[UCI]I804:24011-200000239387

2010 년 2 월

석사학위논문

색상과 에지 특징을 포함하는 새로운 이미지 블록 기술에 근거한 CBIR

조선대학교 대학원

전자공학과

추호명

색상과 에지 특징을 포함하는 새로운 이미지 블록 기술에 근거한 CBIR

Content based Image Retrieval based on

A Novel Image Block Technique Combining

Color and Edge Features

2010 년 2 월 25 일

조선대학교 대학원

전자공학과

추호명

색상과 에지 특징을 포함하는 새로운 이미지 블록 기술에 근거한 CBIR

지도교수 박세승

이 논문을 석사학위신청 논문으로 제출함

2009 년 10 월

조선대학교 대학원

전자공학과

추호명

추호명의 석사학위 논문을 인준함

위원장　조선대학교　교수 ________ (인)

위　원　조선대학교　교수 ________ (인)

위　원　조선대학교　교수 ________ (인)

2009 년 11 월

조선대학교 대학원

ABSTRACT

Content based Image Retrieval based on A Novel Image Block Technique Combining Color and Edge Features

Zou Haoming

Advisor: Prof. Sei Seung Park

Department of Electronics Engineering

Graduate School of Chosun University

The growth of multimedia information has been enormous recently, especially with the advent of the Internet. A huge digital image archive is made up of millions of images, photos created by hospitals, governments, companies and academic organizations. Originally, searching and retrieving images was based on the keywords of images. In the early 1990s, Content-based Image Retrieval (CBIR) was proposed to overcome the limitations of Text-based Image Retrieval. It is a typical task of computer vision. In CBIR, images in a database are indexed using their own primitive visual features instead of human annotations such as shapes, colors, and textures. The use of different visual features is also a criterion to categorize a

CBIR system. Since the visual features of an image are only based on the image itself, there is no problem of subjectivity.

In this paper a CBIR algorithm which was based on image block method that combined both color and edge feature was proposed. In consideration of the main drawback of global histogram representation is dependent of the color without spatial or shape information, a new image block method that divided the image to 8 related blocks which contained more information of the image is utilized to extract image feature. Based on these 8 blocks, histogram equalization and edge detection techniques are also used for image retrieval. The experimental results show that the proposed image block method has better ability of characterizing the image contents than traditional block method and could perform retrieval system efficiently. The simulation was performed using MATLAB 7.0.

Table of Contents

List of Tables

List of Figures

I. Introduction

A. Image processing and Computer vision

1. Image processing

We are in the midst of a visually enchanting world, which manifests itself with a variety of forms and shapes, colors and textures, motion and tranquility. The human perception has the capability to acquire, integrate, and interpret all this abundant visual information around us. It is challenging to impart such capabilities to a machine in order to interpret the visual information embedded in still images, graphics, and video or moving images in our sensory world. It is thus important to understand the techniques of storage, processing, transmission, recognition, and finally interpretation of such visual scenes.

The first step towards designing an image analysis system is digital image acquisition using sensors in optical or thermal wavelengths. A two-dimensional image that is recorded by these sensors is the mapping of the three-dimensional visual world. The captured two dimensional signals are sampled and quantized to yield digital images.

Sometimes we receive noisy images that are degraded by some degrading mechanism. One common source of image degradation is the optical lens system in a digital camera that acquires the visual information. If the camera is not appropriately focused then we get blurred images. Here the blurring mechanism is the defocused camera. Very often one may come across images of outdoor scenes that were procured in a foggy environment. Thus any outdoor scene captured on a foggy winter morning could invariably result into a blurred image. In this case the degradation is due to the fog and mist in the atmosphere, and this type of degradation is known as atmospheric degradation. In some other cases there may be a relative motion between the object and the camera. Thus if the camera is given an impulsive displacement during the image capturing interval while the object is static, the resulting image will invariably be blurred and noisy. In some of the above cases, we need appropriate techniques of refining the images so that the resultant images are of better visual quality, free from aberrations and noises. Image enhancement, filtering, and restoration have been some of the important applications of image processing [1].

Segmentation is the process that subdivides an image into a number of uniformly homogeneous regions. Each homogeneous region is constituent part or object in the entire scene. In other words, segmentation of an image is defined by a set of regions that are connected and nonoverlapping, so that each pixel in a segment in the image acquires a unique region label that indicates the region it belongs to. Segmentation is one of the most important elements in automated image analysis, mainly because at this step the objects or other entities of interest are extracted from an image for subsequent processing, such as description and recognition.

After extracting each segment, the next task is to extract a set of meaningful features such as texture, color, and shape. These are important measurable entities which give measures of various properties of image segments. Some of the texture properties are coarseness, smoothness, regularity, etc, while the common shape descriptors are length, breadth, aspect ratio, area, location, perimeter, compactness, etc. Each segmented region in scene may be characterized by a set of such features.

Finally based on the set of these extracted features, each segmented object is classified to one of a set of meaningful classes. In a digital image of ocean, these classes may be ships or small boats or even naval vessels and a large class of water body. The problems of scene segmentation and object classification are two integrated areas of studies in machine vision. Expert systems, semantic networks, and neural network-based systems have been found to perform such higher-level vision tasks quite efficiently.

2. Computer vision

Computer vision is probably the most exciting branch of image processing, and the number of applications in robotics, automation technology and quality control is constantly increasing [2].

It is the science and technology of machines that see. As a scientific discipline, computer vision is concerned with the theory for building artificial systems that obtain information from images. The image data can take many forms, such as a video sequence, views from multiple cameras, or multi-dimensional data from a medical scanner.

A computer vision system processes images acquired from an electronic camera, which is like the human vision system where the brain processes images derived from the eyes. Computer vision is a rich and rewarding topic for study and research for electronic engineers, computer scientists and many others. Increasingly, it has a commercial future. There are now many vision systems in routine industrial use: cameras inspect mechanical parts to check size, food is inspected for quality, and images used in astronomy benefit from computer vision techniques. Forensic studies and biometrics (ways to recognize people) using computer vision include automatic face recognition and recognizing people by the 'texture' of their irises. These studies are paralleled by biologists and psychologists who continue to study how our human vision system works. Figure 1.1 shows the related fields of computer vision.

