Image Retrieval From Repository using Overlapping Approach

A Dissertation Submitted In Partial Fulfilment of the Requirement for the Award of the Degree Of

Master of Engineering

In

Computer Technology and Applications

SUBMITTED BY

ASHEESH KUMAR

(University Roll No. 8543)

(College Roll No. 03/CTA/09)

UNDER THE ESTEEMED GUIDANCE OF

MR. VINOD KUMAR

(ASSISTANT PROFESSOR, COMPUTER ENGINEERING)

DEPARTMENT OF COMPUTER ENGINEERING

DELHI COLLEGE OF ENGINEERING

UNIVERSITY OF DELHI, DELHI

(2009-2011)

CERTIFICATE

This is to certify that the major report entitled as "**Image Retrieval From repository using Overlapping Approach**" is being submitted by **ASHEESH KUMAR** University Roll No. 8543, in partial fulfilment for the award of "Master of Engineering Degree in Computer Engineering" in Delhi College of Engineering, Delhi University, Delhi. He has worked under my guidance and supervision and has fulfilled the requirements for the submission of this report, which has reached the requisite standard.

I wish him success in all his endeavours.

MR. VINOD KUMAR Date: ………........

Assistant Professor and Project Guide
Department of Computer Engineering
Delhi College of Engineering,
University of Delhi, India

ACKNOWLEDGEMENT

It is my distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor Mr. Vinod Kumar, Department of Computer Engineering, Delhi College of Engineering, for his invaluable guidance, encouragement and patient reviews. His continuous inspiration only has made me complete this dissertation. Without his help and guidance, this dissertation would have been impossible. He remained a pillar of help throughout the project.

I would also like to thank Dr. Daya Gupta, HOD, Department of Computer Engineering, for her valuable suggestions.

I would also like to take this opportunity to present my sincere regards to all the teachers in the Department of Computer Engineering for their support and encouragement.

I would also like to thank my classmates for their unconditional support and motivation during this work.

Lastly, I would like to thank Almighty GOD for his countless blessings.

ASHEESH KUMAR
Master in Engineering
(Computer Technology & Application)
College Roll No. - 03/CTA/09
University Roll No. - 8543
Department of Computer Engineering
Delhi College of Engineering, Delhi-110042

ABSTRACT

Content Based Image Retrieval is an interesting and most emerging field in the area of 'Image Search', finding similar images for the given query image from the image database. Current approaches include the use of color, texture and shape information. Considering these features in individual, most of the retrievals are poor in results and sometimes we are getting some non relevant images for the given query image.

So, this dissertation proposes a method in which we first pre-process the original image based on different factors like edge, saturation, brightness, lightness etc, and then we will perform some similarity test on each set and then find image that is overlapping maximally at particular priority value, by taking all sets together, than display result.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Valuable information can be hidden in images. Plenty of knowledge can be hidden in these image data, it is since 1970th people devoted themselves into image retrieval research The need for image mining is high in view of the fast growing amounts of image data [1].

Due to the digitization of data and advances in technology, it has become extremely easy to obtain and store large quantities of data, particularly Multimedia data. Presently, tools for mining images are few and require human intervention. Feature selection and extraction is the pre-processing step of Image Mining. Obviously this is a critical step in the entire scenario of Image Mining [10]

Most conventional image databases are text based .As a result, image retrieval is based on keyword searching. Text annotated images are simple and easy to manipulate. However, there are two major problems with this method. First, creating keywords for large number of images is time consuming. Moreover, the keywords are inherently subjective and not unique.

Due to these disadvantages, Image retrieval using multiple image features like saturation, brightness, edge etc is an efficient method. Image retrieval based on image property becomes more desirable for developing large volume image retrieval applications.

1.2 Problem Definition

In image mining, color has been extensively used in image matching and retrieval. But color retrieval alone does not give good results. In this thesis we consider the property of the image along with the color to improve the efficiency. Different image parameters are combined in our retrieval system to compute the similar images for the given query image.

1.3 Scope

The scope of this research is color and features of images to improve the efficiency of the Image Mining system. We have computed the image parameter described in Chapter 4 on dataset and implemented color and edge, brightness, hue, saturation etc approach on that. Using Overlapping approach, we retrieved images from the repository for the given query image.

1.4 Organization of Thesis:

The remainder of thesis is organized as

In **Chapter 2** deals with Image mining, content based image retrieval and prior work done in this field. It also describes some current techniques used in image retrieval and some methods to calculate similarity and distances measurements. It also describes some query based methods and field of application.

In **Chapter 3** we cover the introduction of human perception theory, color, color representation attribute, color model (RGB, HSB, CMYK etc.). In this section we also describe image, type of image (binary image, gray scale image, RGB image).

Chapter 4 deals with proposed method of image retrieval.

Chapter 5 covers implementation details of Development Environment Netbeans platform with different classes which I have implemented in JAVA, it shows various java package JAI and other which are used in this thesis.

Chapter 6 covers the conclusion and future work done by us. We finally culminate thesis showing different references including research papers websites and books that I have gone through during my project.

Chapter 2

Overview of Image Retrieval Methods

2.1 Image Mining

As computer technologies become more ubiquitous, besides numerical and categorical data, various digitalized images, sounds, voices, and videos have become part of daily life. Plenty of knowledge can be hidden in these data, it is since 1970th people devoted themselves into image retrieval research, then text based image retrieval technology and context web retrieval technology were proposed, which in a certain extent solved some image retrieval and resource discovery problems. However, people are not satisfied with only being able to access information, because through image retrieval people can only find out the relative information they want, they can't dig out valuable knowledge hidden in large sets of image data.

Image mining concerns the extraction of implicit knowledge, image data relationship, or other patterns not explicitly stored in the images. It is more than just an extension of data mining to image domain. Image mining is an interdisciplinary which draws upon expertise in computer vision, image understanding, data mining, machine learning, database, and artificial intelligence.

2.2 Research issues in image mining [2]

Image mining deals with the extraction of image patterns from a large collection of images. Image mining is different from low-level computer vision and image processing techniques because the focus of image mining is in extraction of patterns from large collection of images, whereas the focus of computer vision and image processing techniques is in understanding and/or extracting specific features from a single image. In image mining, the goal is the discovery of image patterns that are significant in a given collection of images. Perhaps, the most common misconception of image mining is that image mining is nothing more than just applying existing data mining algorithms on images. This is certainly not true because there are important differences between relational databases versus image databases.

2.2.1 Absolute versus relative values.

In relational databases, the data values are semantically meaningful. For example, age is 35 is well understood. But in image databases, the data values themselves may not be significant unless the context supports them. For example, a grey scale value of 46 could appear darker than a grey scale value of 87 if the surrounding context pixels values are all very bright.

2.2.2 Spatial information (Independent versus dependent position)

Another important difference between relational databases and image databases is that the implicit spatial information is critical for interpretation of image contents but there is no such requirement in relational databases. As a result, To overcome this problem by extracting position-independent features from images first before attempting to mine useful patterns from the images.

2.2.3 Unique versus multiple interpretations.

A third important difference deals with image characteristics of having multiple interpretations for the same visual patterns. The traditional data mining algorithm of associating a pattern to a class (interpretation) will not work well here. A new class of discovery algorithms is needed to cater to the special needs in mining useful patterns from images.

2.3 Process of Image Mining [1]

Observing from some of the existing image mining systems, overall process can be divided into the following parts:

2.3.1. Data preprocess

There exist a lot of dirty and noisy data in large image databases, for instance, images that are extremely unclear and images that are already breached. Those data often cause chaos in mining process and give birth to bad mining results, so it is necessary to preprocess data, clean up the noisy, broken, dirty data.

2.3.2. Extracting multi-dimensional feature vectors

Using image processing technologies such as image segmentation, picking up the edge to extract task-related feature vectors, form multi-dimensional feature vectors.

2.3.3. Mining on vectors and acquire high-level knowledge

Various methods such as object recognition, image indexing and retrieval, image classification and clustering, neural network are used on feature vectors for mining and acquiring hidden and valuable high-level knowledge, then evaluate and explain that knowledge.

2.4 Content based image retrieval

In 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.

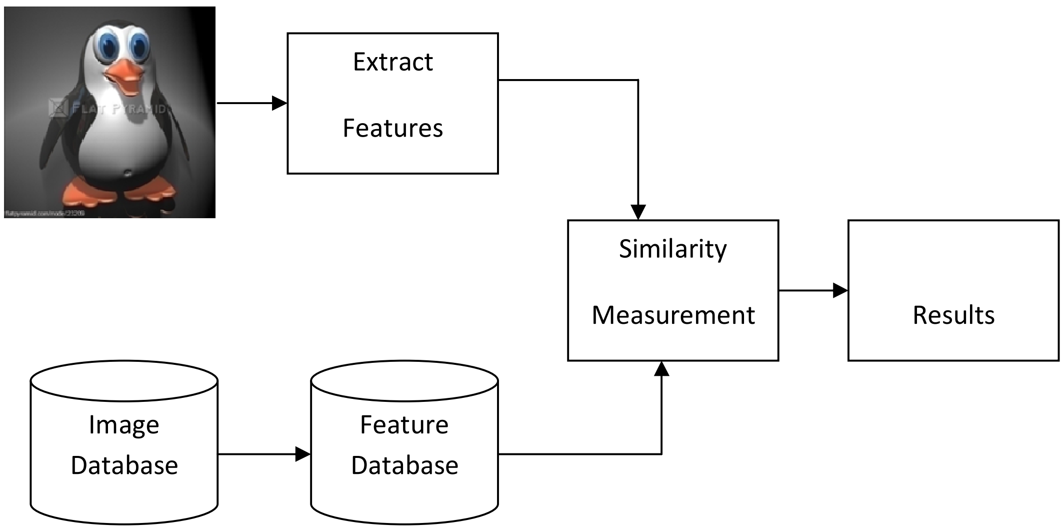


figure 2.1: Content Based Image Retrieval System

2.5 Prior-Work

Early work on image retrieval can be traced back to the late 1970s. In 1979, a conference on Database Techniques for Pictorial Applications [16] was held in Florence. Since then, the application potential of image database management techniques has attracted the attention of researchers. 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.

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 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 [11]. 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.

2.6 Content Comparison Techniques

There are some common methods for extracting content from images so that they can be easily compared. The methods outlined are not specific to any particular application domain.

