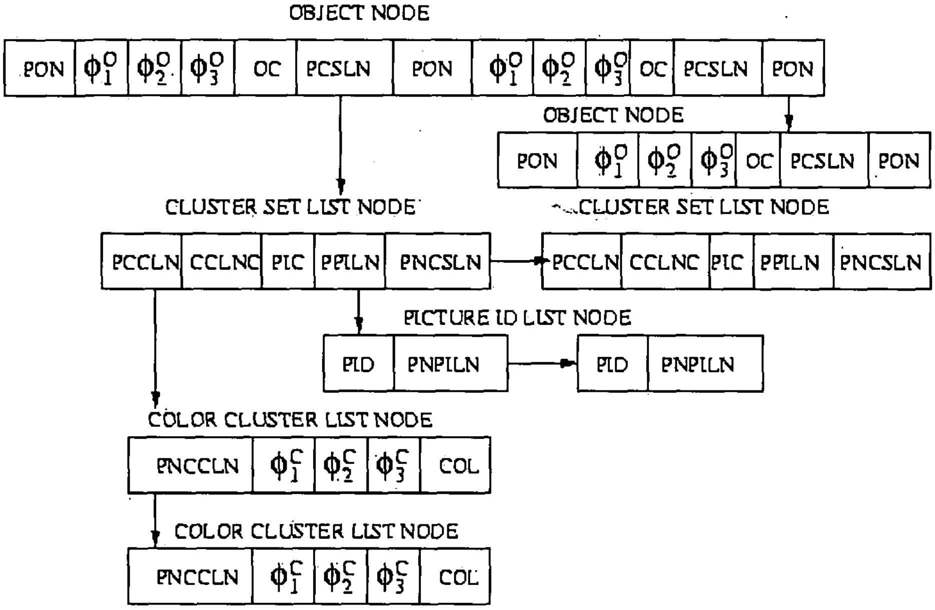


Figure 7.2: Structure of database_frequent and database_all

Both the structures consist of the following Nodes and Parameters:

- Object Node(ON): Holds information regarding an object.
- Cluster Set List Node(CSLN):Holds information regarding the set of color clusters that forms an object.
- Color Cluster List Node(CCLN):Holds information regarding color clusters.
- Picture Id List Node(PILN): Holds the unique file name regarding an image the object belongs to.

Object Node consists of the following parameters:

- Pointer to Object Node (PON).
- φ^{o}_{1} value.
- φ^{o}_{2} value.

- φ^o_3 value.
- Object Count (OC) represents the total number of similar objects the Object Node holds.
- Pointer to Cluster Set List Node (PCSLN).

Cluster Set List Node consists of the following parameters:

- Pointer to Color Cluster List Node (PCCLN).
- Color Cluster List Node Count (CCLNC) represents the total number of Color Cluster List Nodes are there associated with the Cluster Set List Node.
- Picture Id Count (PIC) represents the total number of Picture Ids attached to the Cluster Set List Node.
- Pointer to Picture Id List Node (PPILN).
- Pointer to Next Cluster Set List Node (PNCSLN).

Color Cluster List Node consists of the following parameters:

- Pointer to Next Color Cluster List Node (PNCCLN).
- φ^c_1 value.
- φ^c_2 value.
- φ^c_3 value.
- Color of the Cluster (COL).

Picture Id List Node consists of the following parameters:

- Picture Id, the file name of the image (PID).
- Pointer to Next Picture Id List Node (PNPILN).

To learn objects of interest of a given input image first all combinations of color clusters that form objects are determined. Next compound indices are generated for each identified object. These compound indices are inserted into the database_all database structure and the appropriate Picture Id Count (PIC), Color Cluster List Node Count (CCLNC) and Object Count (OC) parameters are updated accordingly. Next for each object, search is performed first in database_frequent. If the search is positive then the search terminates and the Picture Id Count (PIC) parameter of the appropriate Cluster Set List Node (CSLN) is incremented by 1 and the Object Count (OC) parameter of the appropriate Object Node is incremented by 1. If the search operation in the database_frequent is negative then search is performed in the database_all. In this search operation, the algorithm finds the number of times the object is occurring in the database_all. If this value exceeds a threshold, it assumes that the object is important otherwise the object is not important. If the object is important then the object is copied into database_frequent and is deleted from the database_all. Thus the frequently occurring objects are identified and kept in database_frequent.

Figure 7.3 depicts three objects of three different images. The objects O_1, O_2 and O_3 of the three images I_1, I_2 and I_3 respectively are having the same object indices as the shape of three of them are similar i.e. $O_1 = O_2 = O_3 = <\varphi^{o}_{1}, \varphi^{o}_{2}, \varphi^{o}_{3}>$. The set of color cluster indices of object O_1 matches with the set of color clusters of object O_3 and hence can be defined by <green, φ^{c1}_{1}, φ^{c1}_{2}, φ^{c1}_{3}> and <red, φ^{c2}_{1}, φ^{c2}_{2}, φ^{c2}_{3}>. The set of color cluster indices of object O_2 are defined by <blue, φ^{c1}_{1}, φ^{c1}_{2}, φ^{c1}_{3}> and <pink, φ^{c2}_{1}, φ^{c2}_{2}, φ^{c2}_{3}> though the parameters <φ^{c1}_{1}, φ^{c1}_{2}, φ^{c1}_{3}> and <φ^{c2}_{1}, φ^{c2}_{2}, φ^{c2}_{3} > of all the color indices are equal but the colors are different and hence the color cluster indices is not the same expect for the object O_1 and O_3. The entries of the object indices O_1 and O $_2$ of image I_1 and I_2 in the database_all structure is depicted in Figure 7.4.

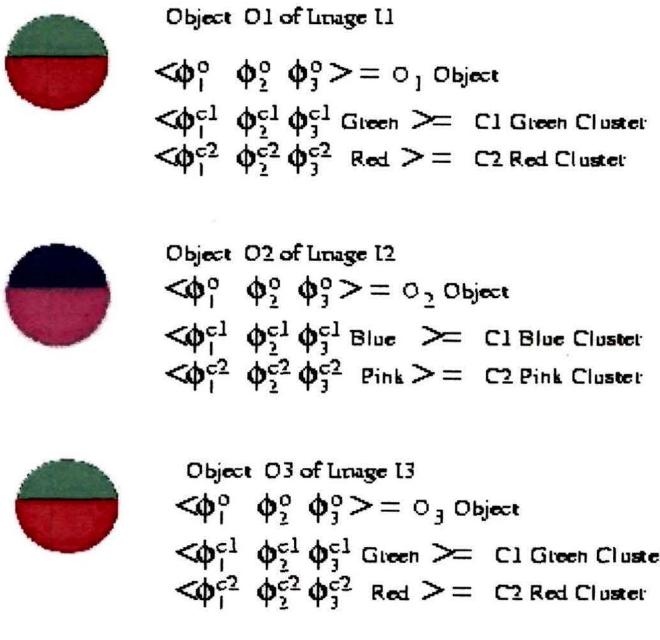

Object O1 of Image I1
$<\phi^{o}_{1}\ \phi^{o}_{2}\ \phi^{o}_{3}> = o_1$ Object
$<\phi^{c1}_{1}\ \phi^{c1}_{2}\ \phi^{c1}_{3}$ Green $>=$ C1 Green Cluster
$<\phi^{c2}_{1}\ \phi^{c2}_{2}\ \phi^{c2}_{3}$ Red $>=$ C2 Red Cluster
Object O2 of Image I2
$<\phi^{o}_{1}\ \phi^{o}_{2}\ \phi^{o}_{3}> = o_2$ Object
$<\phi^{c1}_{1}\ \phi^{c1}_{2}\ \phi^{c1}_{3}$ Blue $>=$ C1 Blue Cluster
$<\phi^{c2}_{1}\ \phi^{c2}_{2}\ \phi^{c2}_{3}$ Pink $>=$ C2 Pink Cluster
Object O3 of Image I3
$<\phi^{o}_{1}\ \phi^{o}_{2}\ \phi^{o}_{3}> = o_3$ Object
$<\phi^{c1}_{1}\ \phi^{c1}_{2}\ \phi^{c1}_{3}$ Green $>=$ C1 Green Cluster
$<\phi^{c2}_{1}\ \phi^{c2}_{2}\ \phi^{c2}_{3}$ Red $>=$ C2 Red Cluster

Figure 7.3: Example of three similar objects of three different images.

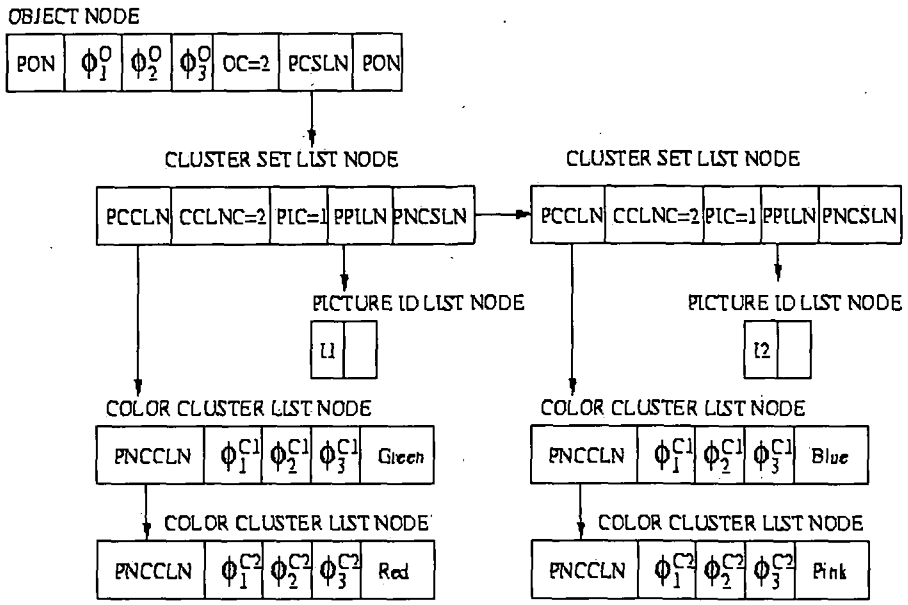


Figure 7.4: Database_all with two entries of objects O_1 and O_2

For a new object O_3 of an image I_3 (as depicted in Figure 7.3) whose object index is $<\varphi^{o}_{1}, \varphi^{o}_{2}, \varphi^{o}_{3}>$ and the set of cluster indices are <green, $\varphi^{c1}_{1}, \varphi^{c1}_{2}, \varphi^{c1}_{3}$> and <red, $\varphi^{c2}_{1}, \varphi^{c2}_{2}, \varphi^{c2}_{3}$>, the search is performed for the matching object index (Figure 7.4). In our example it is found and so the Object Count (OC) parameter in Figure 7.4 is incremented by one and its value becomes 3 in Figure 7.5. Then it searches in the Color Cluster List Nodes associated with the matching Object Node for a matching set of color clusters as that of the query. In our case a matching Color Cluster List Node is found and hence the Picture Id Count (PIC) is incremented by 1 and hence the value of the PIC is thus incremented to 2. Along with this the Picture Id I_3 is appended in the Picture Id List Node. If the threshold value is 2 then this compound Object with the object index and along with the associated Color Cluster indices are copied to database_frequent. Thus Figure 7.5 reflects the updated version of the Figure 7.4 which is a state of database_all after insertion of a new object O_3 of an image I_3.