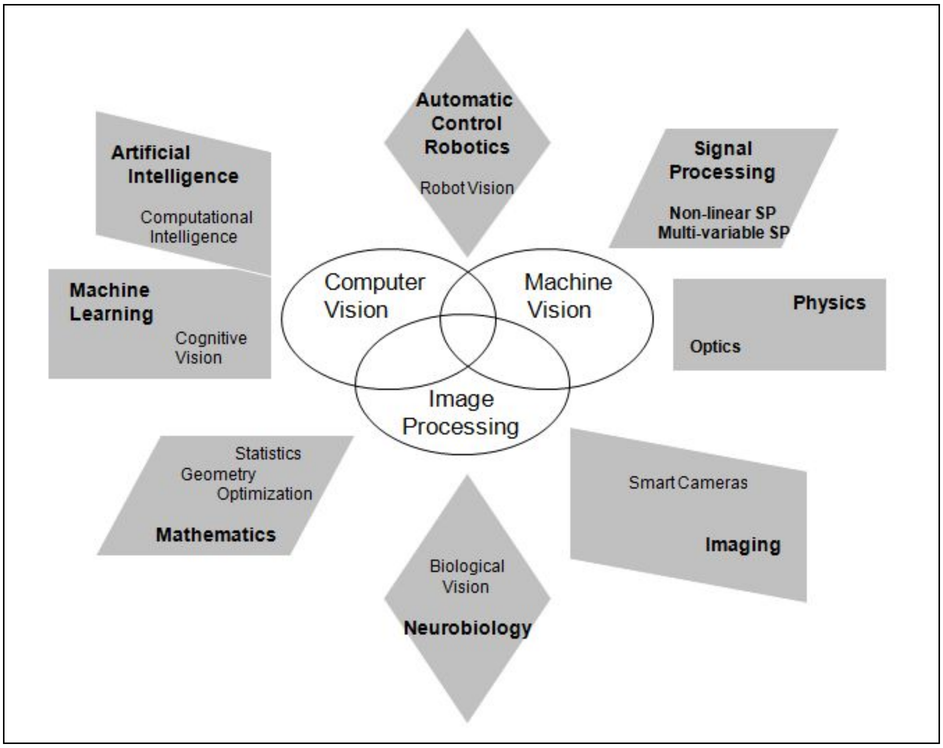


Figure 1.1: The related fields of computer vision.

B. Content-based Image Retrieval

Content-based image retrieval (CBIR), also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR) is the application of computer vision to the image retrieval problem, that is, the problem of searching for digital images in large databases. [3].

"Content-based" means that the search will analyze the actual contents of the image. The term 'content' in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself. Without the ability to examine image content, searches must rely on metadata such as captions or keywords, which may be laborious or expensive to produce. The low level visual features such as color, texture and shape of image are the main contents of CBIR as shown in Figure 1.2.

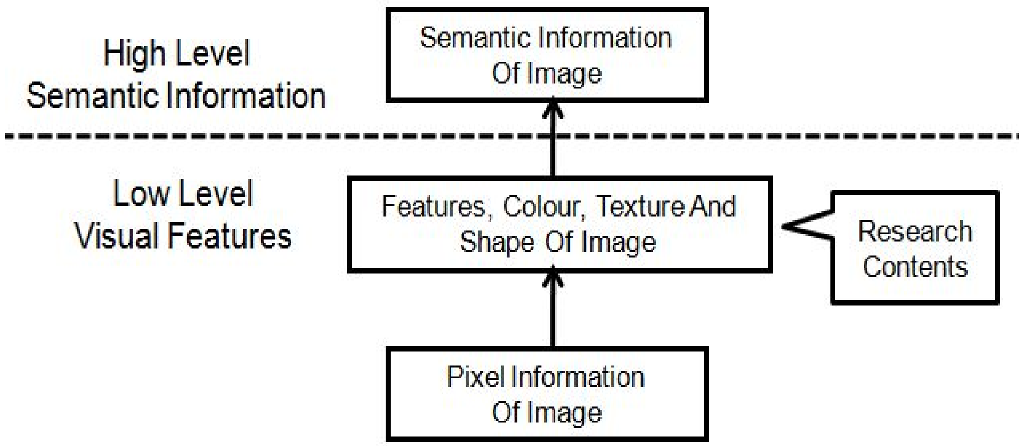


Figure 1.2: Main contents of CBIR.

CBIR is currently an active research area in computing science, as more and more visual information –especially digital images – has been made available in digital archives. The basic task of a CBIR system is to find similar images according to their visual features, within a large image database. Query By Example (QBE) is one of the most popular methodologies used in CBIR systems, in which images are selected from an image database similar to a given image presented by users. IBM's QBIC [4] is such a CBIR system.

Typically, a CBIR system pre-processes an image database in order to extract information from all images in the database. This information is referred to as visual features of images. The visual feature may be represented by different visual feature representations, such as the image's color histogram. Different CBIR systems can be categorized based on different visual features extracted, as well as different abstract visual feature representations used. When an example image is given, the same visual feature is extracted from this image and used to match all the visual features of images in the image database. Based on some distance metric, the image with the smallest distance to the given image is retrieved as the result. In current CBIR systems, instead of only retrieving the best image, a number of similar images are retrieved and sorted according to their distance to the given image.

Since CBIR is the central part of this thesis, in Chapter 2, the main theory of CBIR systems and related work will be discussed in detail.

II. CBIR and Related work

A. Background of CBIR

The growth of multimedia information has been enormous recently, especially with the advent of the Internet. A huge digital image archive is made up of millions of images, photos created by hospitals, governments, companies and academic organizations. These images are useful in many fields. For instance, a large number of satellite photos can be used for weather forecasting purposes; X-ray photos of human bodies are useful in the medical field; images of human faces are critical for the police to identify criminals. However, we cannot access or make use of these images unless they are well organized and easily retrievable. Searching for the face of a specific person in millions of facial images is very tedious and frustrating. Originally, searching an image database was based on human annotation: each image in a database is given some keywords to denote the semantic content of the image; then, all the keywords are used to index images. Thus, searching and retrieving images is based on the keywords of images. This is called Text-based Image Retrieval [5].This approach is easy to understand in comparison with other approaches we will be discussing later. However, as we can easily see, this approach has many limitations.

- As the size of image collections gets increasingly large, manually giving each image an annotation is impractical.
- Annotating an image based on human perception is subjective. Different people may give different annotations to images with similar visual contents.