2.6.1 Color Retrieval

Color is the most extensively used visual content for image retrieval. 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. Retrieving images based on color similarity is achieved by computing a color histogram for each image that identifies the proportion of pixels within an image holding specific values. The first order (mean), the second order (variance) and the third order (skewness) color moments have been proved to be efficient and effective in representing color distributions of images.

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. Another method called color correlogram expresses how the spatial correlation of pairs of colors changes with distance.

2.6.2 Texture Retrieval

Texture is a widely used and intuitively obvious but has no precise definition due to its wide variability. Visual texture in most cases is defined as a repetitive arrangement of some basic pattern. This repetition may not be random. However, a texture pattern normally has some degree of randomness due to randomness in basic pattern as well as due to randomness in the repetition of basic pattern. To quantify texture, this randomness is measured by some means over a small rectangular region called window. Thus, texture in an image turns out to be a local property and depends on the shape and size of the window .Identifying a patch in an image as having uniform texture or discriminating different visual textures obeys the law of similarity. In this case, the texture property is used to produce similarity groupings.

Basically, texture representation methods can be classified into two categories: structural and statistical. Structural methods, including morphological operator and adjacency graph, describe texture by identifying structural primitives and their placement rules. They tend to be most effective when applied to textures that are very regular.

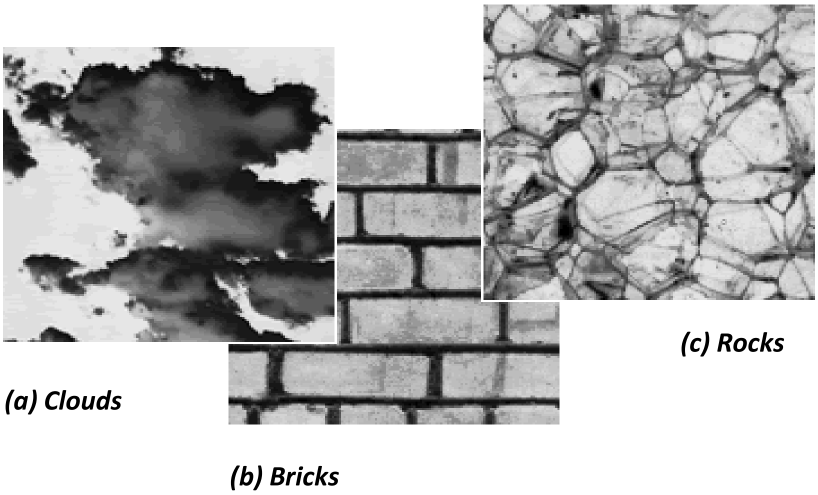


figure2.2 Texture based image retrieval

2.6.3 Shape Retrieval

Shape may be defined as the characteristic surface configuration of an object; an outline or contour. It permits an object to be distinguished from its surroundings by its outline Shape representations can be generally divided into two categories:

- Boundary-based, and
- Region-based.

Boundary-based shape representation only uses the outer boundary of the shape. This is done by describing the considered region using its external characteristics; i.e., the pixels along the object boundary. Region-based shape representation uses the entire shape region by describing the considered region using its internal characteristics; i.e., the pixels contained in that region

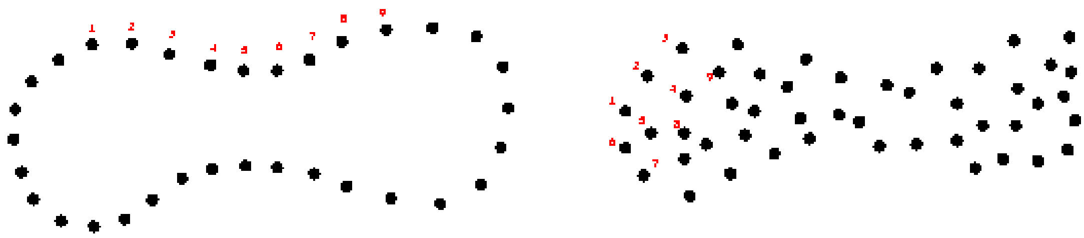


figure 2.3: Boundary-based & Region-based shape representation

2.7 Similarity/Distance Measures [8]

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.

2.7.1 Euclidean Distance

In Euclidean distance based color image segmentation technique, the RGB color model is considered. In the RGB color model, each color appears in its primary spectral components of red, green and blue. Each RGB color pixel is a triplet of values namely Red, Green and Blue. Segmentation provides better results in RGB color model when compared to other color models.

Segmentation in color domain is based on similarity detection rather than discontinuity based. Similarity based detection directly groups the similar pixels.

The algorithm involves the following steps.

1) The first step involves take image frames.

2) On each frame the following operations are performed.

a) Select an estimate of the average color that is to be segmented.

b) Euclidean distance is chosen as the measuring parameter The Euclidean distance between the image pixel 'I' and 'J' is

$$D(I,J)= \sqrt{[(I_R - J_R)^2 + (I_G - J_G)^2 + (I_B - J_B)^2]}$$

Any image pixel 'J' is said to be similar to 'I' if the Euclidean distance between them is less than a specified threshold D_0. Choosing D_0 is dependent on the defect that is to be classified. For all the thermographs of same defect, the value of D_0 is same, hence making this algorithm image independent and parameter independent.

2.7.2 Minkowski-Form Distance

If each dimension or image features 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., Let D(Q,T) be the distance measure between the query image Q and the image T in the database. .

The Minkowski-form distance is defined based on the Lp norm

$$d_p(\mathbf{Q},\mathbf{T}) = (\sum_{i=0}^{N-1} (Q_i - T_i)^p)^{\frac{1}{p}}$$

where Q = {Q0, Q1, … QN-1} and T = {T0, T1, …, TN-1} are the query and target feature vectors respectively. When p=1, 2,…..∞, d (Q, T) is the L 1, L 2 (also called Euclidean distance and L∞ distance respectively. Minkowski-form distance is the not widely used metric for image retrieval.

When p = 1, d1(Q,T) is the city block distance or Manhattan distance (L1)

$$d_1(\mathbf{Q},\mathbf{T}) = \sum_{i=0}^{N-1} | Q_i - T_i |$$

When p = 2, d2(Q, T) is the Euclidean distance (L2)

$$d_2(\mathbf{Q},\mathbf{T}) = (\sum_{i=0}^{N-1} (Q_i - T_i)^2)^{\frac{1}{2}}$$

When p→∞, we get L∞,

$$L_\infty(\mathbf{Q},\mathbf{T}) = \max_{0 \le i < N}\{| Q_i - T_i |\}$$

Used Euclidean distance for color and shape feature, and *L1* distance for texture feature

2.7.3 Histogram intersection

The Histogram intersection can be taken as a special case of *L1* distance, which is Used to compute the similarity between color images. Their objective was to find known objects within images using color histograms. It is able to handle partial matches when the object (with feature Q) size is less than the image (with feature T) size. The original definition of histogram distance is given as The Intersection of the two histograms of *I* and *J* is defined as:

$$S(I,J) = \frac{\sum_{i=1}^{N} \min(f_i(I), f_i(J))}{\sum_{i=1}^{N} f_i(J)}$$

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

2.7.4 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:

$$D(I,J) = \sqrt{(\mathbf{F}_I - \mathbf{F}_J)^T \mathbf{A}(\mathbf{F}_I - \mathbf{F}_J)}$$

Where *A*= [*aij*] is a similarity matrix, and *aij* denotes the similarity between bin *i* and *j*.

$\mathbf{F_I}$ and $\mathbf{F_J}$ are vectors that list all the entries in *fi*(*I*) and *fi*(*J*). Quadratic form distance has been used in many retrieval systems [21] 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.

2.7.5 Cosine Distance

The cosine distance computes the difference in direction, irrespective of vector lengths. The distance is given by the angle between the two vectors. By the rule of dot product

$Q.T= Q`\ T= |Q| . |T| \cos \Theta$

$d_{cos}(Q,T)=1- \cos \Theta = 1- (Q`\ T) / (|Q| . |T|)$

2.7.6 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:

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

Where C^{-1} = 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.

2.8 Query techniques

Different implementations of CBIR make use of different types of user queries.

Query by example

In this technique a user provides example image to the CBIR system for which it wants to search similar images .In this case search is based on some common attributes that provided image sharing with the searched image.

There are many ways by which a user can provide query image:

- An image already in the database can be supplied by the user or it can chose from a random set.
- The user can draw a rough approximation of the image they are looking for, for example with blobs of color or general shapes. For example 30% red ,40% green etc.

This query technique removes the difficulties that can arise when trying to describe images with words.

Semantic retrieval

The ideal CBIR system from a user perspective would involve what is referred to a semantic retrieval, where the user makes a request like "find pictures of lion" or even "find pictures of Abraham Lincoln". This type of open-ended task of searching is very difficult for computers to perform - pictures of Lincoln may not always be facing the camera or in the same pose. Current CBIR systems therefore generally make use of lower-level features like texture, color, and shape, although some systems take advantage of very common higher-level features like faces). Not every CBIR system is generic.

2.9 Field of Application

Content based has the following field of applications:

- Intellectual property
- Fashion and interior design
- Medical diagnosis
- Crime prevention
- The military
- Architectural and engineering design
- Journalism and advertising
- Home entertainment
- Web searching
- Education and training
- Geographical information and remote sensing systems

Chapter 3

Color and Image

3.1 Human visual perception

The eye is made up of several parts. Light enters the eye through the *pupil*, is focused by the *lens*, and passes through to the *retina* on the back of the eye. The retina is covered with sensitive nerve cells called *photoreceptors*; these cells receive the light and pass the stimulus onto the brain. The center of the retina is called the *fovea*; this is also the center of your vision.

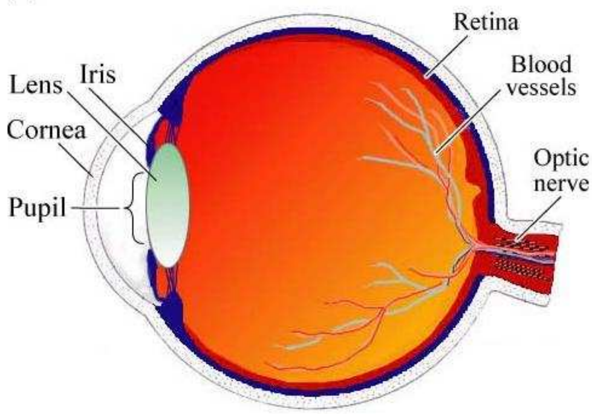


figure 3.1 Cross section through Human Eye

The photoreceptors are divided into two groups: *rods* and *cones*. There are about 75 to 150 million rods, but only about 5 to 8 million cones. The rods provide the ability to detect brightness, and the cones allow you to perceive color. The rods are more *light*-sensitive than the cones, but are not able to distinguish between colors. Most of the cones are in the center of your eye, near the fovea; while the rods are absent from the immediate area of the fovea, but extend on both sides of the back of the retina.