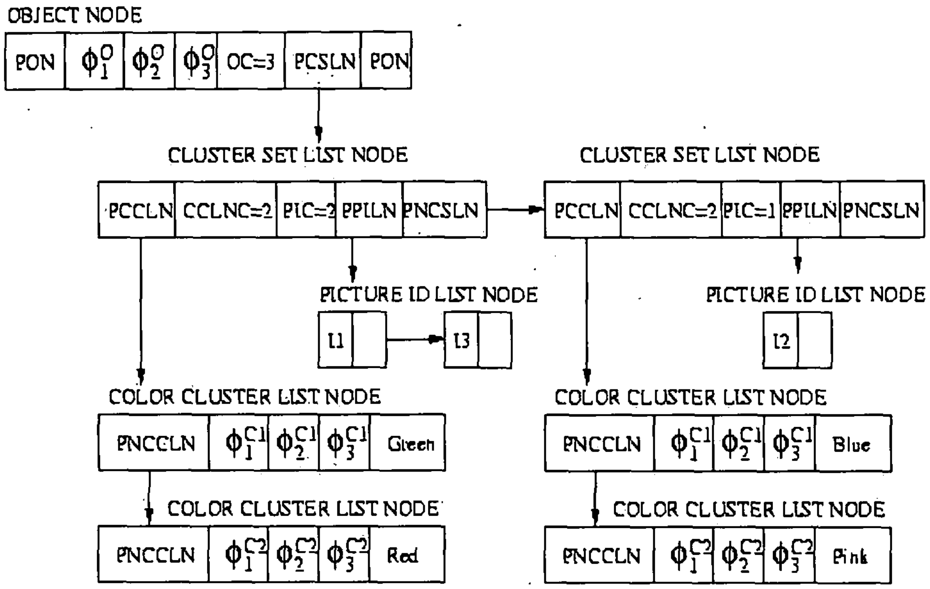


Figure 7.5: Database_all with three entries of objects O_1, O_2 and O_3.

7.4.2 Space Complexity of the Compound Index

For an input image there will be C number of clusters. For this *C* number of clusters there will be 2^C number of objects. After detection of objects of interest there may be 2^C number of objects at the worst case. In this case, there will be *C* numbers of clusters associated with at least one object. The compound index of an Image is constituted by number of objects present in the image and by number of clusters associated with each object. The object index is identified by the tuple $<\varphi^o_1, \varphi^o_2, \varphi^o_3>$ and the set of cluster indices are identified by *<color of the cluster,* $\varphi^c_1, \varphi^c_2, \varphi^c_3$*>.*

Therefore for the above case size of the index of clusters associated to an object in the worst case = C × <color of the cluster, $\varphi^c_1, \varphi^c_2, \varphi^c_3$> and the size of the index of the object in worst case = C × <color of the cluster, $\varphi^c_1, \varphi^c_2, \varphi^c_3$> + $<\varphi^o_1, \varphi^o_2, \varphi^o_3>$

Therefore the size of the compound index of the Image at the worst case

= 2^C × (C × <color of the cluster, $\varphi^c_1, \varphi^c_2, \varphi^c_3$> + $<\varphi^o_1, \varphi^o_2, \varphi^o_3>$)

If the parameters are taken as double then the space complexity becomes

= 2^C × 4C × 64 bits

= 32C × 2^C bytes.

7.5 Matching Engine

For the search operation a Matching Engine has been devised that has the provision for both *global* search and for *local* or *object* level similarity search for an input image. It uses the database_frequent and database_all as depicted in Figure 8.2 for searching images. The matching engine first searches in the database_frequent database and if the search is negative then it searches in the database_all. For both the global and for *local* or *object* level similarity search the algorithm calculates the compound object indices of the query image which consist of the object indices <$\varphi^o{}_{1q}$, $\varphi^o{}_{2q}$, $\varphi^o{}_{3q}$> and the set of color cluster indices <Color of the cluster, $\varphi^c{}_{1q}$, $\varphi^c{}_{2q}$, $\varphi^c{}_{3q}$> associated with each object index. For global similarity search, all the compound indices of the query image are matched with the stored compound indices of images in the database and retrieve those images that match all the compound indices of the query image. For local or object level similarity search, the user chooses some objects from the set of objects generated in the first phase of this algorithm. Similarity search is performed on the database for this set of selected objects. The algorithm retrieves all those images that match the selected compound indices of objects of the query image with the stored database compound indices of objects of images.

For comparison of indices of two objects or two clusters (i.e. say between <$\varphi^o{}_1$, $\varphi^o{}_2$, $\varphi^o{}_3$> (object of database) and <$\varphi^o{}_{1q}$, $\varphi^o{}_{2q}$, $\varphi^o{}_{3q}$> (object of query image) or between <color of cluster, $\varphi^c{}_1$, $\varphi^c{}_2$, $\varphi^c{}_3$> (cluster of database) or <color of cluster$_q$, $\varphi^c{}_{1q}$, $\varphi^c{}_{2q}$, $\varphi^c{}_{3q}$> (cluster of query image)) cosine measure is used. A provision for similarity matching has been kept as user input for accuracy measure. By selecting different accuracy measure the user can converge the search operation to his/her requirement.

7.5.1 Data Structure and Symbols Used for Modified Matching Engine.

Compound_Index : Link list that holds the compound indices of objects of the query image.

Result[i] : Array to hold matching result of each compound objects compared.

Database : Variable to store the name of the database on which search is to be done.

Steps of Modified Matching Engine

```
Step 1: For each compound object index in the Compound_Index list
Step 2: i = 0;
        Database = "database_frequent"
Step 3: Match the <$\varphi^o_{1q}$, $\varphi^o_{2q}$, $\varphi^o_{3q}$> object index of the query compound object index with the
        <$\varphi^o_1$, $\varphi^o_2$, $\varphi^o_3$> object index of the stored compound indices in the Database.
        if matching found
        begin
                Match the set of <color of cluster$_q$, $\varphi^c_{1q}$, $\varphi^c_{2q}$, $\varphi^c_{3q}$> cluster indices of the query
                compound object index with the stored <color of cluster, $\varphi^c_1$, $\varphi^c_2$, $\varphi^c_3$> of the
                cluster indices in the Database.
                if matching found
                        Return associated Picture_Id list;
                        Result[i] = "Matches";
                        i = i + 1;
                        go to Step 1 for next object;
                else
                        if (Database = "database_frequent")
                                Database = "database_all";
                                go to Step 3
                        else
                                Result[i] = "Does Not Match";
                                i = i + 1;
                                go to Step 1 for next object;
                        endif
                endif
        end
        else
        begin
                go to Step 1 for next object;
        end
```

7.6 Complexity Analysis and Comparison

The retrieval complexity of both the schemes depends on the quantization of the image and the clusters produced by the clustering algorithm. Moreover, the retrieval time comparison has been done for Global search and not for Regional or Object level search. Because in regional search, clusters has to be selected manually for object formation in scheme I and hence computation time will vary depending on the object selection for the three schemes. Also, the same object may not be selected by the user for both the Schemes for retrieval purpose as the indexing scheme of clusters and objects are not the same for both the Schemes. But, for Global search both the Schemes generates indices of clusters and objects automatically and search is performed on all the cluster indices for Scheme I and Scheme II of the image. Here manual selection of objects or clusters is not necessary. Thus the retrieval time in this case will be from submission of the query image for similarity search to the retrieval of similar images from the image database.

S = Total number of images in the image database

m, n = Defines the size of the image

N = m×n the total number of pixels in the image

C = Is the total number of Clusters detected in the image.

p = Number of pixels in a cluster

O = Number of objects detected in the image

Scheme I

1. Total retrieval time taken by Scheme I is 5N + SlogS

Scheme II

1. Time taken for Adaptive Quantization = Entropy calculation of the whole image + Estimation of the window size defined by d_n + Final quantization of the Image

   is N + (N×m) + N

   which is N(2+m)

   which is (N×m)
2. Time taken for Cluster generation by modified BOO-Clustering algorithm

   is (N×m)
3. Time taken for Detection of Objects of Interest in an image

   is $2^C -1$
4. Time taken for Compound index calculation for one object

   = Silhouette moment calculation for the whole object + Silhouette moment calculation for the set of clusters associated with the object

   is $(4 \times N + \psi + \nu) + (2^C - 1) \times (4 \times N + \psi + \nu)$

   which is $4N + 4N \times 2^C$

   which is $4N \times 2^C$
5. Time taken for compound index calculation for $(2^C - 1)$ number of objects

   is $(4N \times 2^C) \times (2^C - 1)$

   which is $4N \times 2^{2C}$
6. Retrieval time is $(2^C - 1) \times$ SlogS

   Which is $2^C \times$ SlogS
7. Total retrieval time taken by Scheme III

   = (5) + (6) = $4N \times 2^{2C} + 2^C \times$ SlogS

   Which is $2^C \times (4N\times2^C +$ SlogS)

In all the three Schemes a variant of B-tree is used for indexing purpose. Hence the SlogS part of the retrieval time complexity will nearly remain the same. The complexity will mainly depend on the calculation of the index part. It can be clearly seen from the complexity analysis above that Scheme I is faster than Scheme II (Figure 7.8). But if the retrieval performance is compared it can be clearly seen that Scheme II is better than Scheme I (Figure 7.6).