In the early 1990s, Content-based Image Retrieval (CBIR) was proposed to overcome the limitations of Text-based Image Retrieval. In CBIR, images in a database are indexed using their own primitive visual features instead of human annotations such as shapes, colors, and textures. The use of different visual features is also a criterion to categorize a CBIR system. Since the visual features of an image are only based on the image itself, there is no problem of subjectivity. When an example image is given, its visual features are also extracted and used to match against those in the database. Some distance metrics are used to compute the similarity between the query image and images in the database. The result of the query is a set of images similar to the query image, rather than an exact match. These result images are also sorted according to their distance to the query image. Visual feature extraction, distance metric and different CBIR techniques will be discussed in detail in the following sections. However, CBIR also has its own limitations. The main problem is that it cannot deal with semantic-level image queries effectively. An image always contains some semantic information (for example, an image containing a little boy). Such semantic features can only be represented by some primitive features in present CBIR systems. For example, a semantic-level query searching for images with green grass can be represented by a primitive image query which searches images with green color in the bottom part. Since present CBIR systems cannot extract images' semantic features effectively, they cannot satisfy most semantic-level image queries.

B. Visual Features used in CBIR Systems

In a CBIR system, different visual features of images are automatically extracted and stored for any future retrieval process. Searching the whole image database is based on searching these visual features – metadata of real images. Colors, shapes, and textures are the most widely used in most CBIR systems.

1. Color Features

Color feature is one of the most widely used visual features in image retrieval since color is immediately perceived by human beings [6], when looking at an image. Therefore, Color-based Image Retrieval is the most popular CBIR technique. Using color features in CBIR requires taking many factors into consideration: color model selection, color feature representation, and the metric to compute the distance between color features.

The purpose of a color model is to facilitate the specification of colors in a standard way. In other words, a color model is the quantitative way to represent colors that human beings perceive. Color models used today can be classified into two categories: hardware-oriented and user-oriented [6]. Hardware-oriented color models are used for most color devices. For instance, the RGB (red, green, blue) color model is used for color monitors and cameras; the CMY (cyan, magenta, yellow) color model is used for color printers; and the YIQ color model is used for color TV broadcast. User-oriented color models including HLS, HCV, HSV, MTM and CIE-LUV, are based on the three human perceptions of colors, i.e., hue, saturation, and brightness. The selection of color models determines the way to represent color content of images as well as consecutive color feature representations and the selection of image retrieval techniques. Therefore, a color model is very important in a CBIR system. In order to understand many characteristics of color features used in CBIR systems, some commonly used color models are discussed next.

The RGB color model is the most commonly used color model. In the RGB color model, a color is represented by a combination of three primary colors: red, green, blue. The RGB color model actually can be represented by a color cube, as shown by Figure 2.1. As we can see, red, green, and blue colors are at three corners of the cube while cyan, magenta, and yellow are at three different corners. The values for red, green and blue increase respectively from the origin point along the axis on which they reside. The origin point represents color BLACK, while the point furthest from the origin point represents the color WHITE. The line connecting the BLACK color point and the WHITE color point represents all colors in grayscale. Gray colors on this line also change from pure black to pure white. All the other colors are represented as a point within the cube. The value for a specific color is defined by the co-ordinate of the point in the cube along the red, green, and blue dimensions. Many kinds of image formats such as JPGs and GIFs store and show colors in the RGB color model.

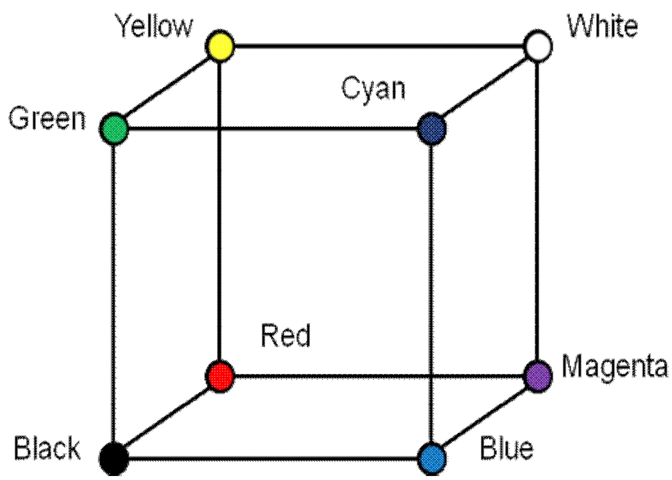


Figure 2.1: RGB color model.

The CMY color model is similar to the RGB color model except that, in its color model cube, the origin point is WHITE and the furthest point opposite to the origin point is the color BLACK as shown in Figure 2.2. The CMY color model is used mainly in color printing. When a color image displayed by a color monitor is going to be printed by a color printer, conversion from the RGB color model to the CMY color model is performed.

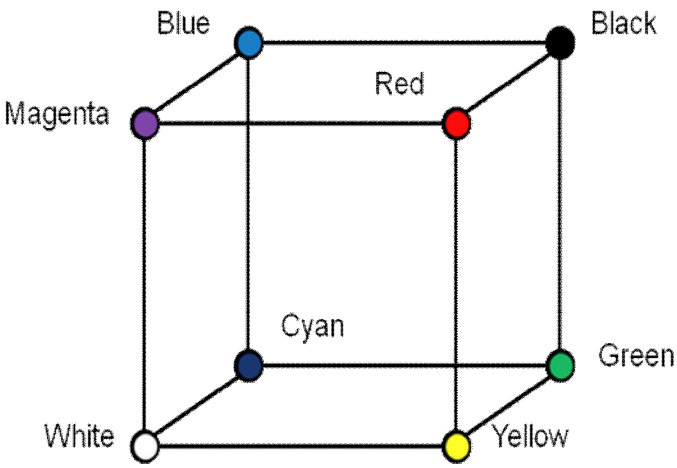


Figure 2.2: CMY color model.

The CMY color can be obtained by converting from RGB color as shown by Equation 2.1:

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.1}$$

There are many other color models such as HSV, L*u*v*, and YIQ. Among these, the RGB color model has been the most widely used. Since many image formats, such as JPGs and GIFs, store and show colors in the RGB color model, most CBIR techniques are based on this color model. In addition, no evidence shows that other color models can represent image visual features better than the RGB color model obviously. Therefore, our thesis also uses techniques based on the RGB color model. The discussion about color features, as well as CBIR techniques, in the following sections will assume that the RGB color model is used, except when specifically specified.

2. Shape features

Shape information of objects in images is also a very important image visual feature. Searching image databases using shape-based techniques is very common in CBIR systems. Usually in such a CBIR system, shape features of all images in the database are extracted and indexed, as well as query images. The system then searches the database to find images with a similar shape features to the query image. Segmentation is the most important step during image shape feature extraction and includes many procedures such as noise removing and edge detection. Figure 2.3 is an example of shape feature extraction.