This is why you can usually only distinguish brightness levels at the edges of your vision.[12]

There are three types of cones; each type is sensitive to different wavelengths of light. The cones for long wavelengths perceive mainly the yellows and reds; medium wavelength cones receive mainly the yellows and greens; and the short wavelength cones are strongly sensitive to the blue and indigo colors.

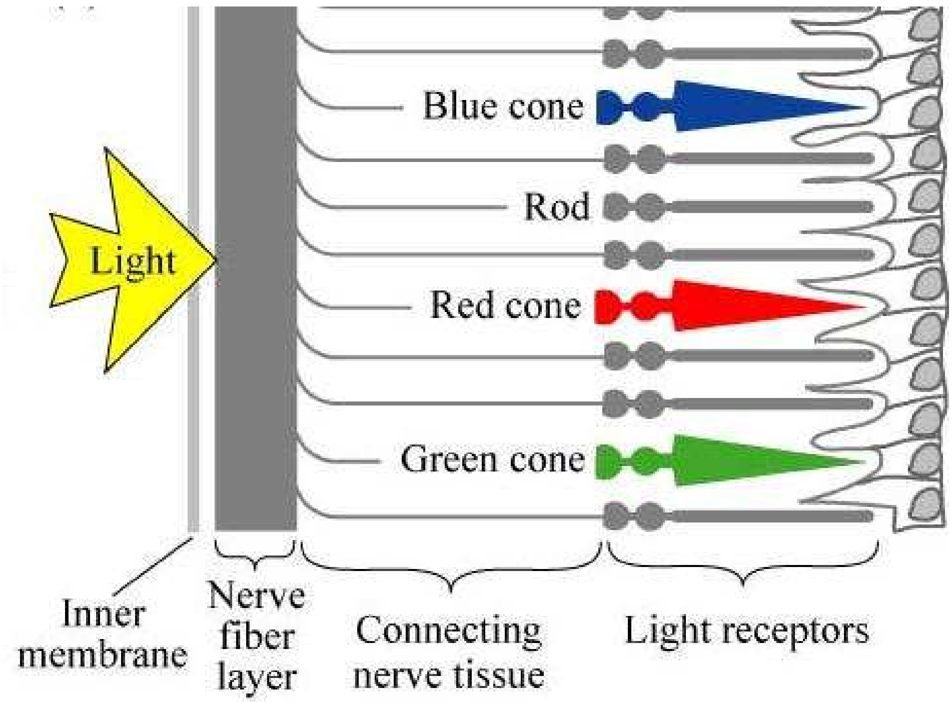


figure 3.2 Semantic view of the retina including rod and Cone light receptors

These three types of cones are not uniformly distributed on the retina. There are about 3.5 million each long and medium wavelength cones and they are mostly in the middle of the retina. On the other hand, there are only about 1 million short wavelength cones; these are distributed over the retina but are more strongly concentrated on the sides of the retina. This means that it is easier to focus on red, yellow, or green objects than on blue ones. [12]

3.2 Color

Color is a sensation produced by the human eye and nervous system. It is related to light, but an understanding of the properties of light is not sufficient to understand color, and is especially not sufficient to understand the art of color reproduction. Overwhelming experimental evidence tells us that the perception of a color is related to the strength of three signals which are transmitted along the optic nerve to the brain. Finding that our sensation of color comes from nerve cells that send messages to the brain about:

- The brightness of color
- Greenness vs. redness
- Blueness vs. yellowness

Color originates in light. Sunlight, as we perceive it, is colorless. In reality, a rainbow is testimony to the fact that all the colors of the spectrum are present in white light. As illustrated in the diagram below, light goes from the source (the sun) to the object (the apple), and finally to the detector (the eye and brain).

1. All the" invisible" colors of sunlight shine on the apple.
2. The surface of a red apple absorbs all the colored light rays, except for those corresponding to red, and reflects this color to the human eye.
3. The eye receives the reflected red light and sends a message to the brain. The most technically accurate definition of color is: "Color is the visual effect that is caused by the spectral composition of the light emitted, transmitted, or reflected by objects."[12]

Light and Color

The human eye is sensitive to electromagnetic radiation with wavelengths between about 380 and 700 nanometers. This radiation is known as light. The visible spectrum is illustrated on the right. The eye has three classes of color-sensitive light receptors called cones, which respond roughly to red, blue and green light (around

650, 530 and 460 nm, respectively). A range of colors can be reproduced by one of two complimentary approaches: The visible light spectrum is the section of the electromagnetic radiation spectrum that is visible to the human eye. It ranges in wavelength from approximately 400 nm (4×10^{-7} m) to 700 nm (7×10^{-7} m). It is also known as the optical spectrum of light.

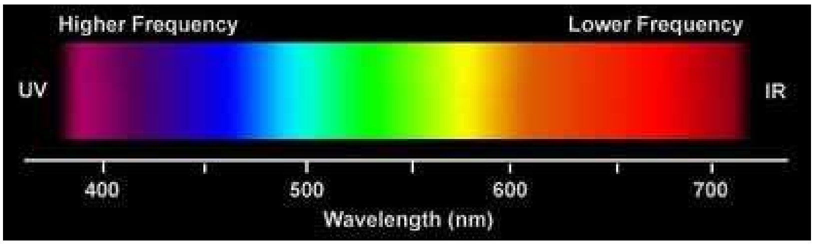


Figure 3.3 Visual color Spectrum

The wavelength (which is related to frequency and energy) of the light determines the perceived color. The ranges of these different colors are listed in the table below. Some sources vary these ranges pretty drastically, and the boundaries of them are somewhat approximate as they blend into each other. The edges of the visible light spectrum blend into the ultraviolet and infrared levels of radiation.

Most light that we interact with is in the form of *white light*, which contains many or these entire wavelength ranges within them. Shining white light through a prism causes the wavelengths to bend at slightly different angles due to optical refraction. The resulting light is, therefore, split across the visible color spectrum [9].

This is what causes a rainbow, with air born water particles acting as the refractive medium. The order of wavelengths (as shown to the right) is in order of wavelength, which can be remembered by the mnemonic "Roy G. Biv" for Red, Orange, Yellow, Green, Blue, Indigo (the blue/violet border), and Violet. You'll notice that in the image and table Cyan is also appears fairly distinctly, between green & blue.

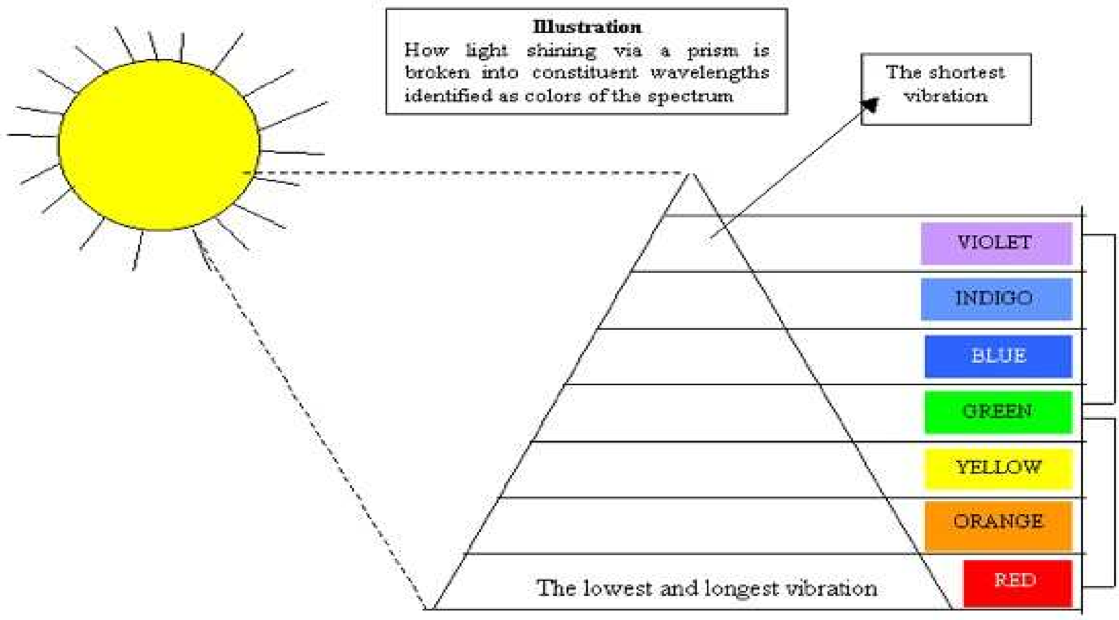


Figure 3.4 Color component of light

The Visible Light Spectrum Color Wavelength (nm)

Red 625 - 740

Orange 590 - 625

Yellow 565 - 590

Green 520 - 565

Cyan 500 - 520

Blue 435 - 500

Violet 380 - 435

By using special sources, refractors, and filters, you can get a narrow band of about

10 nm in wavelength that is considered *monochromatic* light. Lasers are special because they are the most consistent source of narrowly monochromatic light that we can achieve.

3.3 Color-making attributes [11]

3.3.1 Hue

The "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them". It represent dominant wavelength in the mixture of color.

3.3.2 Radiance

The total amount of light comes out from source. Is unit is watts.

3.3.3 Luminance (*Y*)

The amount of energy perceive by an observer, measured in candela per square metre (cd/m^2). Often the term *luminance* is used for the relative luminance, Y/Y_n, where Y_n is the luminance of the reference white point.

luminance can be calculated from linear RGB components:

$$Y = 0.2126 R + 0.7152 G + 0.0722 B$$

3.3.4 Luma (*Y′*)

The weighted sum of gamma-corrected R', G', and B' values, and used in YCbCr, for JPEG compression and video transmission.

3.3.5 Brightness

The "attribute of a visual sensation according to which an area appears to emit more or less light".

3.3.6 Lightness

The "brightness relative to the brightness of a similarly illuminated white". Brightness is a subjective measure.