7.7 Experimental Results

To test the technique, we used a downloaded database ((a) Cohn-Kanade Facial Expression Database; http://www.pitt.edu/~jeffcohon, (b) ftp://ftp.eecs.umich.edu.groups /ai/dberwick/essbthm.zip and other images) consisting of 10000 real world and synthetic images divided into 100 similar groups such as *facial expressions, scenery, animals, cars, flowers* and *space*. For example, the facial expression database consists of nearly 5000 images of different persons and for each person there are nearly 30 different photographs with different facial expressions. All these images have been taken at a particular direction (facing directly) and at a fixed distance from the camera having the same background color. These facial images are of same aspect ratio. Implementation was carried out for Scheme I and Scheme II techniques. All the methods have been implemented in Java platform in an Intel Pentium IV machine. For any input image, indices are generated for the color clusters and objects of interest. Similarity search is done on objects of interest using a matching engine. We have performed a two way test to establish the supremacy of the BOO-Clustering method over the Chromatic Moment and GDBSCAN methods. They are:

1. Retrieval effectiveness of the three methods for global and regional search.
2. Retrieval time comparison by Scheme III, Scheme II and Scheme I for Global Search.

7.7.1 Retrieval Effectiveness for Global and Regional Search

Retrieval effectiveness is calculated by the *average precision recall* (APR) curve as discussed in *Chapter 5*. In our experiment, the APR was plotted for the best 200 queries out of the calculated 500 queries taken at random over the downloaded image database.

The proposed Scheme II was compared with Scheme I for both global and regional search. In global search, all objects produced by both the methods for a query image are taken into account for the search operation. Figure 7.6 reflects the query result of global search of the three techniques. Based on experimental results, it is found that the performance areas for the curves are: 448 (Scheme II), 386 (Scheme I). Hence the improvement produced by the proposed Scheme II over the Scheme I is: 16.06 %.

Scheme II was tested based on some specific objects formed by combination of some selected clusters or a single cluster that forms an object. In case of Scheme I, those clusters are chosen from the query image to form the specific object as that chosen by Scheme II of the query image. Figure 7.7 reflects the query result of the regional search of the two techniques. Based on the experimental results, it is found that the performance areas for the curves are: 411 (Scheme II), 373 (Scheme I). Hence the improvement produced by Scheme II over Scheme I is 10.19 %.

7.7.2 Retrieval Time Comparison of Scheme III, Scheme II and Scheme I for Global Search

In this experiment a normalized database as discussed in the Complexity Analysis and Comparison Section (7.6) has been used. For different database sizes (starting from 500 images to 10000 images) fifty queries were done for each database size. For each query precision, recall and time taken for retrieval are recorded. Out of these fifty queries for a database size, the best two in terms of precision recall is taken. With these data two graphs are drawn: 1) Average Precision Recall for global search (Figure 7.6) and 2) Retrieval Time Comparison (Figure 7.8). It can be clearly seen from the graphs that Scheme II is more effective in terms of retrieval at the cost of time.

.

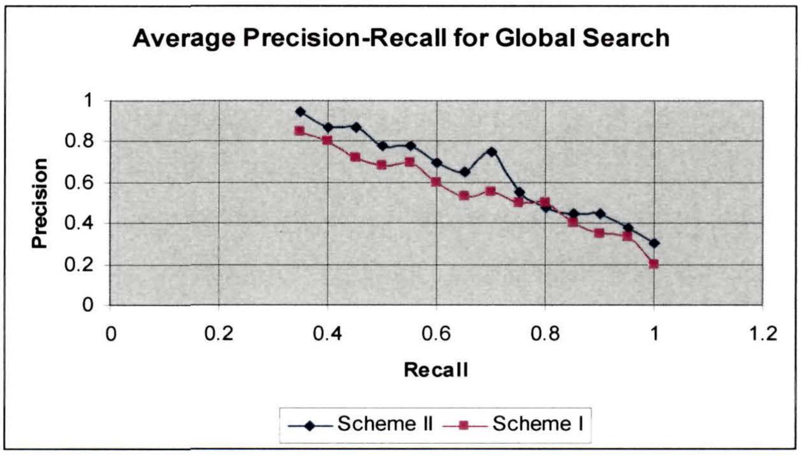


Figure 7.6. Precision Recall for Scheme I and Scheme II for Global search.

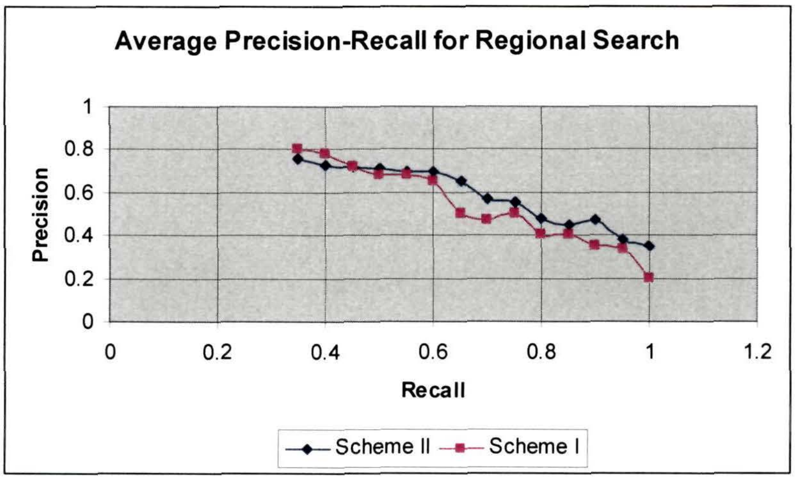


Figure 7.7. Precision Recall for Scheme I, and Scheme II for Regional search.

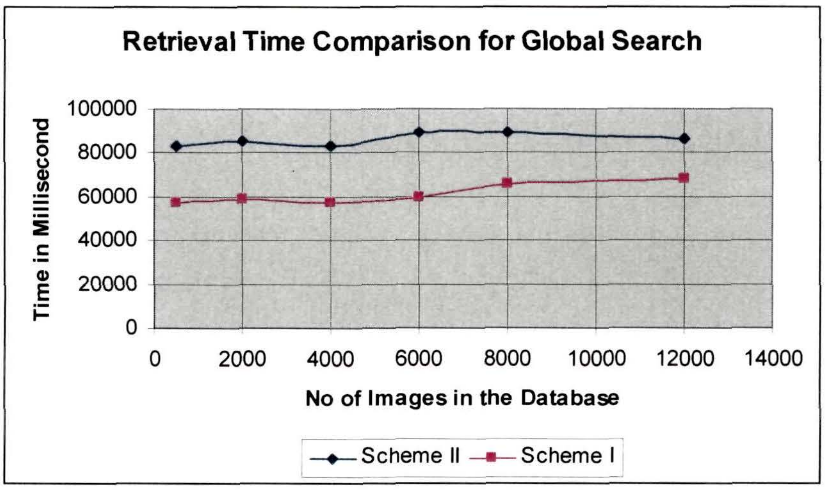
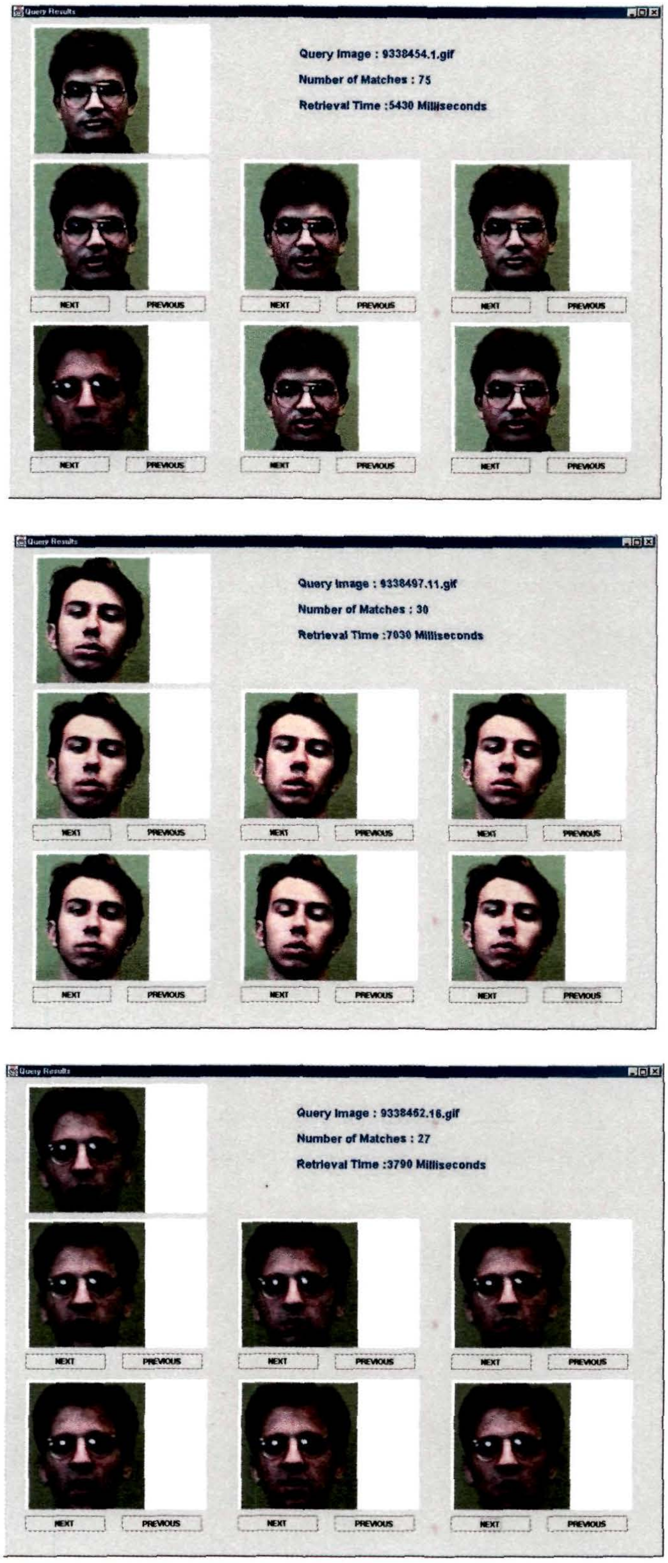


Figure 7.8. Retrieval time comparison for Global Search.