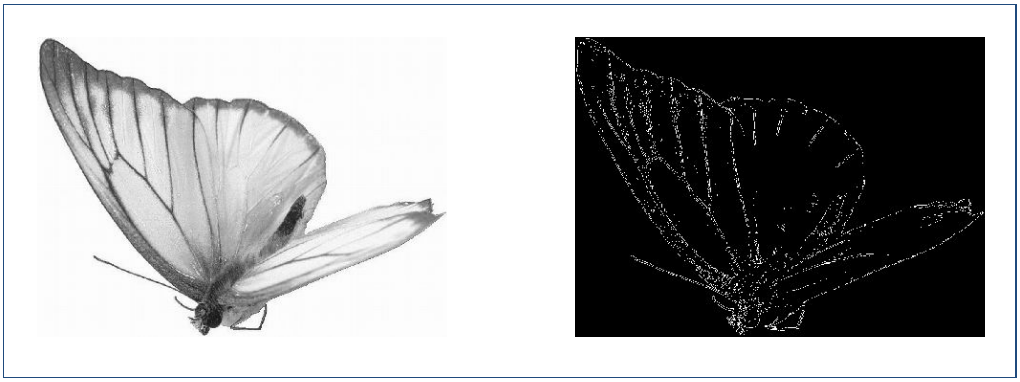


Figure 2.3: Shape extraction for an image with clear edges.

As we can see, since objects in this image have clear and clean boundaries, the shape information is extracted accurately. There are some criteria to build a good Shape-based Image Retrieval system.

- Shape features extracted should be invariant to image rotation, scale and translation.
- Shape features should be able to be extracted from images easily and correctly without being affected by noise.
- The similarity algorithm to compute the distance between two shapes should be similar to human perception. This means, images with more visual similar shapes of objects should have smaller distance between each other.

Shape-based Image Retrieval techniques can be categorized into boundary-based and region-based, according to shape feature representations. Boundary-based shape feature representation uses the outer edges of objects in an image, while region-based feature representation uses the entire shape region. Fourier Descriptor and Moment Invariants are the most famous techniques for these two categories, respectively. Fourier Descriptor [7] uses Fourier transformed edges as shape features. Moment Invariants [8] uses region-based moments as shape features which are invariant to transformations [9].

There is one main problem using shape features in CBIR: it can only deal with images with clear and clean boundaries. Accurate extraction of shape features of images with complex contours and significant noise is still being studied. As we can see from Figure 2.4, since contours of objects in the image are very complex and surrounded by many small objects which can be regarded as noise, extracting edges for that image is very difficult.

Figure 2.4: An image with complex objects.

3. Texture features

"Texture features of images refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity". An image's texture content provides information of image properties such as smoothness, coarseness, and regularity which is useful in a CBIR system.

The general methodology of Texture-based Image Retrieval is described below. Images in a database, as well as query images, are first segmented into regions of same textures, and then a similarity measure is used to compute the distance between pairs of such regions. Finally, the distance between two images is the sum of distances between all pairs of regions with the same texture.

There are two main problems in Texture-based Image Retrieval.

- Texture segmentation is difficult, and the notion of "same" texture is not well defined.
- Image retrieval based on comparing texture segments is usually sensitive to over-segmentation and under-segmentation.

Since there are some problems using only an image's textures as the visual feature to process image retrieval, in a practical CBIR system, texture features are always used in combination with other visual features, such as shapes and colors.

C. Histogram Distance Metrics

As mentioned, color features are the most widely used in current CBIR systems. There are many Color-based Image Retrieval techniques based on color histograms, color moments, and color sets. Most of them attempt to combine color information with other image features, such as spatial information, to improve image retrieval results.

There are a number of metrics for calculating the distance between two images' histograms. As an example, to explain different histogram distance metrics, we assume two images, A and B, are both quantized to 4 colors. Their normalized color histograms are shown below:

$$H^A = \{20\%, 30\%, 10\%, 40\%\}$$

$$H^B = \{10\%, 10\%, 50\%, 30\%\}$$

The simplest way to calculate the distance between the two color histograms is L1 distance. It calculates the absolute value of the difference between the same colors of two histograms and sums all of them as the total distance. As Equation 2.2 shows:

$$d(A,B) = \sum_{j=1}^{n} |H_j^A - H_j^B| \tag{2.2}$$

H_j^A and H_j^B are the normalized values in the colour histogram of colour for image A and image B respectively. $\boldsymbol{n}$ is the total number of colours. So, for our example, the distance between two images, A and B, is:

$$d\ (A, B)=|0.2-0.1|+|0.3-0.1|+|0.1- 0.5|+|0.4-0.3|=0.8$$

The distance between two histograms can also be calculated using L2 distance as Equation 2.3 shows:

$$d(A,B) = \sqrt{\sum_{j=1}^{n} (H_j^A - H_j^B)^2} \quad (2.3)$$

H_j^A , H_j^B , j and n denote the same meaning as in Equation 2. For our example, the distance between the two color histograms using this metric is:

$$d(A,B)=\sqrt{(0.2\text{-}0.1)^2 + (0.3-0.1)^2 + (0.1-0.5)^2 + (0.4-0.3)^2} = 0.47$$

D. CBIR Systems

Since Content-based Image Retrieval has recently been an active research area, many different CBIR systems have been developed. In this section, we select some of the famous CBIR system and describe their characteristics.

Query By Image Content (QBIC) is the first and most and was developed by IBM Almaden Research Center. It is available either in stand-alone form, or as part of other IBM products, such as the DB2 digital library. Visual features used in QBIC include colors, shapes, and textures. The color features used are 3-dimensional vector in RGB, YIQ, Lab and Munsell color model. The shape features consist of shape area, circularity, eccentricity, major axis orientation, and a set of algebraic moment invariants. Coarseness, contrast, and directionality are texture features used in the system. In QBIC, different distance metrics such as histogram-based algorithms are used to calculate the distance between two images. QBIC supports queries based on example images, user-constructed sketches and drawings, selected colors, and texture patterns.

The VIRAGE Image Engine is another CBIR system, developed by VIRAGE Technologies, Inc. It uses colors, shapes, and textures as visual features of images. These visual features can be used separately or combined together to provide a more specific image searching. One of the advantages of using the VIRAGE system is that users can adjust the weights associated with the atomic features according to their own emphasis [9].

The image searching engine, NETRA, was developed by University of California, Santa Barbara [10]. Images are segmented into regions with the same color. For each of these regions, the color, texture, shape and spatial features are extracted as visual features of images. These features can then be combined to search and retrieve similar regions from the database. In other words, this representation allows the user to compose interesting queries such as "retrieve all images that contain regions that have the color of object A, texture of object B, shape of object C, and lie in the upper of the image" [10].