3.3.7 Saturation

Saturation is the purity of the color and is the amount of pure color mixed with white color. It varies from white to pure color. It is measured in percent from 0 to 100. The higher the percentage, the more pure will be the color.

3.4 Color Models

A color model is a specification of a 3-D coordinate system and a subspace within that system where each color is represented by a single point. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called color space.

Each industry that uses color employs the most suitable color model. For example, the RGB color model is used in computer graphics, A computer may describe a color using the amounts of red, green and blue phosphor emission required to match a color.YUV or YCbCr are used in video systems, PhotoYCC* is used in Photo CD* production and so on.

A printing press may produce a specific color in terms of the reflectance and absorbance of cyan, magenta, yellow and black inks on the printing paper. A color is thus usually specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used. They do not tell us what the color is, that depends on what color space is being used. Transferring color information from one industry to another requires transformation from one set of values to another. Intel IPP provides a wide number of functions to convert different color spaces to RGB and vice versa.

3.4.1 Tristimulus color space

The eye seems to be composed of rods and cones. Rods respond only to the intensity of the light falling on them, and are more sensitive to low light levels than are cones. The cones are of three different types and respond differently to different wavelengths. What we perceive as color seems to depend on characteristics of *brightness*, *hue*, and *saturation*. We generally regard the basic colors as Red, Green, and Blue, and define other colors as a mix of these three, the amount of each basic color being specified by *tristimulus* values of X, Y, and Z, respectively. A color is then specified by its trichromatic coefficients:[12]

x_{red} = X/X+Y+Z ; x_{red}= amount of red light;

y_{green}= Y/ X+Y+Z ; y_{green}= amount of green light.

z_{blue} = Z/ X+Y+Z. ; z_{blue} = amount of blue light.

x + y + z = 1; Normalize Form

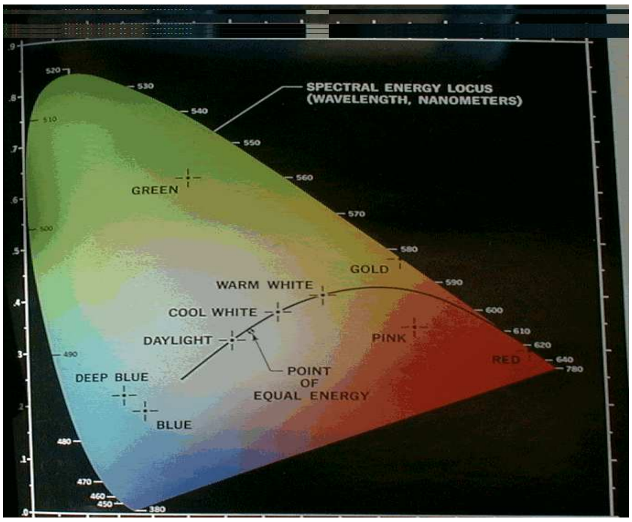


Figure 3.5 CIE chromaticity diagram

which gives color composition as a function of *x* (red) and *y* (green), with the amount of *z* being determined from **z = 1 - (x + y)**. Points around the border of the diagram are considered fully saturated, while the saturation goes to zero as one moves on a straight line from a boundary point to the equal energy point which represents white. A straight line joining any two points represents all the colors that are possible by combining those two colors in varying amounts, while the triangle enclosing three non-collinear points represents all the colors that can be produced by combinations of the three colors represented by the triangle corners [12].

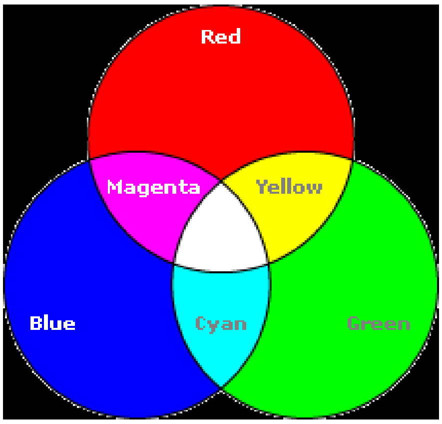
3.4.2 RGB Model

The RGB color model defines color using Red (R), Green (G) and Blue (B) light.

- Each color is measured with a value ranging from 0 to 255 where 0 is no light and 255 is maximum intensity. This is how much information can be stored in 1 Byte of computer memory (256 pieces).
- To define all three colors, you need 3 Bytes (or 24-bits) of information.
- The RGB color model is an Additive Color Model.
- . With RGB, mixing of red and green equally gives yellow, mixing of green and blue creates cyan and the mixing of red and blue creates magenta.
- Additive color uses transmitted light to display color.

- Computer Monitors and the human eye use RGB to determine color.
- Monitor can create millions of colors by combining different percentages of three primaries, red, green and blue.
- The combination of amounts of individual red, green, and blue light defines the resulting RGB color.
- When you add red light, blue light, and green light together and each component has a value of 255, then the resulting color is white. When the value of each component is 0, the resulting color is pure black.
- With the RGB additive model, computers can display up to 16.7 million colors.
- Image processing software like Photoshop you can see that these RGB colors are added with the help of numerical value, which is between 0 to 255.

The basic advantage of RGB model is; it is useful for full color editing because it has wide range of colors. But at the same time this model is said to device dependent. It means the way colors displayed on the screen depends on the hardware used to display it.



(a)

The RGB (**R**ed, **G**reen, **B**lue) color model is shown in the figure on the left. This is built on a cube with Cartesian coordinates. Each dimension of the cube represents a primary color. Each point within the cube represents a particular hue; the coordinates of that point show the contributions of each primary color toward the given color.

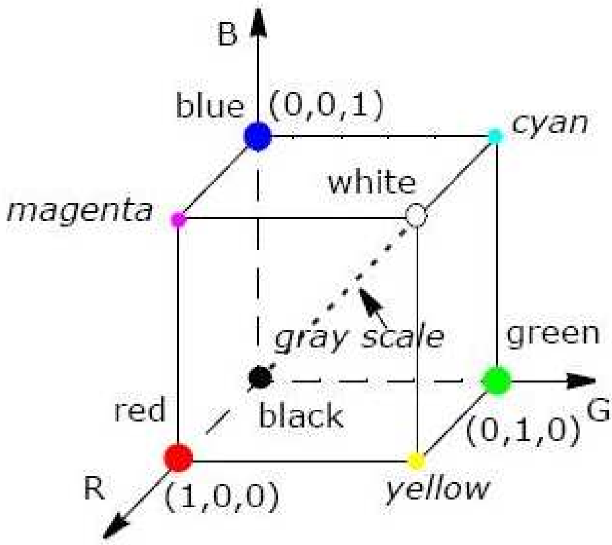


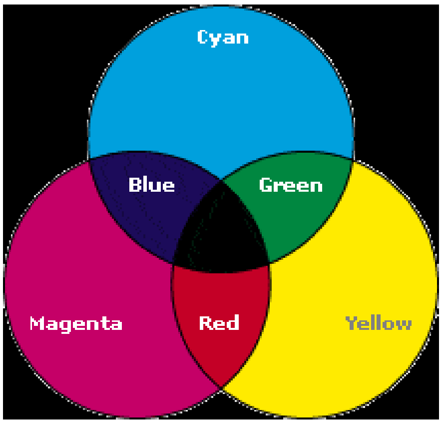
figure 3.6 (a)RGB color component (b)RGB Color Cube

The first coordinate represents the amount of red present in the hue; the second coordinate represents green; and the third coordinate refers to the amount of blue. Each coordinate must have a value between 0 and 255 for a point to be on or within the cube. Hence, pure red has the coordinate (255, 0, 0); green is located at (0, 255, 0); and blue is at (0, 0, 255). Thus, yellow is at location (255, 255, 0), and since orange is between red and yellow, its location on this cube is (255, 127, 0). The diagonal, marked as a dashed line between the colors *black* (0, 0, 0) and *white* (255, 255, 255), provides the various shades of grey. This model thus has the capability of representing 256^3 or more than sixteen million colors.

3.4.3 CMYK Color Model

- The CMYK model defines color using Cyan (C), Magenta (M), Yellow (Y) and Black (K) inks or pigments.
- The CMYK color model is a subset of the RGB model and is primarily used in color print production
- Each color contains an amount of ink that is measured with a percent from 0 to 100. A value of 100 means that the inks are applied at full saturation.

- The CMYK color model is a Subtractive Color **Model.**
- Subtractive color uses reflected light to display color.
- Printed materials are produced using the CMYK color model.
- The combination of the amounts of cyan, magenta, yellow, and black ink defines the resulting CMYK color.
- When you combine cyan, magenta, yellow, and black ink together and each component has a value of 100, then the resulting color, in theory, should be black. When the value of each component is 0, the resulting color is pure white.
- With the CMYK subtractive model, in theory, you should be able to product millions of colors, but due to the limitations of printing inks and the printing process you can only produce thousands of colors in print. Computers can display millions of CMYK colors, although they can't all be reproduced on a printer.



(a) *CMYK Color component Cube*

The CMYK color model is a subset of the RGB model and is primarily used in color print production. CMYK is an acronym for cyan, magenta, and yellow along with black (noted as K).

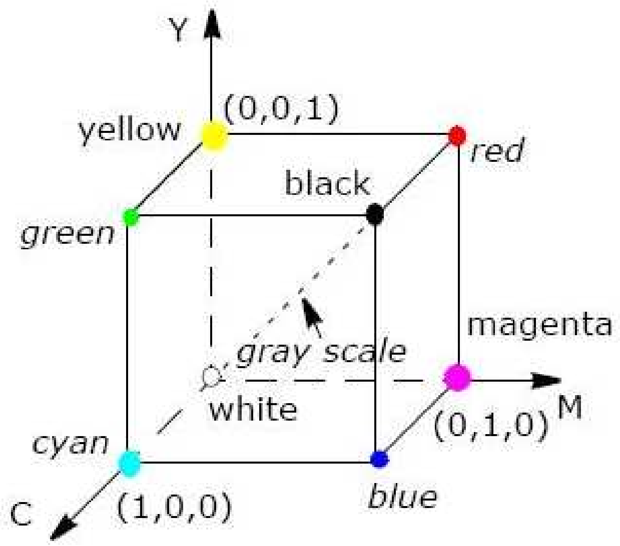


figure 3.7 (b) CMYK Color component Cube

CMY values are at three corners; red, green, and blue are the three other corners, white is at the origin; and black is at the corner farthest from the origin.

We can find CMY value from RGB of image using given equation.