7.8 Discussion

In this scheme quantization is performed based on the distribution of colors present in the image. No input parameter is necessary for clustering the pixels of the image. Indices of all objects are not kept for searching purpose. But those objects are kept which are found important. This detection of objects is done based on the percentage frequency of occurrence of the objects. This process of detection of objects is fully automatic. Second order Silhouette moments are used for index generation of the objects. These indices of objects are stored in a space efficient tree structure. An efficient Matching engine has been devised for searching objects in the tree structure.



Query Image : 9338454.1.gif

Number of Matches : 75

Retrieval Time :5430 Milliseconds

Query Image : 9338497.11.gif

Number of Matches : 30

Retrieval Time :7030 Milliseconds

Query Image : 9338462.16.gif

Number of Matches : 27

Retrieval Time :3790 Milliseconds

Figure 7.9. Some retrieval results of Scheme III

Chapter 8

Conclusions and Future Works

It is a truism to observe that images are currently used in every walk of life. Images are increasingly used to convey information in areas as diverse as map-making, weather forecasting and to persuade or convey a mood, as in advertising. The increasing use of images by various categories of people necessitates storage and retrieval of images in an efficient manner.

Digitized images consist purely of arrays of pixel intensities, with no inherent meaning. CBIR is used for retrieving desired images from a large collection of such images on the basis of features (such as color, texture and shape) that can be automatically extracted from the images themselves. In CBIR useful information (such as the presence of a particular color, shape or texture) has to be extracted from raw data and retrieval has to be done based on the similarity measure of this raw data without assigning any keywords to the images. Thus it is a challenging task to develop a fully automatic CBIR system that operates successfully over a wide range of applications.

The work presented in this thesis is an attempt towards such a CBIR framework. Based on our theoretical and implementation experiences following concluding remarks are made.

- A clustering technique has been established to be very useful in the context of efficient CBIR system design. A faster and quality clustering technique BOO-Clustering has been presented in the thesis. The technique has been established to perform satisfactorily in comparison to its other counterparts.
- We have explored three different schemes in the thesis for CBIR which work based on the proposed clustering technique. Starting from a semi-automatic CBIR ((Scheme I) we move towards a fully automatic CBIR system (Scheme II).

- In Scheme I as well as in Scheme II, clusters as well as objects are considered for index generation. However, in Scheme I, object selection is done manually. The variant of BOO-Clustering used in Scheme II is same as that of Scheme I. In Scheme I, second order silhouette moments are calculated for the selected objects and clusters. The first three second order silhouette moments and average color of the object or the color of the cluster are used as index. The index is invariant with respect to rotation, translation, uniform and non-uniform scaling. These indices are stored in a spatial cluster-object tree which helps for searching images. A matching engine has been devised which helps in global and regional search. The main drawback of this scheme is that the objects are to be selected manually. Hence it is time consuming and user subjective - improper selection of clusters (to form an object) may take place.
- In Scheme II, a new adaptive color quantization method is introduced. In contrast to Scheme I, Scheme II uses a parameter less BOO-Clustering algorithm that produces the same number of useful color clusters of an image irrespective of number of times it is executed for the same image. A heuristic approach (termed as learning objects of interest) is used to detect the important clusters to form objects in an image. A compound index of the object is formed by the first three second order silhouette moments and color of the clusters of the object. These indices are stored using similar tree structures to facilitate the search operation. Also a modified matching engine has been devised for global and regional search.

However, it has been strongly felt that following enhancements can be done in the near future.

- Presently learning mechanism is based on frequency of occurrence, which can be extended by considering the other objective functions such as interestingness, comprehensiveness etc.
- Query submission method can be further extended by incorporating keyword annotations.
- Scheme III can be further enhanced by incorporating a cluster based searching mechanism to facilitate efficient search.

List of Publications

1. P. J. Dutta, D. Bhattacharryya, J. Kalita, and M. Dutta. "Spatial Color Indexing Using Clustering Technique", *The 8th World Multi-Conference on Systemics, Cybernatics and Informatics (SCI' 2004).* Vol VI: Image, Acoustics, Signal Processing & Optical Systems, Technologies and Applications, Orlando, Florida, pp. 216-221, July 2004.

2. P. J. Dutta, D Bhattacharyya, J Kalita and M Dutta, "Clustering Approach to Content Based Image Retrieval," Geometric Modeling and Imaging-New Trends (GMAI06), London, 2006.

Bibliography

[AAK02] Atsalakis, A., Kroupis, N., Soudris, D., Papamarkos, N. "A window-based color quantization technique and its embedded implementation", *IEEE ICIP 2002* 23, 365–368, 2002.

[ABD79] A. Blaser, Database Techniques for Pictorial Applications, *Lecture Notes in Computer Science*, Vol.81, Springer Verlag GmbH, 1979.

[ABH01] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. "Support vector clustering", *J. of Machine Learning Research*, vol. 2, pp. 125-137, 2001.

[ADN77] A. Dempster, N. Laird, and D. Rubin. "Maximum likelihood estimation from incomplete data via the EM algorithm". *J. Royal Statistical Soc.* B, 39:1-38, 1977.

[AEC93} A. E. Cawkill, "The British Library's Picture Research Projects: Image, Word, and Retrieval,"*Advanced Imaging*,Vol.8, No.10, pp.38-40, October 1993.

[AJR97] A.J. Round, A.W.G. Duller, and P.J. Fish, "Color Segmentation for Lesion Classification," *Proc. of the 19th Annual Int'l Conf. of the IEEE Engineering in Medicine and Biology Society*, Chicago, IL, 30 Oct.-2 Nov. 1997, Vol. 2 pp. 582-585.

[AKJ00] A.K. Jain, R. Dubes, and J. Mao. "Statistical pattern recognition: A review". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):4-38, January 2000.

[AKJ88] A.K. Jain and R. Dubes. *Algorithms for Clustering Data*. Prentice Hall, April 1988.

[AKJ89] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[AKJ91] A. K. Jain, and F. Farroknia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, Vo.24, No.12, pp. 1167-1186, 1991.

[AKJ92] Amlan Kundu and Jia Lin Chen. Texture classification using qmf bank-based subband decomposition. *CVGIP: Graphical Models and Image Processing*, 54(5), 369-384, September 1992.

[ALJ93] Andrew Laine and Jian Fan. Texture classification by wavelet packet signatures. *IEEE Trans. Patt. Reciog. and Machine Intell.*, 15(11): 1186-1191, 1993.

[AMN94] A. Moghaddamzadeh and N. Bourbakis, "A Fuzzy Technique for Image Segmentation of Color Images," *Proc. of the 3rd Int'l Fuzzy Systems Conference*, Orlando, FL, 26-29 June 1994, Vol. 1, pp. 83-88.

[AMN97] A. Moghaddamzadeh and N. Bourbakis, "A Fuzzy Region Growing Approach for Segmentation of Color Images," *Pattern Recognition*, Vol. 30, No. 6, pp. 867-881, June 1997.

[AMW00] A. M. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.22, No.12, pp. 1349-1380, Dec. 2000.

[APW96] A. Pentland, R. W. Picard and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision,* 1996.

[ARA82] A. Rosenfeld and A. Kak, "Digital Picture Processing", Vol. 2, 2nd Ed., Academic Press, New York, NY, 1982.

[ARR99] A. Rao, R K. Srihari, and Z. Zhang. Spatial color histograms for content-based image retrieval. *11th IEEE International Conference on Tools with Artificial Intelligence.* Chicago, Illinois, November, 1999.

[ATN97] A. Tremeau and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation," *Pattern Recognition*, Vol. 30, No. 7, pp. 1191-1204, July 1997.

[AVM01] A. Vailaya, M. A. G. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for content-based indexing," *IEEE Trans. on Image Processing*, Vol.10, No.1, Jan. 2001.

[BCM97] B. Cramariuc, M. Gabbouj, and J. Astola, "Clustering Based Region Growing Algorithm for Color Image Segmentation," *Proc. of the 13th Int'l Conf. on Digital Signal Processing*, Santorini, Greece, 2-4 July 1997, Vol. 2,pp. 857-860.

[BFS95] B. Furht, S. W. Smoliar, and H.J. Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, 1995.

[BLS95] B. Li and Song De Ma. On the relation between region and contour representation. In *Proc. IEEE Int. Conf. on Image Proc.,* 1995.

[BMM97] Babu M. Mehtre, M. Kankanhalli and Wing Foon Lee. Shape measures for content based image retrieval: A comparison. *Information Processing and Management*, 33(3), 1997.

[CBT03] C. Bauckhage, T. Kaster, M. Pfeiffer, and G Sagerer. "Content-Based Image Retrieval by Multimodal Interaction", *Proc. IEEE IECON'03*, pp. 1865-1870, Roanoke, VA, 2003.

[CCG90] C. Calvin, Gotlieb and Herbert E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics and Image Processing*, 51, 1990.

[CCM00] Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and JitendraMalik. Blobworld: A system for region-based image indexing and retrieval. In Huijsmans and Smeulders [DPH99].

[CCM99] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik, "Blobworld: A System for Region-Based Image Indexing and Retrieval", *Proc. Visual Information Systems*, pp. 509-516, June 1999.

[CCM99] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," In D. P. Huijsmans and A. W. M. Smeulders, ed. *Visual Information and Information System, Proceedings of the Third International Conference VISUAL'99*, Amsterdam, The Netherlands, June 1999, Lecture Notes in Computer Science 1614. Springer, 1999.

[CCV96] Chad Carson and Virginia E. Ogle. Storage and retrieval of feature data for a very large online image collection. *IEEE Computer Society Bulletin of the Technical Committee on Data Engineering*, 19(4):19–27, December 1996.