III. Proposed algorithm

A. Overview

Content-based Image Retrieval (CBIR) is currently an active research area in computing science. A CBIR algorithm which was based on image block method that combined both color and edge feature was proposed. In consideration of the main drawback of global histogram representation is dependent of the color without spatial or shape information, a new image block method that divided the image to 8 related blocks which contained more information of the image is utilized to extract image feature. Based on these 8 blocks, histogram equalization and edge detection techniques are also used for image retrieval. The experimental results show that the proposed image block method has better ability of characterizing the image contents than traditional block method and could perform retrieval system efficiently.

The procedure of the proposed algorithm has 7 steps shown in Figure 3.1 below.

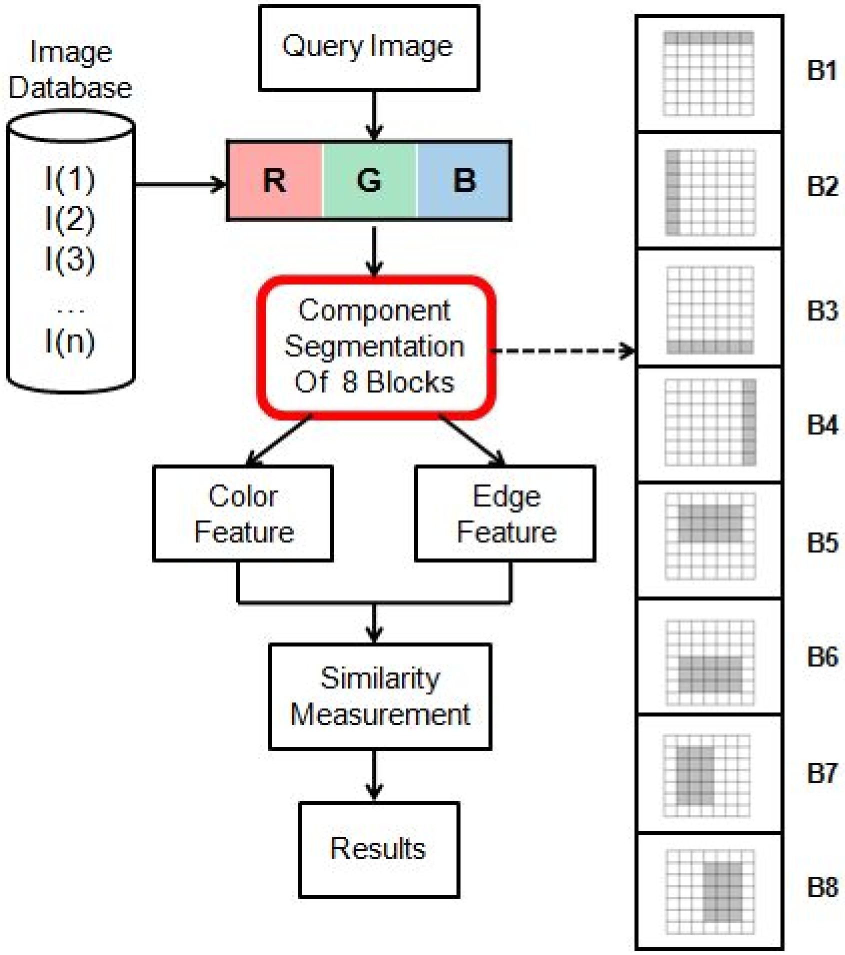


Figure 3.1： Flow chart of the algorithm.

1. Input a query image first.
2. R, G, B components are obtained from the query image.
3. 8 blocks are produced using the proposed block method.
4. After getting image blocks, the color feature is obtained from each block. An explanation will be shown in chapter C.

5. Also from these 8 blocks, the edge feature is obtained. An explanation will be shown in chapter D.
6. We use the extracted image features for image similarity measurement. The feature obtained from the query image is compared with trained database.
7. We can obtain the final retrieval results.

From the procedure of the proposed algorithm, we could notice that feature extraction part is based on color feature and edge feature from image blocks. The text that follows detail explains the proposed image block method and the procedure of the achievement of color feature and edge feature.

B. Proposed image block method

In the Global color histogram, the visual feature of a color image may be represented by different visual feature representations, such as the image's color histogram. The main drawback of global histograms for classification is that the representation is dependent of the color of the object being studied, ignoring its shape and texture. Color histograms can potentially be identical for two images with different object content which happens to share color information [11]. Conversely, without spatial or shape information, similar objects of different color may be indistinguishable based solely on color histogram comparisons.

In the Traditional block method, image block based histogram could amend the disadvantage of global histogram lacking spatial information to some extent. Such method blocked the image in some way then calculate color feature separately. Blocking the image to several parts could help finding interested target part effectively. Malki J and Dimai [12][13] proposed a regular image block method in Figure 3.2. It could improve the discriminatory performance obviously.

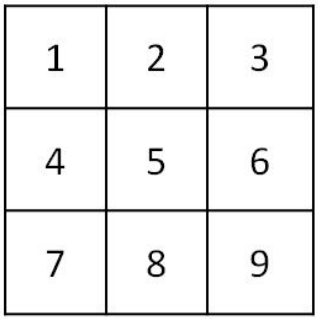


Figure 3.2: Traditional block method.

Since the position of the target object always in the middle of the image, in this paper, the image is divided by 7x7 blocks (49 sub-blocks of same size). The proposed method has 8 blocks by combining some of the sub-blocks as shown in Figure 3.3.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

B1: 1,2,3,4,5,6,7
B2: 1, 8, 15, 22,29,36,43
B3: 43,44,45,46,47,48,49
B4: 7,14,21,28,35,42,49
B5: 9,10,11,12,13,16,17,18,19,20,23,24,25,26,27
B6: 23,24,25,26,27,30,31,32,33,34,37,38,39,40,41
B7: 9,16,23,30,37,10,17,24,31,38,11,18,25,32,39
B8: 11,18,25,32,39,12,19,26,33,40,13,20,27,34,41

Figure 3.3: Proposed image block method.

An RGB image is used just as an example as shown in Figure3.4. Firstly convert the RGB image to gray scale and get R component, G component and B component as shown in Figure 3.6. Then 8 blocks are obtained using the proposed image block method for each component. Figure 3.7 shows the 8 blocks obtained using the proposed method of R component and the histograms of each block.