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Normalize equation

3.4.4 HSB Color Model

- The HSB color model defines three fundamental properties of color: Hue, Saturation, and Brightness.

- It is predicated on the principle that every real color originates from a single pure color (Hue), which is then mixed with various amount of white or/and black color to give various shades of that pure color.
- Hue is the name or pure value of the color such as red, green, yellow, etc. It is measured in degrees from 0 to 360. (0 is Red, 60 is Yellow, 120 is Green, 180 is Cyan, 240 is Blue and 300 is Magenta.).
- According to the CIE(Commission International de1 Eclairage),hue is the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors, red, yellow, green and blue or a combination of two of them. In other words, hue is color type, such as red or green.
- Saturation is the purity of the color and is the amount of pure color mixed with white color. It varies from white to pure color. It is measured in percent from 0 to 100. The higher the percentage, the more pure will be the color.
- Saturation is the colorfulness of an area judged in proportion to its brightness. In the cone, saturation is the distance from the centre of a circular cross-section of the cone, the height where this cross section is taken is determined by the value, which is the distance from the pointed end of the cone.
- Brightness determines the intensity of the color and is the amount of pure color mixed with black color It varies from black to pure color. It is measured in percent from 0 to 100. The higher the percentage, the brighter the color.

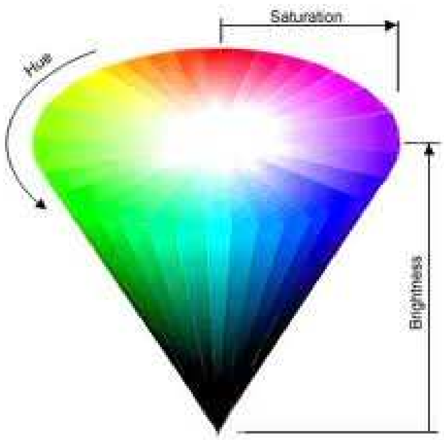


Figure 3.8 HSB Color Cone

Color Model Compare

Common High Color Models	Number of Values for each Component	Number of Possible Colors
HSB Color	Hue = 361 Saturation = 101 Brightness = 101	3,682,561
RGB Color (24-bit True Color)	Red = 256 Green = 256 Blue = 256	16,777,216
CMYK Color	Cyan = 101 Magenta = 101 Yellow = 101 Black = 101	104,060,401

3.4.5 YUV and YIQ color space

The **YUV** and **YIQ** color spaces used for television broadcast. They encode a color image or video taking human perception into account, allowing reduced bandwidth for chrominance components, thereby typically enabling transmission errors or compression artifacts to be more efficiently masked by the human perception than using a "direct" The Y'UV color model is used in the NTSC, PAL, and SECAM composite color video standards. Previous black-and-white systems used only luma (Y') information. Color information (U and V) was added separately via a sub-carrier so that a black-and-white receiver would still be able to receive and display a color picture transmission in the receivers native black-and-white format. Y' stands for the luma component (the brightness) and U and V are the chrominance (color) components; luminance is denoted by Y and luma by Y' – the prime symbols (') denote gamma compression with "luminance" meaning perceptual (color science) brightness, while "luma" is electronic (voltage of display) brightness. YIQ is same as

YUV, employed mainly in North and Central America and Japan.*I* stands for *in-phase*, while *Q* stands for *quadrature*, referring to the components used in quadrature amplitude modulation. In YUV, the U and V components can be thought of as X and Y coordinates within the color space. I and Q can be thought of as a second pair of axes on the same graph, rotated 33°; therefore IQ and UV represent different coordinate systems on the same plane

Different color modes have different size values, as shown below:

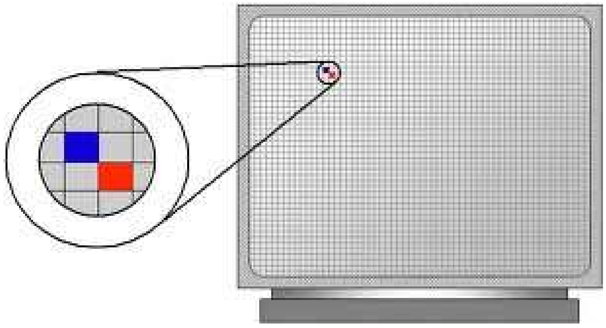
Image Type	**Bytes per pixel**
8 bit Grayscale	1 byte per pixel
16 bit Grayscale	2 bytes per pixel
24 bit RGB	3 bytes per pixel Most common for photos, for example JPG
32 bit CMYK	4 bytes per pixel For Prepress
48 bit RGB	6 bytes per pixel.

3.5 IMAGE

Pixel

- The word *pixel* is based on a contraction of *pix* ("pictures") and *el* (for "element").
- In digital imaging, a **pixel**, (**picture element)** is a single point in a raster image, or the smallest addressable screen element in a display device.
- It is the smallest unit of picture that can be represented or controlled. Each pixel has its own address. The address of a pixel corresponds to its coordinates.
- Pixels are normally arranged in a two-dimensional grid, and are often represented using dots or squares.

- Each pixel is a sample of an original image; more samples typically provide more accurate representations of the original.
- The intensity of each pixel is variable. In color image systems, a color is typically represented by three or four component intensities such as red, green, and blue, or cyan, magenta, yellow, and black.



(a) Pixel in monitor

A **(digital) color image** is a digital image that includes color information for each pixel. A digital image is composed of *pixels* which can be thought of as small dots on the screen. A digital image is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels A digital image is an instruction of how to color each pixel. A typical size of an image is 512-by-512 pixels. In the general case we say that an image is of size *m*-by-*n* if it is composed of *m* pixels in the vertical direction and *n* pixels in the horizontal direction.

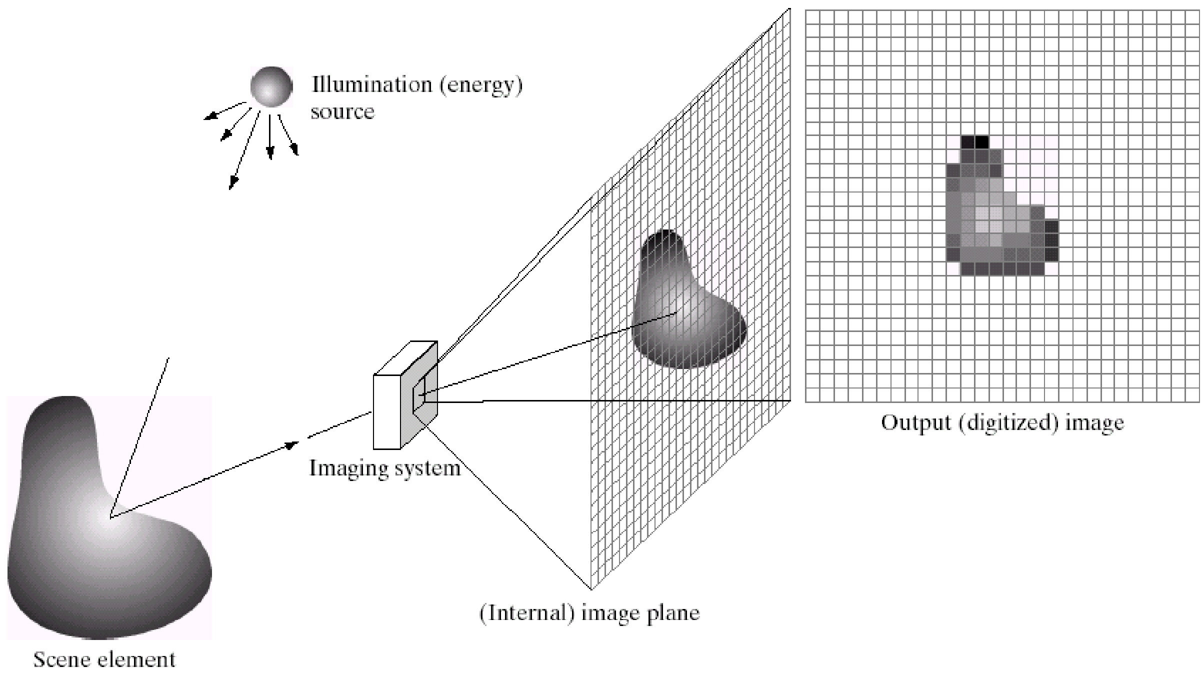


Figure 3.9(b) pixel representation of image

Let us say that we have an image on the format 512-by-1024 pixels. This means that the data for the image must contain information about 524288 pixels, which requires a lot of memory. Hence, compressing images is essential for efficient image processing. There are also a few compression techniques to reduce the amount of data required to store an image.

Bits per pixel

The number of distinct colors that can be represented by a pixel depends on the number of bits per pixel (bpp). A 1 bpp image uses 1-bit for each pixel, so each pixel can be either on or off. Each additional bit doubles the number of colors available, so a 2 bpp image can have 4 colors, and a 3 bpp image can have 8 colors:

- 1 bpp, 2^1 = 2 colors (monochrome)
- 2 bpp, 2^2 = 4 colors
- 3 bpp, 2^3 = 8 colors
- 8 bpp, 2^8 = 256 colors
- 16 bpp, 2^{16} = 65,536 colors ("Highcolor")

- 24 bpp, $2^{24} \approx 16.8$ million colors ("Truecolor")

For color depths of 15 or more bits per pixel, the depth is normally the sum of the bits allocated to each of the red, green, and blue components. Highcolor, usually meaning 16 bpp, normally has five bits for red and blue, and six bits for green, as the human eye is more sensitive to errors in green than in the other two primary colors. For applications involving transparency, the 16 bits may be divided into five bits each of red, green, and blue, with one bit left for transparency. A 24-bit depth allows 8 bits per component. On some systems, 32-bit depth is available: this means that each 24-bit pixel has an extra 8 bits to describe its opacity (for purposes of combining with another image).

3.6 Type of images

3.6.1 Binary image

- Binary image is a digital image that has only two possible values for each pixel.
- Two colors used for a binary image are black and white though any two colors can be used.
- Binary images are also called *bi-level* or *two-level*. (The names black-and-white, B&W, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as grayscale images.)
- The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.
- In Photoshop parlance, a binary image is the same as an image in "Bitmap" mode.
- Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and dithering.

- Some input/output devices, such as laser printers, fax machines, and bilevel computer displays, can only handle bi-level images.

A binary image is usually stored in memory as a bitmap, a packed array of bits. A 640×480 image requires 37.5 KB of storage. Because of the small size of the image files, fax machines and document management solutions usually use this format.