[CFE94] C. Faloutsos et al, "Efficient and effective querying by image content," *Journal of intelligent information systems*, Vol.3, pp.231-262, 1994.

[CGH96] C. Gene, H. Chuang and C. C. Jay Kuo. Wavelet descriptor of planner curves: Theory and applications. *IEEE Transaction on Image Procedings,* 5(1):56-70, January, 1996.

[CSA97] C. Scheering and A. Knoll. "Fast color image segmentation using a pre-clustered chromaticity-plane". In *Proc. of 1997 IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing, ICASSP'97*, volume 4, pages 3145-3147, April 1997.

[CTZ71] C. T. Zahn. "Graph-theoretical methods for detecting and describing gestalt glusters". *IEEE Trans. on Computers*, 20:68-86, 1971.

[CTZ72] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. on Computers*, 1972.

[DCM97] D. Comaniciu and P. Meer. "Robust analysis of feature spaces: Color image segmentation". In *Proc. of CVPR97*, volume 1, pages 750-755, 1997.

[DCP02] D. Comaniciu and P. Meer. "Mean shift: A robust approach toward feature space analysis". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603-619, May 2002.

[DCP03] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based object tracking". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564-577, May 2003.

[DCP97] D. Comaniciu and P. Meer. "Robust analysis of feature spaces; color image segmentation", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* pp. 750-755, 1997.

[DCP99] D. Comaniciu and P. Meer. "Mean shift analysis and applications". In *Proc. IEEE Int. Conf. on Computer Vision*, volume 1, pages 1197-1203, 1999.

[DCT95] D.C. Tseng, Y.F. Li, and C.T. Tung. "Circular histogram thresholding for color image segmentation". In *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition*, volume 2, pages 673-676, August 1995.

[DCZ95] D. Copper and Z. Lei. On representation and invariant recognition of complex objects based on patches and parts. *Spinger Lecture Notes in Computer Science series, 3D Objects Representation for Computer Vision,* pages 139-153, 1995.

[DGM02] D. Goldman, M. Yang, and N. Bourbakis. "A neural network-based segmentation tool for color images". In *Proc. 14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002),* pages 500-511, November 2002.

[DHB81] D. H. Ballard. Generalized Hough transform to detect arbitrary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):111–122, 1981.

[DKP95] D.K. Panjwani and G. Healey, "Markov Random Field Models for Unsupervised Segmentation of Textured Color Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-17, No. 10, pp. 939-954, Oct. 1995.

[DKY95] Deepak Kapur, Y. N. Lakshman, and Tushar Saxena. Computing invariants using elimination methods. In *Proc. IEEE Int. Conf. on Image Proc.*, 1995.

[DPH99] D. P. Huijsmans and A. W. M. Smeulders, editors. *Visual Information and Information Systems, Proceedings of the Third International Conference VISUAL '99, Amsterdam, The Netherlands, June 1999*, Lecture Notes in Computer Science 1614. Springer, 1999.

[DTS94] D. Tegolo, "Shape analysis for image retrieval," *Proc. of SPIE, Storage and Retrieval for Image and Video Databases -II*, no. 2185, San Jose, CA, pp. 59-69, February 1994.

[DWR96] D. White and R. Jain. Similarity indexing with the SS-tree. In *Proceedings of the 12th International Conference on Data Engineering, New Orleans, LA*, 1996.

[EMA91] E. M. Arkin, L.P. Chew, D..P. Huttenlocher, K. Kedem, and J.S.B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209-226, 1991.

[EMC98] E. Mathias, "Comparing the influence of color spaces and metrics in content-based image retrieval," *Proceedings of International Symposium on Computer Graphics, Image Processing, and Vision*, pp. 371 -378, 1998.

[EML91] Esther M. Arkin, L. Chew, D. Huttenlooher, K. Kedem, and J Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transaction Pattern Recognition and Machine Intelligence*, 13(3), March 1991.

[EPK77] E. Persoon, and K. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, and Cybern.*, Vol. 7, pp. 170-179, 1977.

[ESA95] E. Saber, A.M. Tekalp, R. Eschbach, and K. Knox. “Annotation of natural scenes using adaptive color segmentation”. In *Proc. of the SPIE - The Int'l Soc. for Optical Eng., Image and Video Proc.*, volume III, pages 72-80, February 1995.

[ESA96] E. Saber, A.M. Tekalp, and G. Bozdagi. “Fusion of color and edge information for improved segmentation and edge linking”. In *Proc of 1996 IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing, ICASSP'96*, volume 4, pages 2176-2179, May 1996.

[EVR00] Excalibur visual retrievalware. http://vrw.excalib.com/.

[FGJ98] F. Guo, J. Jin, and D. Feng, "Measuring image similarity using the geometrical distribution of image contents", *Proc. of ICSP*, pp.1108-1112, 1998.

[FPC94] F. Perez and C. Koch. “Toward color image segmentation in analog VLSI: Algorithm and hardware”. *Int'l Journal of Computer Vision*, 12(1):17-42, February 1994.

[FZB98] F. Ziliani and B. Jensen, “Unsupervised Image Segmentation Using the Modified Pyramidal Linking Approach,” *Proc. of 1998 Int'l Conf. on Image Processing (ICIP'98)*, Chicago, IL, 4-7 Oct. 1998, Vol. III, pp. 303-307.

[GDF96] G. D. Finlayson, "Color in perspective," *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol.8, No. 10, pp.1034-1038, Oct. 1996.

[GJK90] G.J. Klinker, S.A. Shafer, and T. Kanade, “A Physical Approach to Color Image Understanding,” *Int'l Journal of Computer Vision*, Vol. 4, pp. 7-38, 1990.

[GMD00] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[GMP88] Gerrautz, M., Purgathofer, W. “A simple method for color quantization: Octree quantization.” *Proceedings of CG International_88*, pp. 219–230, 1988.

[GMT97] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, 1997.

[GPI03] G. Paschos, I. Radev, N, Prabakar, Image "Content-Based Retrieval Using Chromaticity Moments", *IEEE Transactions on Knowledge and Data Engineering*, Vol 15, No 5, pp. 1069-1072, September/October 2003.

[GPR96] G. Pass, and R. Zabith, "Histogram refinement for content-based image retrieval," *IEEE Workshop on Applications of Computer Vision*, pp. 96-102, 1996.

[GPR99] G.Pass, and R. Zabith, "Comparing images using joint histograms," *Multimedia Systems*, Vol.7, pp.234-240, 1999.

[GRC83] G.R. Cross and A.K. Jain, "Markov Random Field Texture Models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 1, pp. 25-39, Jan. 1983.

[GSC97] G. Sapiro. "Color snakes". *Computer Vision and Image Understanding*, 68(2):247-253, November 1997.

[GSV96] G. Sapiro. "Vector (self) snakes: A geometric framework for color, texture and multiscale image segmentation". In *Proc. Int. Conference on Image Processing, volume 1*, pages 817-820, 1996.

[GTR91] Gabriel Taubin. Recognition and positioning of rigid objects using algebraic moment invariants. In *SPIE Vol 1570 Geometric Methods in Computer Vision*, 1991.

[GYM98] G. Guo, S. Yu, and S. Ma. "Unsupervised segmentation of color images". In *Proc. of 1998 Int'l Conf. on Image Processing, ICIP'98*, volume III, pages 299-302, October 1998.

[HAR98] H. A. Rowley, S. Baluja, and T. Kanade. "Neural network-based face detection". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23-38, 1998.

[HBS00] H. Burkhardt, and S. Siggelkow, "Invariant features for discriminating between equivalence classes," *Nonlinear Model-based Image Video Processing and Analysis*, John Wiley and Sons, 2000.

[HCF00] H.C. Fu, P.S. Lai, R.S. Lou, and H.T. Pao. "Face detection and eye localization by neural network based color segmentation". In *Proceedings of the 2000 IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing X, 2000*, volume 2, pages 507-516, 2000.

[HDC77] H. Digabel and C. Lantuejoul, "Iterative Algorithms", *Proc. of the 2nd European Symp. on Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, Caen, France, 1977*

[HFK91] H. F. Korth and A. Silberschatz. *Database System Concepts*. McGraw-Hill, 1991.

[HGB77] H. G. Barrow. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc 5th Int. Joint Conference Artificial Intelligence*, 1977.

[HJF95] H. J. Zhang, and D. Zhong, "A Scheme for visual feature-based image indexing," *Proc. of SPIE conf. on Storage and Retrieval for Image and Video Databases III*, pp. 36-46, San Jose, Feb. 1995.

[HJO95] H. J. Zhang, *et al*, "Image retrieval based on color features: An evaluation study," *SPIE Conf. on Digital Storage and Archival*, Pennsylvania, Oct. 25-27, 1995.

[HKT95] H. Kauppinen, T. Seppnäen, and M. Pietikäinen, "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification," *IEEE Trans. Pattern Anal. and Machine Intell.*, Vol. 17, No. 2, pp. 201-207, 1995.

[HON94] H. Okii, N. Kaneki, H. Hara, and K. Ono. "Automatic color segmentation method using a neural network model for stained images". *IEICE Trans. on Information and Systems (Japan)*, E77-D(3):343-350, March 1994.

[HST84] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys*, Vol.16, No.2, pp.187-260, 1984.

[HTM78] Hideyuki Tamura, Shunji Mori and Takashi Yamawaki. Texture features corresponding to visual perception. *IEEE Trans. on Sys. Man. and Cyb*, SMC-8(6), 1978.

[HTN84[H. Tamura, and N.Yokoya, "Image database systems: A survey, " *Pattern Recognition*, Vol.17,.No.1, pp. 29-43, 1984.

[HVJ91] H. V. Jagadish, "A retrieval technique for similar shapes," *Proc. of Int. Conf. on Management of Data, SIGMOID'91*, Denver, CO, pp. 208-217, May 1991.

[HVT88] H. Voorhees, and T. Poggio. "Computing texture boundaries from images," *Nature*, 333:364-367, 1988.

[HWF98] H. Wang, F. Guo, D. Feng, and J. Jin, "A signature for content-based image retrieval using a geometrical transform," *Proc. Of ACM MM'98*, Bristol, UK, 1998.