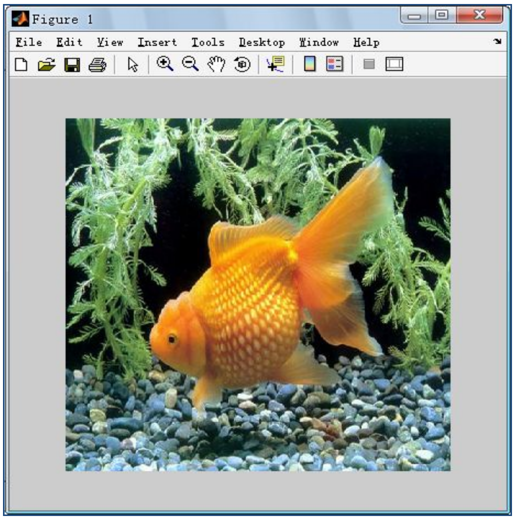


Figure 3.4: RGB image.

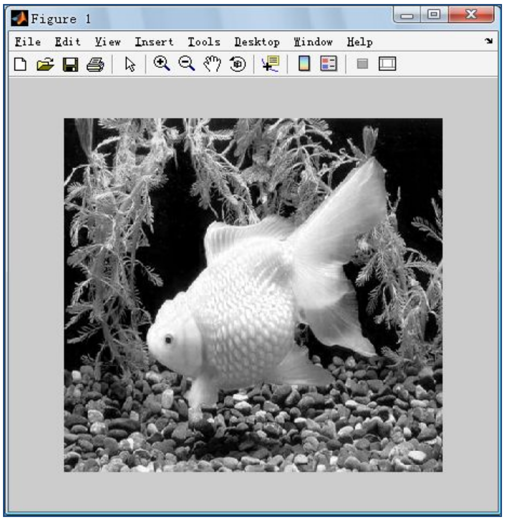


Figure 3.5: Gray scale image.

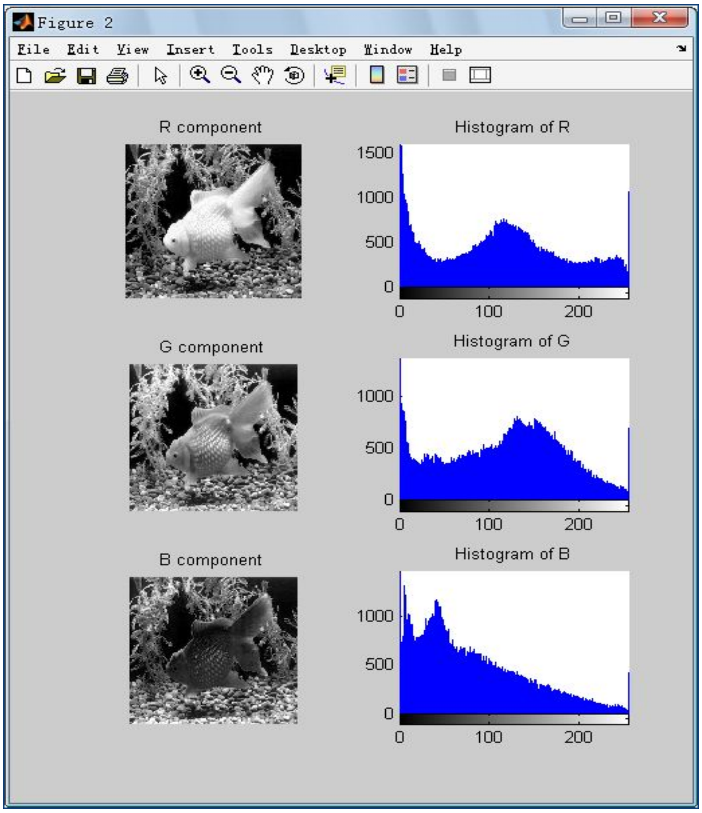


Figure 3.6: Three components of RGB image.

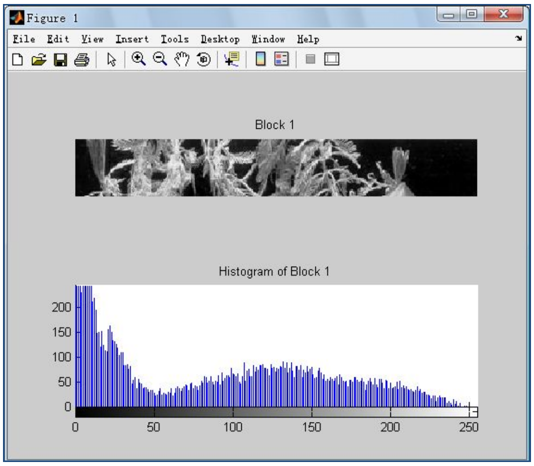


Figure 1
File Edit View Insert Tools Desktop Window Help
Block 1
Histogram of Block 1
200
150
100
50
0
0
50
100
150
200
250

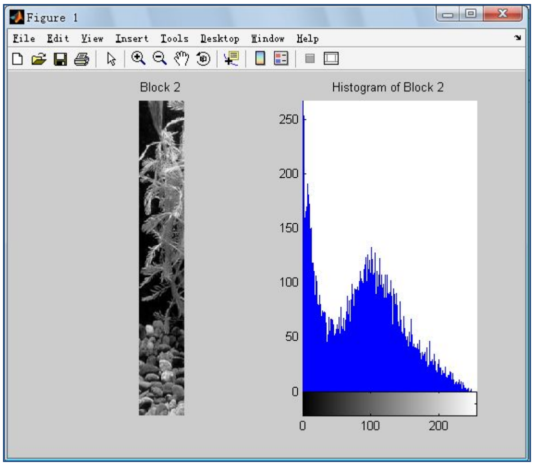


Figure 1
File Edit View Insert Tools Desktop Window Help
Block 2
Histogram of Block 2
250
200
150
100
50
0
0
100
200

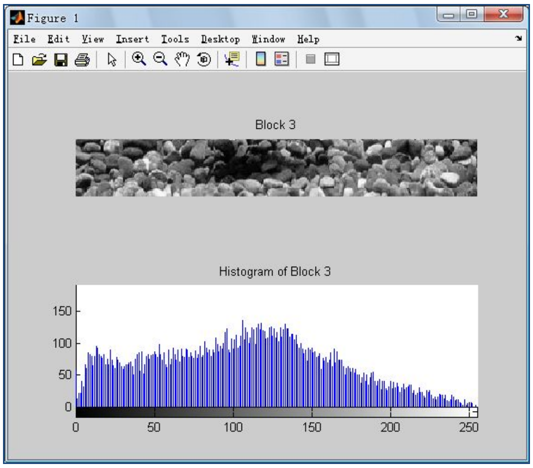


Figure 1
File Edit View Insert Tools Desktop Window Help
Block 3
Histogram of Block 3
150
100
50
0
0
50
100
150
200
250