Figure 3.10 Leena Binary Image

The interpretation of the pixel's binary value is also device-dependent. Some systems interprets the bit value of 0 as black and 1 as white, while others reversed the meaning of the values.

3.6.2 Gray Scale Image

- A grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information.
- It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored
- Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

- Grayscale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bi-level or binary images).

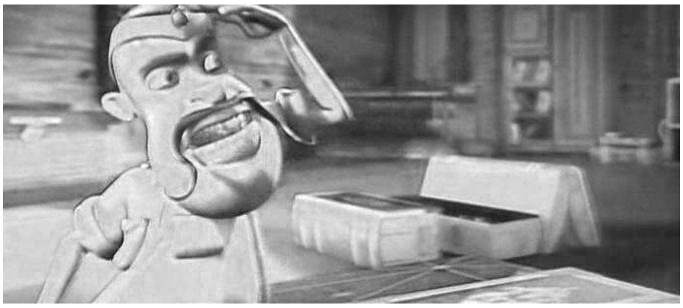
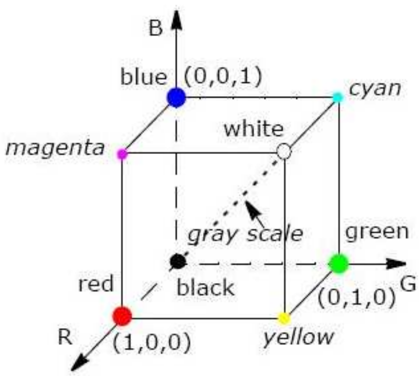
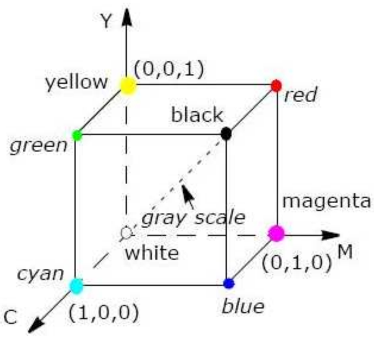


figure 3.11 Grayscale images

- Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation (i.e., one color).

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible_light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image; see the section about converting to grayscale.





RGB Color Cube *CMYK Color Cube*

Rgb2gray converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components:

Y=0.2989 * R + 0.5870 * G + 0.1140 * B

Grayscale as single channels of multichannel color images

Color images are often built of several stacked color channels, each of them representing value levels of the given channel.

For example, RGB images are composed of three independent channels for red, green and blue primary color components; CMYK images have four channels for cyan, magenta, yellow and black ink plates, etc.

Here is an example of color channel splitting of a full RGB color image. The column at left shows the isolated color channels in natural colors, while at right there are their grayscale equivalences:

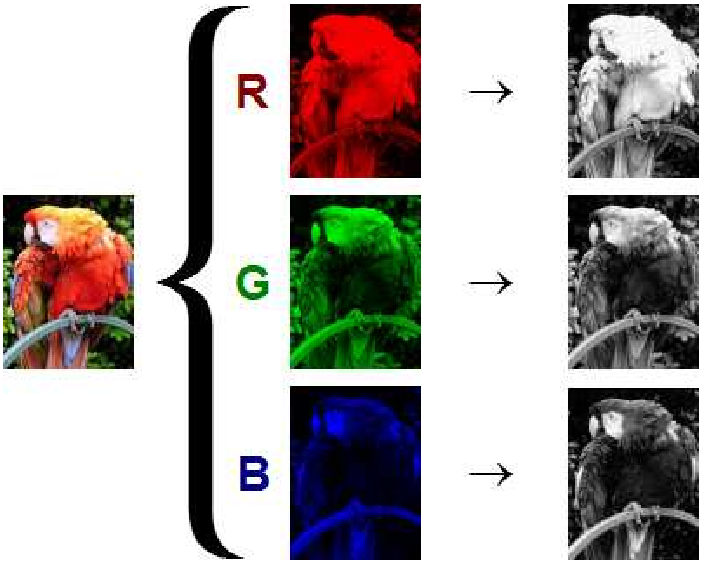


figure 3.12 RGB to Gray conversion

The reverse is also possible: to build a full color image from their separate grayscale channels. By mangling channels, using offsets, rotating and other manipulations, artistic effects can be achieved instead of accurately reproducing the original image.

3.6.3 RGB Image

- An RGB (red, green, blue) image is a three-dimensional byte array that explicitly stores a color value for each pixel.
- RGB image arrays are made up of width, height, and three channels of color information. Scanned photographs are commonly stored as RGB images.

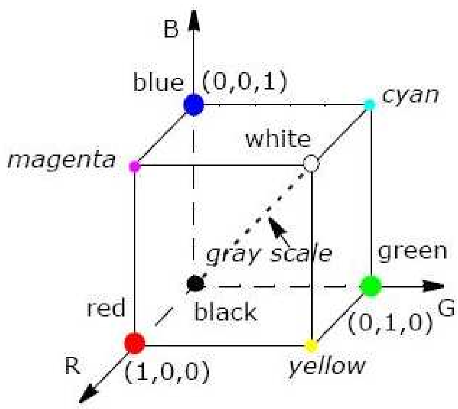


figure 3.13 RGB Cube

- The color information is stored in three sections of a third dimension of the image. These sections are known as color channels, color bands, or color layers.
- One channel represents the amount of red in the image (the red channel), one channel represents the amount of green in the image (the green channel), and one channel represents the amount of blue in the image (the blue channel).

3.7 Image Segmentation

Image segmentation refers to the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze [9]. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. In other word, Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).[9] When applied to a stack of images, typical in Medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like Marching cubes.

Image Segmentation Technique:

Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc.

Edge-based methods center around contour detection: their weakness in connecting together broken contour lines make them, too, prone to failure in the presence of blurring.

A **region-based method** usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge. Hybrid techniques using a mix of the methods above are also popular.

A **connectivity-preserving relaxation-based segmentation** method, usually referred to as the active contour model, was proposed recently. The main idea is to start with some initial boundary shapes represented in the form of spline curves, and iteratively modify it by applying various shrink/expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an ``elastic'' contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is no easy task.

Chapter 4
Proposed Method

The management of a large number of images in a multimedia database has received much attention in recent years. Most of the earlier works are largely focused on techniques to extract useful information (such as color, texture or shape) that represents images. Rapid retrieval is becoming important issue as image databases continue to grow in size and a slow will no longer be acceptable to the user community.

So in this chapter we proposed an algorithm, in which we will first preprocess the original set of images on certain factor like hue, saturation value, edges, brightness, etc. **We can also consider segmented images , texture, color value as factors but to keep our demonstration simple we will consider simple factors like edge value , saturation value , brightness, hue, edge etc.**

For all factors, we will generate different image sets, corresponding to each factor taken. After preprocessing we will then select the image to be compares and will apply similarity test within same image set.

Then we will note down the result as per degree of similarity value, for different result set. Most similar image will be given priority 1, then next one will be of priority 2 and so on.

After finding priority values for each image set ,we will traverse priority wise and we will now check the maximally occurring image at given priority and will add that image to our result , as per rules proposed in algorithm.

Finally we will get the set which represent the set of images having similar to chosen images in order from most similar to least similar.

Note that our result is based on chosen factors. More appropriate will be chosen factors, better will be results.

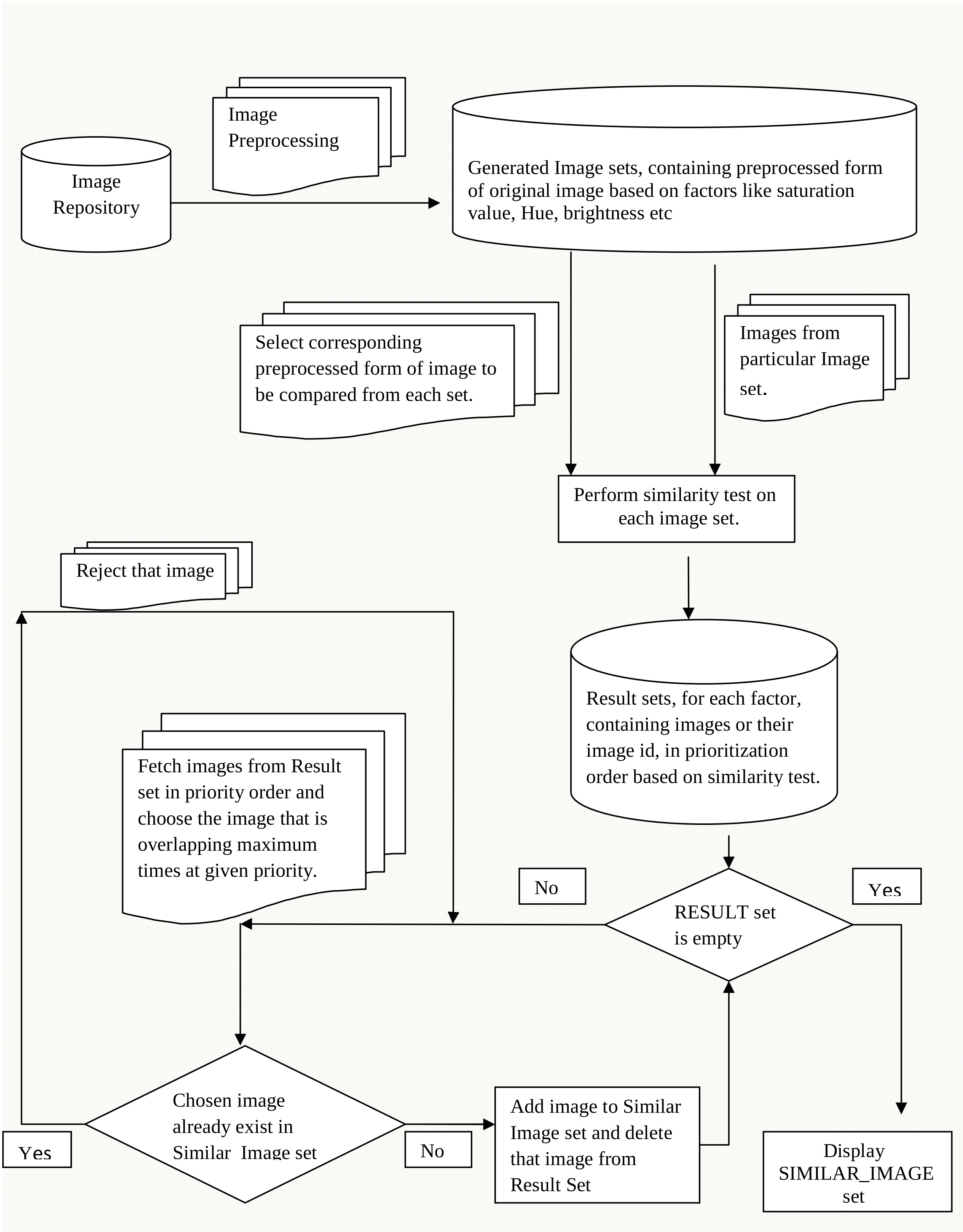


Fig 4.1. Block diagram of searching image in repository based on overlapping method..