[ISH00] Hsieh, Kuo-Chin Fan. "An adaptive clustering algorithm for color quantization," *Pattern Recognition Letters* 21 (2000) 337-346, 2000.

[JAA00] J. Assfalg, A. D. Bimbo, and P. Pala, "Using multiple examples for content-based retrieval," *Proc. Int'l Conf. Multimedia and Expo*, 2000.

[JAC00] J.A. Catalan, and J.S. Jin, "Dimension reduction of texture features for image retrieval using hybrid associative neural networks," *IEEE International Conference on Multimedia and Expo*, Vol.2, pp. 1211 -1214, 2000.

[JBC96] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Gorowitz, R. Humphrey, R. Jain, and C. Shu. Virage image search engine: an open framework for image management. In *Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases IV, San Jose, CA*, pages 76–87, February 1996.

[JDC93] J. Dowe, "Content-based retrieval in multimedia imaging," *In Proc. SPIE Storage and Retrieval for Image and Video Database*, 1993.

[JDF90] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer graphics: principles and practice*, 2nd ed., Reading, Mass, Addison-Wesley, 1990.

[JEG92] J. E. Gary, and R. Mehrotra, "Shape similarity-based retrieval in image database systems," *Proc. of SPIE, Image Storage and Retrieval Systems*, Vol. 1662, pp. 2-8, 1992.

[JGX93] Joy, G., Xiang, Z. "Center-cut for color image quantization," *Visual Computation* 10 (1), 62–66, 1993.

[JHE95] J. Hafner, *et al.*, "Efficient color histogram indexing for quadratic form distance functions," IEEE *Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, pp. 729-736, July 1995.

[JHI97] J. Huang, *et al.*, "Image indexing using color correlogram," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 762-768, Puerto Rico, June 1997.

[JHS97] J. Huang, S. R. Kumar, and M. Metra, "Combining supervised learning with color correlograms for content-based image retrieval," *Proc. of ACM Multimedia'95*, pp. 325-334, Nov. 1997.

[JHS99] J. Huang, S.R. Kumar, M. Metra, W. J., Zhu, and R. Zabith, "Spatial color indexing and applications," *Int'l J. Computer Vision*, Vol.35, No.3, pp. 245-268, 1999.

[JLR97] J. Luo, R.T. Gray, and H.C. Lee. "Towards physics-based segmentation of photographic color image". In *Proc. of 1997 Int'l Conference on Image Processing, ICIP'97*, volume III, pages 58-61, October 1997.

[JLR98] J. Luo, R.T. Gray, and H.C. Lee. "Incorporation of derivative priors in adaptive Bayesian color image segmentation". In *Proc. of 1998 Int'l Conf. on Image Processing, ICIP'98*, volume III, pages 780-784, October 1998.

[JLY94] J. Liu and Y.-H. Yang, "Multi-resolution Color Image Segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-16, No. 7, pp. 689-700, July 1994.

[JMB00] Jean Marie Buijs andMichael Lew. Visual learning of simple semantics in imagescape. In Huijsmans and Smeulders [DPH99], pages 131–138.

[JMZ00] J. M. Zachary. "An Information Theoretic Approach to Content Based Image Retrieval", *PhD thesis*, Louisiana State University, 2000.

[JNH84] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The grid file: an adaptable symmetric multikey file structure," *ACM Trans. on Database Systems*, pp. 38-71, March 1984.

[JPW98] J. P. Wang. "Stochastic relaxation on partitions with connected components and its application to image segmentation". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):619-635, June 1998.

[JRS94] John R. Smith and Shih-Fu Chang. Transform features for texture classification and discrimination in large image databases. In *Proc. IEEE Int. Conf. on Image Proc.*, 1994.

[JRS96] John R. Smith and Shih-Fu Chang. Automated binary texture feature sets for image retrieval. In *Proc. ICASSP-96,* Atlanta, GA, 1996.

[JRS97] J. R. Smith and S.-F. Chang. Querying by color regions using the VisualSEEk content-based visual query system. In M. T. Maybury, editor, *Intelligent Multimedia Information Retrieval.* AAAI Press, 1997.

[JSJ97] J. Shi and J. Malik. "Normalized cuts and image segmentation". In *Proc. of Int. Conf. Computer Vision and Pattern Recognition*, pages 731-737, 1997.

[JSM00] J Sander, M Ester, HP Kriegel, X Xu. "Density-Based Clustering Spatial Databases: The Algorithm GDBSCAN and its Applications:" http://Citeseer.nj.nec.com

[JSS96]· J. Smith and S. F. Chang, "Tools and Techniques for Color Image Retrieval", *SPIE Proceedings,* pp. 1630-1639, 1996.

[JSS98] J. Shi, S. Belongie, T. Leung, and J. Malik. "Image and video segmentation: The normalized cut framework". In *Proc. of Int. Conf. Image Processing*, volume 1, pages 943-947, 1998.

[JTR81] J. T. Robinson, "The k-d-B-tree: a search structure for large multidimensional dynamic indexes," *Proc. ofSIGMOD Conference*, Ann Arbor, April 1981.

[JVM99] J. Vendrig, M. Worring, and A. W. M. Smeulders, "Filter image browsing: exploiting interaction in retrieval," *Proc. Viusl'99: Information and Information System*, 1999.

[KAW90] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant Fourier descriptors to recognition of 3D objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 640-647, 1990.

[KHS97] Kyoji Hirata, Sougata Mukherjea, Yusaku Okamura, Wen-Syan Li, and Yoshinori Hara. Objectbased navigation: An intuitive navigation style for content-oriented integration environment. In *Proceedings of the eighth ACM conference on Hypertext, '97, Southampton, UK,* pages 75–86, 1997. http://journals.ecs.soton.ac.uk/˜lac/ht97/.

[KHY93] Kyoji Hirata, Yoshinori Hara, Naoki Shibata, and Fusako Hirabayashi. Media-based navigation for hypermedia systems. In *Proceedings of the fifth ACM conference on Hypertext, November 14-18, '93, Seattle, WA, USA*, pages 159–173, 1993.

[KSC94] K. Saarinen, "Color Image Segmentation by a Watershed Algorithm and Region Adjacency Graph Processing," Proc. of 1994 Int'l Conf. on Image Processing (ICIP'94), Austin, TX, 13-16 Nov. 1994, Vol. III, pp. 1021-1025.

[KSF81] K. S. Fu and J. K. Mui. "A survey on image segmentation". *Pattern Recognition,* 13:3-16, 1981.

[KSI96] K. Sobottka and I. Pitas. "Segmentation and tracking of faces in color images". In *Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition*, pages 236-241, 1996.

[KST94] K. S. Thyagarajan, Tom Nguyen and Charles Persons. A maximum likelihood approach to texture classification using wavelet transform. In *Proc. IEEE Int. Conf. on Image Proc.*, 1994.

[KTK99] K. Takahashi and K. Abe. "Color image segmentation using isodata clustering algorithm". *Trans. of the Institute of Electronics, Information and Communication Engineers D-II,* J82D-II(4):751-762, April 1999.

[KUC94] K. Uchimura. "Color images segmentation using tree representation". *Trans. of the Institute of Electrical Engineers of Japan,* 114-C, Part C(12):1320-1321, December 1994.

[LJL94] L. J. Liu, J. F. Lu, J. Y. Yang, K. Liu, Y.G.Wu, and S. J. Li. "Efficient segmentation of nuclei in different color spaces". In *Proc. of the SPIE - The Int'l Soc. for Optical Eng., Appl. of Digital Image Proc.*, volume XVII, pages 773-778, July 1994.

[LLS01] L. Lucchese and S.K. Mitra. "Color image segmentation: A stateof-the-art survey". *Proc. of the Indian National Science Academy (INSA-A),* 67, A(2):207-221, March 2001.

[LLS98] L. Lucchese and S.K. Mitra. "An algorithm for unsupervised color image segmentation". In *Proc. of 1998 IEEE 2nd Workshop on Multimedia Signal Processing*, pages 33-38, December 1998.

[LLS99] L. Lucchese and S.K. Mitra. "Advances in color image segmentation". In *Proc. Global Telecommunications Conference Globecom*, pages 2038-2044, December 1999.

[LSM97] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic Watershed Segmentation of Randomly Textured Color Images," *IEEE Trans. on Image Processing*, Vol. IP-6, No.11, pp. 1530-1544, Nov. 1997.

[LSM98] L. Shafarenko, M. Petrou, and J. Kittler. "Histogram-based segmentation in a perceptually uniform color space". *IEEE Transaction on Image Processing*, IP-7(9):1354-1358, September 1998.

[LYF94] L. Yang, and F. Algregtsen, "Fast computation of invariant geometric moments: A new method giving correct results," *Proc. IEEE Int. Conf. on Image Processing,* 1994.

[MAF02] M.A.F. Figueiredo and A.K. Jain. "Unsupervised learning of finite mixture models". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):381-396, March 2002.

[MCA97] M. Chapron. "A chromatic contour detector based on abrupt change techniques". In *Proc. of Int. Conf. on Image Processing, ICIP'97*, volume III, pages 18-21, October 1997.

[MCH97] M. Celenk, "Hierarchical Color Clustering for Segmentation of Textured Images," *Proc. of the 29th Southeastern Symposium on System Theory*, Cookeville, TN, 9-11 Mar. 1997, pp. 483-487.

[MCM98] M. Celenk and M.U. de Haag. "Optimal thresholding for color images". In *Proc. of the SPIE - The Int'l Soc. for Optical Eng., Nonlinear Image Processing*, volume IX, pages 250-259, Jan 1998.

[MDE97] M. Das, E. M. Riseman, and B. Draper. Focus: Searching for multi-colored objects in a diverse image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition '97*, pages 756–761, June 1997.

[MFH95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system." *IEEE Computer*, Vol.28, No.9, pp. 23-32, Sept. 1995.

[MHC04] M.H.C. Law, M.A.F. Figueiredo, and A.K. Jain. "Simultaneous feature selection and clustering using mixture models". *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 26(9):1154-1166, September 2004.

[MHG94] M. H. Gross, R. Koch, L. Lippert, and A. Dreger. Multiscale image texture analysis in wavelet spaces. In *Proc. IEEE International Conference on Image Processing*, 1994.