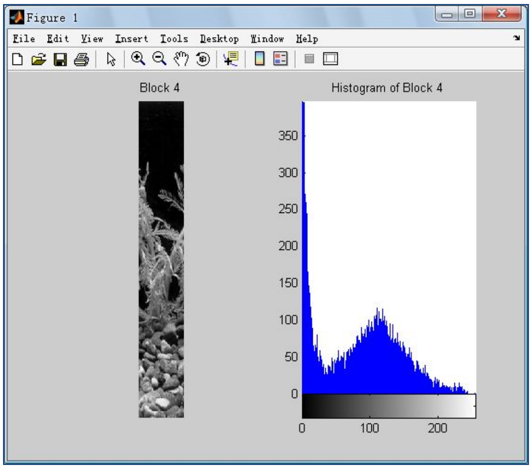


Figure 1
File Edit View Insert Tools Desktop Window Help
Block 4
Histogram of Block 4
350
300
250
200
150
100
50
0
0
100
200

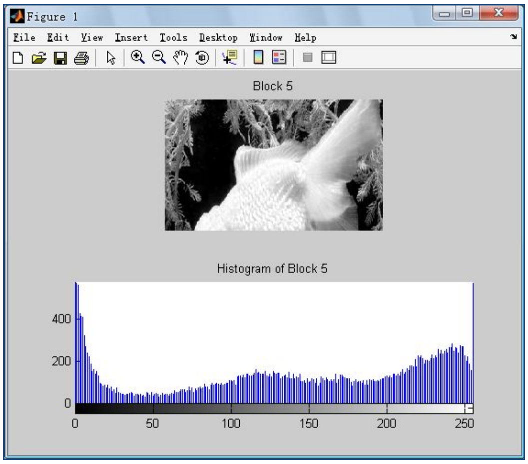


Figure 1
File Edit View Insert Tools Desktop Window Help
Block 5
Histogram of Block 5
400
200
0
0
50
100
150
200
250

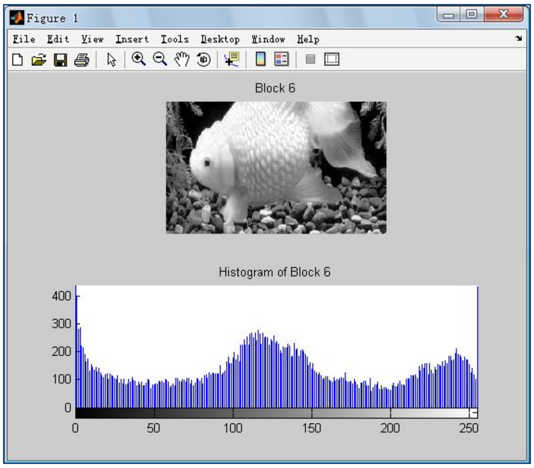


Figure 1
File Edit View Insert Tools Desktop Window Help
Block 6
Histogram of Block 6
400
300
200
100
0
0
50
100
150
200
250

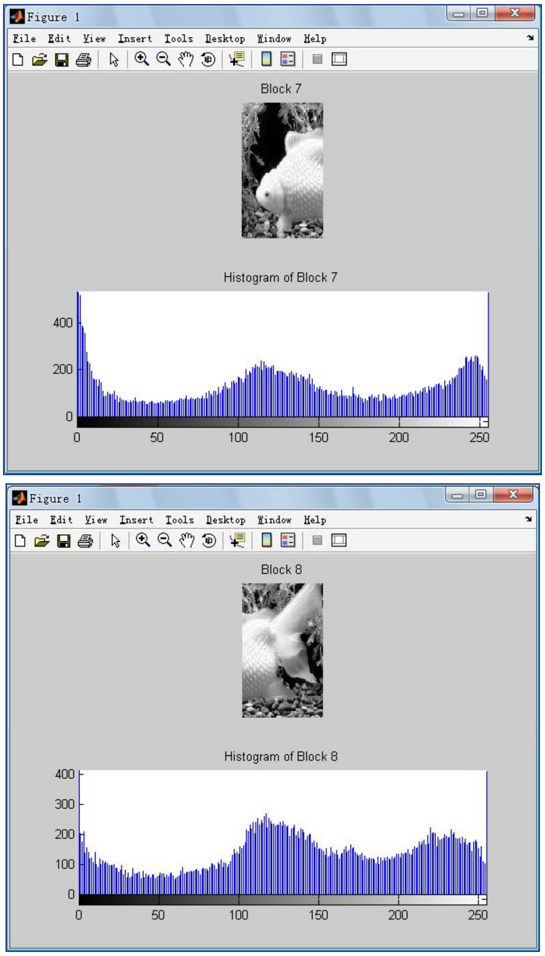


Figure 3.7: Image blocks and histogram of each block.

From Figure 3.7 we can notice that the middle part of the image is divided to 4 related blocks. B1, B2, B3 and B4 indicate the background of the image. Compared with the traditional block method, these 8 blocks contain more information with both color and spatial distribution feature.

C. Color feature extraction

1. Grayscale image and histogram equalization

In photography and computing, a grayscale or greyscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. 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 [14].

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.

The intensity of a pixel is expressed within a given range between a minimum and a maximum, inclusive. This range is represented in an abstract way as a range from 0 (total absence, black) and 1 (total presence, white), with any fractional values in between. Figure 3.8 shows the adjacent image specifies 256 different shades of gray in a 16 x 16 grid.

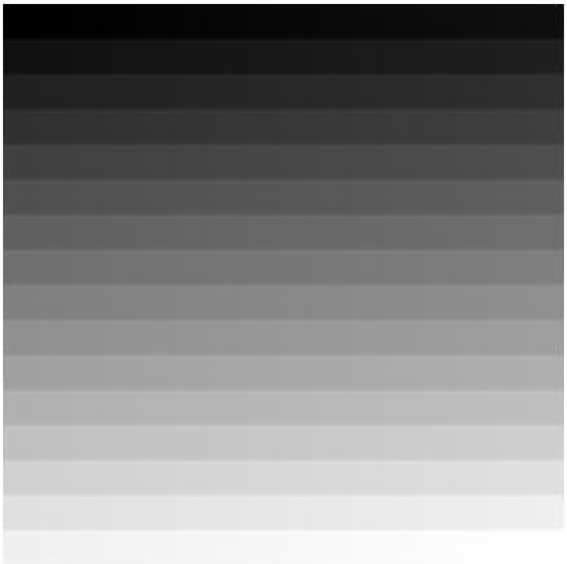


Figure 3.8: The adjacent image specifies 256 different shades of gray in a 16 x 16 grid.

Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images. Here is an example of color channel splitting of a full RGB color image as shown in Figure 3.9.