Assumptions:

- Here we are assuming that images in repository are animated images.

Algorithm:

<Preprocessing Step>

Step 1: Generate various sets of images. Each set contains preprocessed form of original Image based on certain factors. These factors include hue, saturation, brightness, Edge etc.

$IMAGE_i$ = <Different set of images preprocessed on certain factor>, where i € Total number of factors considered.

<Similarity Test over each $IMAGE_i$ set>

Step 2: Now select the corresponding preprocessed form of the image from each $IMAGE_i$ set ,which you want to compare and perform similarity test over each $IMAGE_i$ set.

Step 3: Note the prioritization order of images, which is result of each similarity test, and save it in $RESULT_i$ set in ascending value of priority value. Images with highest Similarity value have priority 1 , then next image will have priority 2 and so on. i.e.

$RESULT_i$ = Similarity Test($IMAGE_i$) ,where i € Total number of factors considered.

Note: For demo purpose, our similarity test will be as follows. First resize the image into 300*300 pixels. Then divide the whole image into 5*5 regions, and each region containing 30*30 pixels. To find similarity, we will then calculate the Euclidean distance between the regions and accumulate. Smaller will be Euclidean distance, higher will be similarity.

<Overlapping image at particular priority in different RESULT $_i$ set>

Step 5: Set TEMP_PRIORITY = 1

Set SIMILAR_IMAGE = <NULL>

where TEMP_PRIORITY is a variable keeping track of current priority number and SIMILAR_IMAGE is a set of similar images as a result of our algorithm , i.e set of images similar to chosen image .

Step 6: While all RESULT $_i$ sets is not empty,

REPEAT steps (a) to (d)

a) Fetch the image$_j$ from all RESULT $_i$ sets that is overlapping maximum times at given priority, having priority = TEMP_PRIORITY.

b) If image$_j$ already exist in SIMILAR_IMAGE set ,

i) From all result set RESULT $_i$, delete image$_j$ i.e RESULT $_i$ = RESULT $_i$ – image$_j$,

ii) goto step (a) .

c) If two image$_j$ e g. x, y have same no of occurrence then put it in waiting queue W. leave SIMILAR_IMAGE $_j$ output image blank for that RESULT$_i$ and set l=j.

i) Go for next row and step a.

j) If maximum occurred image x is already in W queue, put y into SIMILAR_IMAGE$_l$ set and y in SIMILAR_IMAGE $_k$.

c) SIMILAR_SET = SIMILAR_SET + image$_j$.

d) TEMP_PRIORITY = TEMP_PRIORITY + 1.

STEP 7: Display SIMILAR_IMAGE set as final result.

Example

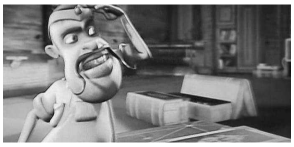
We are taking an example to clarify our approach. Initially we have set of Colored animated images. Now we will preprocess each image on various factors like edge, saturation, lightness etc and save it in IMAGE $_i$ set correspondingly.

IMAGE $_i$ = <Different set of images preprocessed on certain factor>,

Where i € Total number of factors considered.

Image Preprocessing

$IMAGE_1$ = <Binary>



$IMAGE_2$ = <Lightness>

$IMAGE_3$ = <Saturation >

$IMAGE_1$ = <Edge >

Now select the image that you want to compare and perform similarity test and save result in RESULT $_i$ set i.e.

RESULT $_i$ = Similarity Test(IMAGE $_i$) , where i € Total number of factors considered.

RESULT 1 based on Edge Similarity Test	RESULT 2 based on Lightness Similarity Test	RESULT 3 based on Saturation Similarity Test	RESULT 4 based on Binary Similarity Test
• Pic5	• Pic5	• Pic5	• Pic5
• Pic7	• Pic3	• Pic7	• Pic7
• Pic2	• Pic8	• Pic2	• Pic2
• Pic4	• Pic7	• Pic4	• Pic4
• ...	• ...	• ...	
• ...	• ...	• ...	

Now we will fetch images from result set RESULT $_i$, which is maximally overlapping at given priority value and will add it to similar image set SIMILAR_IMAGE $_i$.as per algorithm.

Edge E	Value V	Binary B	Saturation S	Luminance
Pic5	Pic5	Pic5	Pic5	Pic5
Pic7	Pic7	Pic8	Pic7	Pic8
Pic18	Pic18	Pic19	Pic18	Pic17
Pic8	Pic8	Pic7	Pic7	Pic7
:	:	:	:	:

Pic5 is maximally overlapping at priority 1, so add it to SIMILAR_IMAGE set

Pic7 is maximally overlapping at priority 2, so add it to SIMILAR_IMAGE set

Pic7 is maximally overlapping at priority 4, but it is already existing in SIMILAR_IMAGE set , so we will delete it from RESULT set from this priority and will look for next maximally overlapping image at priority 4, which is Pic8, so add Pic8 to SIMILAR_IMAGE set

At the end , we will display SIMILAR_IMAGE set as result , which represents set of images from most similar upto least similar image , of chosen reference image.

Chapter 5

Experimental Results

This Chapter deals with the results of retrieving images from the Repositary using the method which is discussed in previous chapters of this thesis.

5.1 Development Environment

5.1.1Java (programming language)

Java is a programming language originally developed by James Gosling at Sun Microsystems (which is now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications and image processing

5.1.2 NetBeans Platform

The Net Beans Platform is a broad java Swing-based framework on which you can base large desktop applications. The IDE itself is based on the Net Beans Platform. The Platform contains APIs that simplify the handling of windows, actions, files, and many other things typical in applications.

Each distinct feature in a NetBeans Platform application can be provided by a distinct Net Beans module, which is comparable to a plug-in. A Net Beans module is a group of Java classes that provides an application with a specific feature.

You can also create new modules for NetBeans IDE itself. For example, you can write modules that make your favourite cutting-edge technologies available to users of NetBeans IDE. Alternatively, you might create a module to provide an additional editor feature. NetBeans is a free, open-source Integrated Development Environment for software developers.

5.1.3 Working with Images

Images are described by a width and a height, measured in pixels, and has a coordinate system that is independent of the drawing surface.

There are a number of common tasks when working with images.

Loading an external GIF, PNG JPEG image format file into Java 2D™'s internal image representation.

Directly creating a Java 2D image and rendering to it.

Drawing the contents of a Java 2D image on to a drawing surface.

Saving the contents of a Java 2D image to an external GIF, PNG, or JPEG image file.

Class

There are two main classes that work with images:

The **java.awt.Image** class is the superclass that represents graphical images as rectangular arrays of pixels.

The **java.awt.image.BufferedImage** class, which extends the Image class to allow the application to operate directly with image data (for example, retrieving or setting up the pixel color). Applications can directly construct instances of this class.

BufferedImage

The BufferedImage class is a cornerstone of the Java 2D immediate-mode imaging API. It manages the image in memory and provides methods for storing, interpreting, and obtaining pixel data. Since BufferedImage is a subclass of Image it can be rendered by the Graphics and Graphics2D methods that accept an Image parameter.

A BufferedImage is essentially an Image with an accessible data buffer. It is therefore more efficient to work directly with BufferedImage. A BufferedImage has a ColorModel and a Raster of image data. The ColorModel provides a color interpretation of the image's pixel data.

The Raster performs the following functions:

- Represents the rectangular coordinates of the image.
- Maintains image data in memory.
- Provides a mechanism for creating multiple sub images from a single image data buffer.
- Provides methods for accessing specific pixels within the image.

Basic operations with images

The basic operations with images are represented in the following sections:

Reading/Loading an image

This section explains how to load an image from an external image format into a Java application using the Image I/O API.

Drawing an image

This section teaches how to display images using the drawImage method of the Graphics and Graphics2D classes.

Creating and drawing to an image

This section describes how to create an image and how to use the image itself as a drawing surface.

Writing/saving an image

This section explains how to save created images in an appropriate format.

5.1.4 Java Advanced Imaging (JAI)

Java Advanced Imaging (JAI) is a Java platform extension API that provides a set of object-oriented interfaces that support a simple, high-level programming model which allows developers to create their own image manipulation routines without the additional cost or licensing restrictions, associated with commercial image processing software.

5.2 Experimental Setup

We have created Graphical user interface in java using SWING which provides user a interface by which one can select query image from the repository and can find similar images for that.

Operating System used is Windows Xp, with 3.0 GHz CPU. We have taken more than 40 mixed natural images as a database.