[MIA89] M. Ioka, "A method of defining the similarity of images on the basis of color information,"*Technical Report RT-00JAA000*, IBM Tokyo Research Laboratory, Tokyo, Japan, Nov. 1989.

[MJS91] M. J. Swain, and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp.11-32, 1991.

[MKA87] M. Kass, A. Witkin, and D. Terzopoulos. "Snakes: Active contour models". *Int. J. of Computer Vision*, 1:321-331, 1987.

[MKH62] M. K. Hu. Visual pattern recognition by moment invariants, computer methods in image analysis. *IRE Transactions on Information Theory,* 8, 1962.

[MKH77] M. K. Hu, "Visual pattern recognition by moment invariants," in J. K. Aggarwal, R. O. Duda, and A. Rosenfeld, *Computer Methods in Image Analysis*, IEEE computer Society, Los Angeles, CA, 1977.

[MKM96] M. K. Mandal, T. Aboulnasr and S. Panchanathan, "Image Indexing Using Moments and Wavelets". *IEEE Transaction on Consumer Electronics*, vol. 42, No 3, pp. 557-565, 1996.

[MLK97] Michael Lew, Kim Lempinen, and Nies Huijmans. Webcrawling using sketches. In *Proceedings of the 2nd International Conference on Visual Information Systems, San Diego, December '97*, pages 77–84, 1997.

[MMC94] M.M. Chang, I. Sezan, and M. Tekalp. "Adaptive Bayesian segmentation of color images". *Journal of Electronic Imaging*, 3(4):404-414, October 1994.

[MOY97] Michael Ortega,Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang. Supporting similarity queries in MARS. In *Proceedings of the 5th ACM International Multimedia Conference, Seattle, Washington, 8-14 Nov. '97*, pages 403–413, 1997.

[MSM95] M. Stricker, and M. Orengo, "Similarity of color images", *SPIE Storage and Retrieval for Image and Video Databases III*, vol. 2185, pp.381-392, Feb. 1995.

[MSM96] M. Stricker, and M. Orengo, "Color indexing with weak spatial constraint," *Proc. SPIE Conf. on Visual Communications,* 1996.

[MSR99] M. Sammouda, R. Sammouda, N. Niki, and K. Mukai. "Segmentation and analysis of liver cancer pathological color images based on artificial neural networks". In *Proc. 1999 International conference on Image Processing, ICIP 99*, volume 3, pages 392-396, October 1999.

[NBH90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R_-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 322–331, 1990.

[NBT90] N. Beckmann, *et al*, "The R*-tree: An efficient robust access method for points and rectangles," *ACM SIGMOD Int. Conf. on Management of Data*, Atlantic City, May 1990.

[NFF94] N. Funakubo. "Feature extraction of color texture using neural networks for region segmentation". *In Proc. 20th Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON '94*, volume 2, pages 852-856, September 1994.

[NHC98] N. H. C. Yung and H. S. Lai. "Segmentation of color images based on the gravitational clustering concept". *The Journal of SPIE Optical Engineering*, 37(3):989-1000, March 1998.

[NIK98] N. Ikonomakis, K.N. Plataniotis, and A.N. Venetsanopoulos, "Gray-scale and Colour Image Segmentation via Region Growing and Region Merging," *Canadian Journal of Electrical and Computer Engineering*, Vol. 23, No. 1-2, IEEE Canada, Jan.-Apr. 1998, pp. 43-47.

[NPP93] N.P. Pal and S.K. Pal. "A review on image segmentation techniques". *Pattern Recognition*, 26(9):1277-1294, 1993.

[NRP93] N.R. Pal and S.K. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, Vol. 26, No. 9, pp. 1277-1294, 1993.

[NSC79] N. S. Chang, and K. S. Fu, "A relational database system for images," *Technical Report TR-EE 79-82*, Purdue University, May 1979.

[NSC80] N. S. Chang, and K. S. Fu, "Query by pictorial example," *IEEE Trans. on Software Engineering*, Vol.6, No.6, pp. 519-524, Nov.1980.

[PBT66] P. Brodatz, "Textures: A photographic album for artists & designers," Dover, NY, 1966.

[PCB97] P. Colantoni and B. Laget, "Color Image Segmentation Using Region Adjacency Graphs," *Proc. of the 6th Int'l Conf. on Image Processing and Its Applications*, Dublin, Ireland, 14-17 July 1997, Vol. 2, pp. 698-702

[PCD97] P. Campadelli, D. Medici, and R. Schettini. "Color image segmentation using Hopfield networks". *Image and Vision Computing Image*, 15(3):161-166, March 1997.

[PFF98] P.F. Felzenszwalb and D.P. Huttenlocher. "Image segmentation using local variation". In *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, pages 98-104, 1998.

[PHC82] P. Heckbert, "Color Image Quantization for Frame Buffer Display," *Computer Graphics*, vol. 16, no. 3, pp. 297–307, 1982.

[PJB81] P. J. Burt, T.-H. Hong, and A. Rosenfeld, "Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation," *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. SMC-11, No. 12, pp. 802-809, Dec. 1981.

[PJD04] P. J. Dutta, D. Bhattacharryya, J. Kalita, and M. Dutta. "Spatial Color Indexing Using Clustering Technique", *The 8th World Multi-Conference on Systemics, Cybernatics and Informatics (SCI' 2004)*. Vol VI: Image, Acoustics, Signal Processing & Optical Systems, Technologies and Applications, Orlando, Florida, pp. 216-221, July 2004.

[PJD06] P. J. Dutta, D Bhattacharyya, J Kalita and M Dutta, "Clustering Approach to Content Based Image Retrieval," Geometric Modeling and Imaging-New Trends (GMAI06), London, 2006.

[POF98] P. O. Fjaellstroem. "Algorithms for graph partitioning: A survey". *Linköoping Electronic Articles in Computer and Information Science*, 3(10), 1998.

[RBO00] R. Brunelli and O. Mich. Image retrieval by examples. *IEEE Transactions on Multimedia*, 2(JAA00):164–171, September 2000.

[RCG92] R.C. Gonzales and R.C. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.

[RJA95] R. Jain, A. Pentland, and D. Petkovic, *Workshop Report: NSF-ARPA Workshop on Visual Information Management Systems*, Cambridge, Mass, USA, June 1995.

[RJP92] R. Jain, *Proc. US NSF Workshop Visual Information Management Systems*, 1992.

[RJR95] R. Jain, R. Kasturi, and B.G. Schunck, "Machine Vision, McGraw-Hill, Inc.", New York, NY, 1995.

[RKS99] R. K. Srihari, Z. F. Zhang, and A. Rao. "Image background search: Combining objects detection techniques with content-based image retrieval (cbir) systems." *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '99), in conjunction with CVPR '99,* June 1999.

[RMH73] Robert M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Trans. on Sys. Man. and Cyb.*, SMC-3(6), 1973.

[RMH85] R.M. Haralick and L.G. Shapiro. "Survey on image segmentation techniques". *Computer Vision, Graphics and Image Processing*, Vol. 29, No. 1, pp. 100-132, Jan. 1985.

[RMK98] R. Mukundan and K. R. Ramakrishan. *Moment Functions in Image Analysis.* World Scientific, 1998.

[RSM94] R. Schettini and M. Suardi, "A Low-level Segmentation Procedure for Color Images," *Proc. of the 7th European Signal Processing Conference (EUSIPCO-94),* Lausanne, Switzerland, 13-16 Sept. 1994, Vol. I, pp. 26-29.

[RUG97] R. Urquhart. "Graph theoretical clustering based on limited neighborhood sets". In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 731-737, 1997.

[SAS85] S.A. Shafer, "Using Color to Separate Reflection Components," *Color Research and Application*, Vol. 10, No. 4, pp. 210-218, 1985.

[SGD84] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pp. 721-741, Nov. 1984.

[SHP98] S.H. Park, I.D. Yun, and S.U. Lee. "Color image segmentation based on 3D clustering: Morphological approach". *Pattern Recognition*, 31(8):1061-1076, August 1998.

[SID96] Special Issue on Digital Libraries: Representation and Retrieval, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-18, No. 8, Aug. 1996.

[SII96] S. Itoh and I. Matsuda, "Segmentation of Colour Still Images Using Voronoi Diagrams," *Proc. of the 8th European Signal Processing Conference (EUSIPCO-96)*, Trieste, Italy, 10-13 Sept. 1996, Vol. III, pp. 1869-1872.

[SIS98] Special Issue on Segmentation, Description, and Retrieval of Video Content, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. CASVT-8, No. 5, Sept. 1998.

[SJH98] S. Ji and H.W. Park, "Image Segmentation of Color Image Based on Region Coherency," *Proc. of 1998 Int'l Conf. On Image Processing (ICIP'98)*, Chicago, IL, 4-7 Oct. 1998, Vol. I, pp. 80-83, Oct. 1998.

[SKC81] S. K. Chang, and T. L. Kunii, "Pictorial database systems," *IEEE Computer Magazine*, Vol. 14, No.11, pp.13-21, Nov.1981.

[SKC87] S. K. Chang, Q. Y. Shi, and C. Y. Yan, "Iconic indexing by 2-D strings," *IEEE Trans. on Pattern Anal. Machine Intell.*, Vol.9, No.3, pp. 413-428, May 1987.

[SKC88] S. K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, "An intelligent image database system,"*IEEE Trans. on Software Engineering*, Vol.14, No.5, pp. 681-688, May 1988.

[SKC92] S. K. Chang, and A. Hsu, "Image information systems: where do we go from here?" *IEEE Trans. on Knowledge and Data Engineering*, Vol.5, No.5, pp. 431-442, Oct.1992.

[SKE88] S. K. Chang, E. Jungert, and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2D string", *Technical Report*, University of Pittsburgh, 1988.

[SMK97] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Towards a multimedia world wide web information retrieval engine. In Sixth International WWW Conference, 7-11 April '97, Santa Clara, CA, USA, 1997. http://decweb.ethz.ch/WWW6/Technical/Paper00JAA00/PaperJAA00.html.

[SMK99] Sougata Mukherjea, Kyoji Hirata, and Yoshinori Hara. Amore: A world wide web image retrieval engine. The WWW Journal, 2(3):115–132, 1999. http://www.baltzer.nl/www/contents/1999/2-JAA00.html.

[SNK97] S.N. Krjukov, T.O. Semenkova, V.A. Pavlova, and B.I. Arnt. "Back propagation neural network for adaptive color image segmentation". In *Proc. SPIE Applications of Artificial Neural Networks in Image Processing II*, volume 3030, pages 70-74, March 1997.

[SRR95] S. Ray, R. H. Turi, and P. E. Tischer. "Clustering-based colour image segmentation: An evaluation study". In *Proc. of Digital Image Computing: Technology and Applications*, pages 86-92, December 1995.

[SSA95] S. Sclaroff, and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 6, pp. 545-561, June 1995.

[STB92] S. T. Bow, *Pattern Recognition and Image Preprocessing*, Marcel Dekker, Inc., New York, NY, 1992.

[SYL90] S. Y. Lee, and F. H. Hsu, "2D C-string: a new spatial knowledge representation for image database systems," *Pattern Recognition*, Vol. 23, pp 1077-1087, 1990.

[SYL92] S. Y. Lee, M.C. Yang, and J. W. Chen, "2D B-string: a spatial knowledge representation for image database system," *Proc. ICSC'92 Second Int. computer Sci. Conf.*, pp.609-615, 1992.

[SZL95] S.Z. Li. *Markov Random Field Modeling in Computer Vision.* Ed. Tosiyasu L. Kunii, Springer-Verlag, Berlin, 1995.

[TCC93] Tianhorng Chang and C. C. Jay Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transaction on Image Processing,* 2(4): 429-441, October 1993.

[TCP96] T. Carron and P. Lambert, "Integration of Linguistic Knowledge for Color Image Segmentation," *Proc. of the 8th European Signal Processing Conference (EUSIPCO-96)*, Trieste, Italy, 10-13 Sept. 1996, Vol. 3, pp. 1729-1732.

[TGA00] T. Gevers, and A.W.M.Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval," *IEEE Trans. on image processing*, Vol.9, No.1, pp102-119, 2000.

[TGA97] T. Gevers and A.W.M. Smeulders, "Combining Region Splitting and Edge Detection Through Guided Delaunay Image Subdivision," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 17-19 June 1997, pp. 1021-1026.

[TGA99] T. Gevers, and A. W. M. Smeulders, "Content-based image retrieval by viewpoint-invariant image indexing," *Image and Vision Computing*, Vol.17, No.7, pp.475-488, 1999.

[TGS98] T. Gevers, S. Ghebreab, and A.W.M. Smeulders. "Color invariant snakes". In *Proc. of the 9th British Machine Vision Conference*, volume 2, pages 578-588, September 1998.

[TGV94] T. Gevers and V. K. Kajcovski, "Image Segmentation By Directed Region Subdivision," *Proc. of the 12th Int'l Conf. on Pattern Recognition*, Jerusalem, Israel, 9-13 Oct. 1994, Vol. 1, pp. 342-346.

[TNP92] T.N. Pappas. "An adaptive clustering algorithm for image segmentation". *IEEE Trans. on Signal Processing*, 40(4):901-913, 1992.

[TPM96] T. P. Minka, and R. W. Picard, "Interactive learning using a 'society of models', " *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 447-452, 1996.

[TUM94] T. Uchiyama and M.A. Arbib. "Color image segmentation using competitive learning". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(12):1197-1206, December 1994.

[TVA93] T. Vlachos and A.G. Constantinides. "Graph-theoretical approach to colour picture segmentation and contour classification". In *IEE Proceedings*, Part I, volume 140, pages 36-45, February 1993.

[TYI98] T. Yamazaki, "Introduction of E M algorithm into color image segmentation". In *Proc. Int. Conf. on Image Processing, ICIPS'98*, pp. 368-371, August 1998.

[TZR96] Tian, Z., Raghu, R., Miron, L. BIRCH: "An efficient data clustering method for very large databases". *SIGMOD _96*, Montreal, Canada, pp. 103–114, 1996.

[UEG97] U. Elsner. "Graph partitioning: A survey". *Technical Report 393/97-27, Numerische Simulation auf Massiv Parallelen Rechnern, Technische Universitäat Chemnitz*, December 1997.

[VAC97] V.A. Christopoulos, P. De Muynck, and J. Cornelis, "Colour Image Segmentation for Low Bit Rate Segmented Image Coding," *Proc. of the 13th Int'l Conf. on Digital Signal Processing, Santorini*, Greece, 2-4 July 1997, Vol. 2, pp. 861-864.

[VEO95] Virginia E. Ogle and Michael Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, September 1995.

[VLB96] V. Lozano and B. Laget, "Fractional Pyramids for Color Image Segmentation," *Proc. of Southwest Symposium on Image Analysis and Interpretation*, San Antonio, TX, 8-9 April 1996, pp. 13-17.

[VNG95] V. N. Gudivada, and V. V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity," *ACM Trans. on Information Systems*, Vol. 13, No. 2, pp. 115-144, April 1995.

[VRL98] V. Rehrmann and L. Priese, "Fast and Robust Segmentation of Natural Color Scenes," *Proc. of the 3rd Asian Conf. on Computer Vision (ACCV'98)*, Hong Kong, 8-10 Jan 1998, Vol. 1, pp. 598-606.

[WIG90] W. I. Grosky, and R. Mehrotra, "Index based object recognition in pictorial data management," *CVGIP*, Vol. 52, No. 3, pp. 416-436, 1990.

[WJK95] W. J. Krzanowski, *Recent Advances in Descriptive Multivariate Analysis*, Chapter 2, Oxford science publications, 1995.

[WNQ93] W. Niblack et al., "Querying images by content, using color, texture, and shape," *SPIE Conference on Storage and Retrieval for Image and Video Database*, Vol. 1908, pp.173-187, April 1993.

[WNR93] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The qbic project: Quering images by content using color, texture, and shape. In *Poceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases, 2-3 February '93, San Jose, CA*, pages 173–187, 1993.

[WWC97] W. Wang, C. Sun, and H. Chao. "Color image segmentation and understanding through connected components". In *Proc. of 1997 IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 2, pages 1089-1093, October 1997.

[WYB97] W. Y. Ma, and B. S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 744-749, Puerto Rico, June 1997.

[WYJ97] W.Y. Ma and B. Manjunath. "NeTra: A Toolbox for Navigating Large Image Databases", *Proc. IEEE Int'l Conf. Image Processing*, pp. 568-571, 1997.

[WYM97] W. Y. Ma. *NETRA: A Toolbox for Navigating Large Image Databases*. PhD thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara, June 1997.

[WYM99] W. Y. Ma, and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *Multimedia Systems*, Vol.7, No.3, pp.:184-198, 1999.

[YDB01] Y. Deng and B. S. Manjunath. "Unsupervised Segmentation of color-texture regions in images and video", *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI'01)*, 2001.

[YDB99] Y. Deng, B.S. Manjunath, and H. Shin, "Color Image Segmentation," to appear in *Proc. of 1999 Int'l Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, CO, 23-25 June, 1999.

[YGH94] Y. Gong, H. J. Zhang, and T. C. Chua, "An image database system with content capturing and fast image indexing abilities", *Proc. IEEE International Conference on Multimedia Computing and Systems*, Boston, pp.121-130, 14-19 May 1994.

[YKI98] Y. Kanai, "Image Segmentation Using Intensity and Color Information," *Proc. of the SPIE - The International Society for Optical Engineering Proc. SPIE - Visual Communications and Image Processing '98*, San Jose, CA, 28-30 Jan. 1998, pp. 709-720.

[YRA96] Young Rui, Alfred C. She and Thomas S. Huang. Modified fourier descriptors for shape representation – a practical approach. In *Proc. of First International Workshop on Image Database and Multi Media Search,* 1996.

[YRA97] Y. Rui, *et al*, "A relevance feedback architecture in content-based multimedia information retrieval systems," *Proc of IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997.

[YRT97] Y. Rui, T.S.Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," *Proceedings of International Conference on Image Processing*, Vol.2, pp. 815 -818, 1997.

[YRT98] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology*, 1998.

[YRT99] Y. Rui, T. S. Huang, and S. F. Chang, "Image retrieval: current techniques, promising directions and open issues, " *Journal of Visual Communication and Image Representation*, Vol.10, pp. 39-62, 1999.

[YXE97] Y. Xu and E.C Uberbacher. "2D image segmentation using minimum spanning trees". *Image and Vision Computing*, 15(1):47-57, 1997.

[ZLD00] Z. Lei, D. Keren and D. B. Cooper. Computationally fast Bayesian recognition of complex objects based on mutual algebraic invariants. In *Proc. IEEE Int. Conf. on Image Proc.*

[ZNL98] Z. N. Li, O. R. Za¨ıane, and B. Yan. "C-bird: Content-based image retrieval from image repositories using chromaticity and recognition kernel". Technical Report CMPT98-03, School of Computing Science, Simon Fraser University, Canada, February 1998. ftp://ftp.fas.sfu.ca/pub/cs/TR/1998/CMPT1998-03.ps.gz.

[ZNL99] Z.N. Li, O. R. Za¨ıane, and Z. Tauber. Illumination invariance and object model in content-based image and video retrieval. Journal of Visual Communication and Image Representation, 10(3):219–244, September 1999. http://www.cs.sfu.ca/cs/undergrad/CM/CMPT3065/CBS/CBIRD.pdf.

[ZPF98] Zoran Peˇcenovi´c. Finding rainbows on the internet. Technical report, Section of Communication Systems, ´ Ecole Polytechnique F´ed´eral de Lausanne, 1998.

[ZPI97] Zoran Peˇcenovi´c. Image retrieval using latent semantic indexing. Master's thesis, Department of Electrical Engineering, ´ Ecole Polytechnique F´ed´eral de Lausanne, 1997.

[ZWR93] Z. Wu and R. Leahy. "An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(11):1101-1113, November 1993.

[ZYW00] Zhao, Y.W., Wang, W. L. "A new clustering algorithm for color quantization and its applications", *J. Computer Aided Des. Comput. Graph.* 12 (5), 340–343, 2000.