Now after building graphical user interface we have select the Input Image from repository

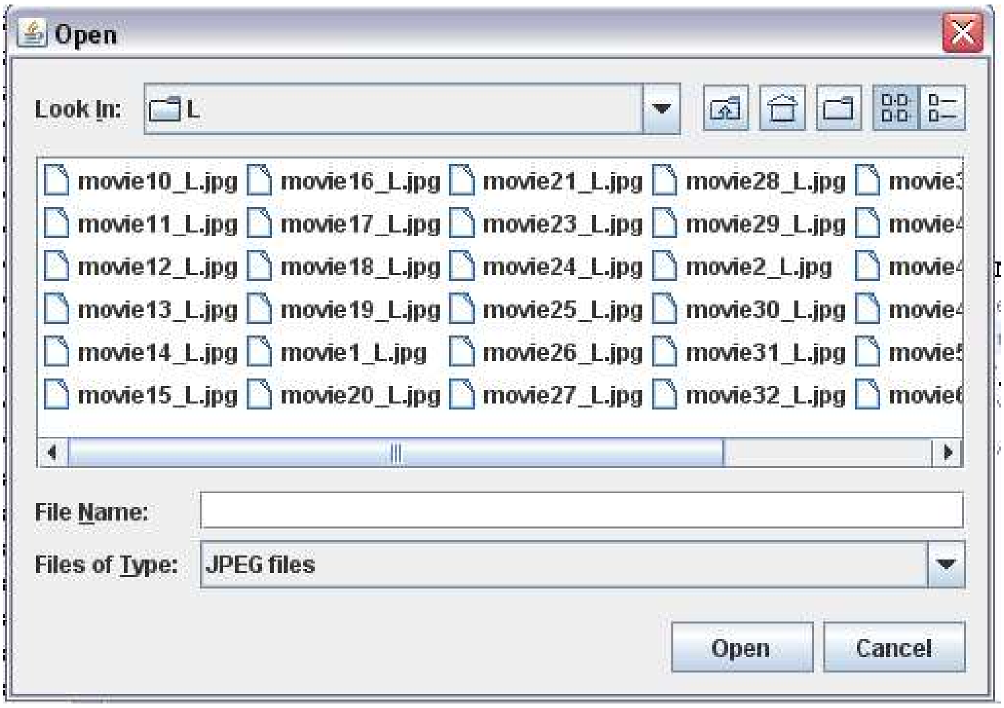


Figure 5.1 Selection of input image

(a)

(b)

figure5.2: (a) Query image (b) Compare luminance value of images.

figure5.3: Compare in Edge value of images with query image

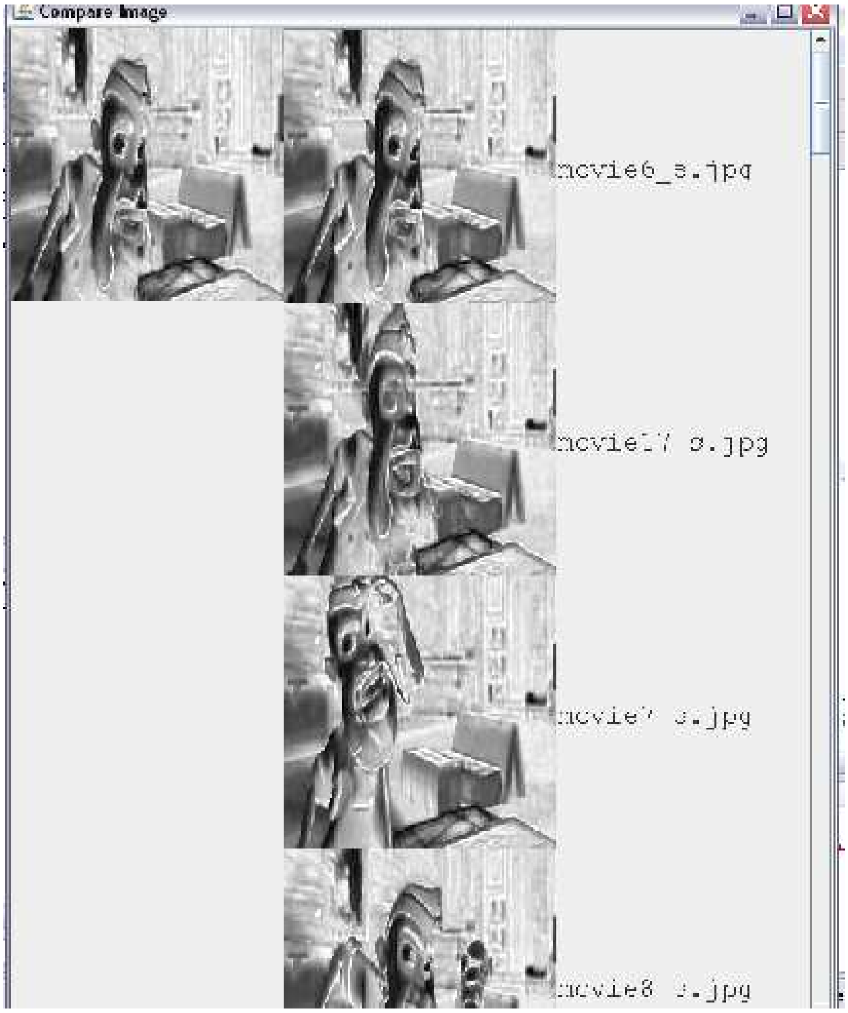


figure5.4: Compare in Saturation value of images with query image

5.3 Obtained Result

After comparing the query image with different image over different parameter. We store the result in Table to conclude our Result. Here we are taking the sample of 7 images set to find out our result using our algorithms.

Edge	**Saturation**	**Luminance**	**Brightness**	**HUE**	**Result**
5	5	5	5	5	**5**
17	7	7	7	7	**7**
7	6	18	18	18	**18**
18	18	17	17	19	**17**
8	8	6	6	9	**6**
6	17	8	8	8	**8**
25	21	25	10	25	**25**

5.4 Final Result

In the above table we will now evaluate each row and will find maximally occurring image at each row and will add it to SIMILAR_IMAGE set. If the image already exists in SIMILAR_Image set then we will evaluate next maximally occurring image at that row.

SIMILAR_IMAGE set
5
7
18
17
6
8
25

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The dramatic rise in the sizes of images databases has stirred the development of effective and efficient retrieval systems. The development of these systems started with retrieving images using color feature is not sufficient. Systems retrieve images from repository using Overlapping cluster in given visual features such as color, texture and shape, as opposed to depending on image descriptions or textual indexing. In this project, we have researched various modes of representing and retrieving the image properties of color, and image feature. The application performs a simple search in an image dataset for an input query image, using image feature such as edge, saturation, luminance etc. It then compares the query image with data set image on different images feature using the Euclidean Distance Equation. Further enhancing the result, the application performs a texture-based search in the color results, using wavelet decomposition and energy level calculation. It then compares the texture features obtained using the Euclidean Distance Equation. And it is shown that our method gives better results than taking color only.

6.2 Future Work

The work presented in this thesis can be further enhanced by finding the appropriate factors which represents the image uniquely. More appropriate will be factors , and more will be the factors , larger will be degree of similarity between matching images , hence more appropriate will be the results.

References

1. Hu Min and Yang Shuangyuan “ Overview of Image Mining Research” at The 5th International Conference on Computer Science & Education Hefei, China. August 24–27, 2010
2. Ji Zhang ,Wynne Hsu and Mong Li Lee “IMAGE MINING: ISSUES, FRAMEWORKS AND TECHNIQUES” .
3. Peter Stanchev Kettering University Flint, Michigan “USING IMAGE MINING FOR IMAGE RETRIEVAL” **Conference**” Computer Science and Technology”, Maxico May 19-21 2003.
4. Aura Conci, Everest Mathias and M. M. Castro “IMAGE MINING BY COLOR CONTENT”.
5. Feature Extraction for Image Mining “Deepak Kolippakkam, Huan Liu and Amit Mandvikar”
6. Yu changjinand Xia hongxia “The Investigation of Image Mining Framework” at 2009 IEEE
7. Ido Omer and Michael Werman” Color Lines: Image Specific Color Representation”
8. Dengsheng Zhang and Guojun Lu “EVALUATION OF SIMILARITY MEASUREMENT FOR IMAGE RETRIEVAL”
9. Linda G. Shapiro and George C. Stockman (2001): “Computer Vision”,
10. Patricia G. Foschi , Deepak Kolippakkam, Huan Liu and Amit Mandvikar “Feature Extraction for Image Mining”.

11. Fuhui Long, Hongijang Zhang, David Dagan Feng, "Fundamentals of Content Based ImageRetrieval".http://research.microsoft.com/asia/dload_files/group/mcomputing/2003P/ch01_Long_v40-proof.pdf
12. Sharmin Siddique, "A Wavelet Based Technique for Analysis and Classification of Texture Images," Carleton University, Ottawa, Canada, Project Report 70.593, April 2002.
13. Chin Chen-Chaur, Ting Chu Hsueh, "Similarity Measurement between Images", IEEE Annual International Computer Software and Applications Conference COMPSAC'05), pp.207-248, 2005.
14. Timothy K.Shih,Jung-yao Huang,Ching-Sheng Wang,Jason C.Hung And Chuan-Ho Kao, " An intelligent content-Based image retrieval System based on color ,Shape and spatial relations" Proceedings Natl.Si.Counc.Roc(A), volume 25,No.4, pp.231-243,2001.
15. Virginia E. Ogle and Michael Stonebraker, "Chabot: Retrieval from a relational database of images", IEEE Computer, 28(9), pp.40–48, September 1995.
16. A. Blaser, "Database Techniques for Pictorial Applications", Lecture Notes in Computer Science, Springer Verlag GmbH, Volume 81, 1979.
17. J. Kreyss, M. R¨oper, P. Alshuth, Th. Hermes, and O. Herzog, "Video retrieval by still image analysis ImageMiner", In Proceedings of IS&T/SPIE's Symposium on Electronic Imaging: Science & Technologies, pp.8-14 Feb. '97, San Jose, CA, 1997.

18. S. Sclaroff, L. Taycher, and M. La Cascia, “ Imagerover: A content-based image browser for the world wide web”, In Proceedings IEEE Workshop on Content-based Access of Image and Video Libraries, June ’97, 1997.

19. Leonid Taycher, Marco La Cascia, and Stan Sclaroff, “Image digestion and relevancefeedback in the ImageRover WWW search engine”, In Proceedings of the 2nd International Conference on Visual Information Systems, San Diego, December ’97, pp 85–94, 1997.

20. A. B. Benitez, M. Beigi, and S.-F. Chang, “Using relevance feedback in content-based image metasearch”, IEEE Internet Computing, 2(4):59–69, July/August 1998.

21. Wei-Ying Ma and B. S. Manjunath, “ Netra: A toolbox for navigating large image databases”,Multimedia Systems, 7(3):184–198, 1999.

22. Alex Pentland, RosalindW. Picard, and Stanley Sclaroff, “Photobook: Content-based manipulation of image databases”, International Journal of Computer Vision, 18(3):233–254, June 1996.

23. 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 Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases, 2-3 February ’93, San Jose, CA, pages 173–187, 1993.

24. J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Gorowitz, R. Humphrey, R. Jain, andC.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.

25. 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.
26. J. R. Smith, "Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis" PhD thesis, Graduate School of Arts and Sciences, Columbia University, February 1997.
27. Michael J. Swain, Charles Frankel, and Vassilis Athitsos, "WebSeer: An image search engine for the World Wide Web", Technical Report TR-96-14, Department of Computer Science, University of Chicago, July 1996.