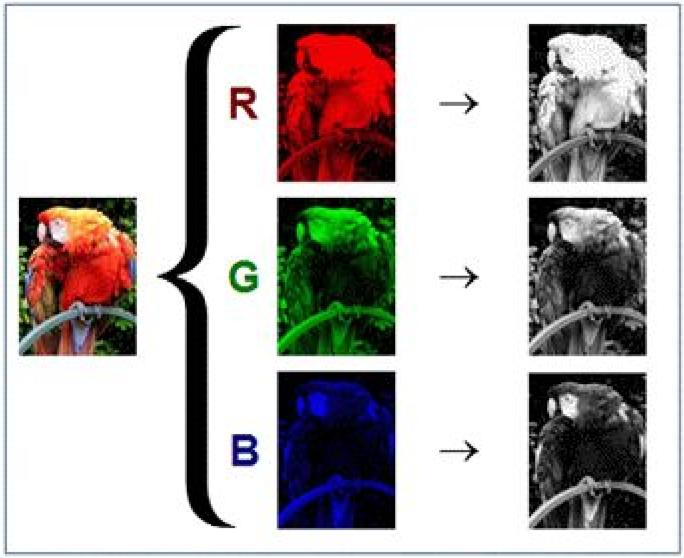


Figure 3.9: Color channel splitting of a full RGB color image.

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast without affecting the global contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values [15] [16].

Assume for a moment that intensity levels are continuous quantities normalized to the range [0,1], and let $p_r(r)$ denote the probability density function (PDF) of the intensity levels in a given image, where the subscript is used for that we perform the following transformation on the input levels to obtain output (processed) intensity levels, s,

$$s = T(r) = \int_0^r p_r(\omega)d\omega \tag{3.1}$$

Where w is a dummy variable of integration. It can be shown that the probability density function of the output levels is uniform; that is,

$$p_s(s) = \begin{cases} 1 & for \quad 0 \le s \le 1 \\ 0 & otherwise \end{cases} \tag{3.2}$$

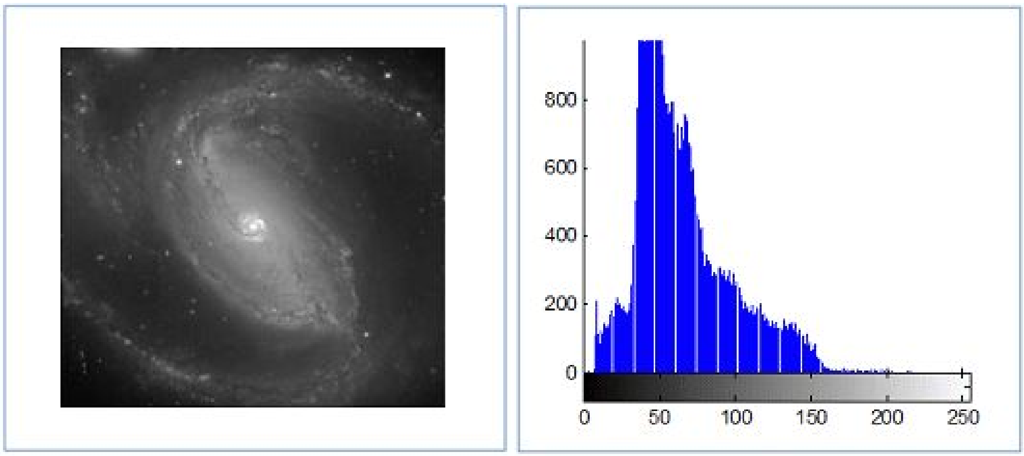
In other words, the preceding transformation generates an image whose intensity levels is equally likely, and, in addition, covers the entire range [0, 1]. The net result of this intensity-level equalization process is an image with increased dynamic range, which will tend to have higher contrast. Note that the transformation function is really nothing more than the cumulative distribution function （CDF）.

When dealing with discrete quantities we work with histograms and call the preceding technique histogram equalization, although, in general, the histogram of the processed image will not be uniform, due to the discrete nature of the variables. Let p_r (r_j), j=1, 2, …, L, denote the histogram associated with the intensity levels of a given image, and recall that the values in a normalized histogram are approximations to the probability of occurrence of each intensity level in the image. For discrete quantities we work with summations, and the equalization transformation becomes

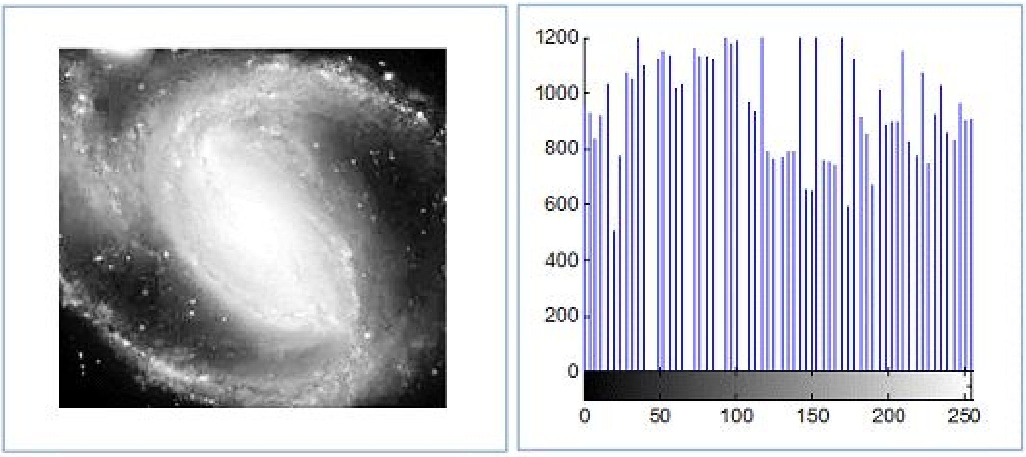
$$
\begin{aligned}
s_k &= T(r_k) \\
&= \sum_{j=1}^{k} p_r(r_j) \\
&= \sum_{j=1}^{k} \frac{n_j}{n}
\end{aligned}
\tag{3.3}
$$

for $k=1,2,\ldots,L$, where s_k is the intensity value in the output (processed) Image corresponding to value r_k in the input image.

The above describes histogram equalization on a grey scale image. However it can also be used on color images by applying the same method separately to the Red, Green and Blue components of the RGB color values of the image. Still, it should be noted that applying the same method on the Red, Green, and Blue components of an RGB image may yield dramatic changes in the image's color balance since the relative distributions of the color channels change as a result of applying the algorithm. Figure 3.10 shows an example of histogram equalization.



(a) (b)

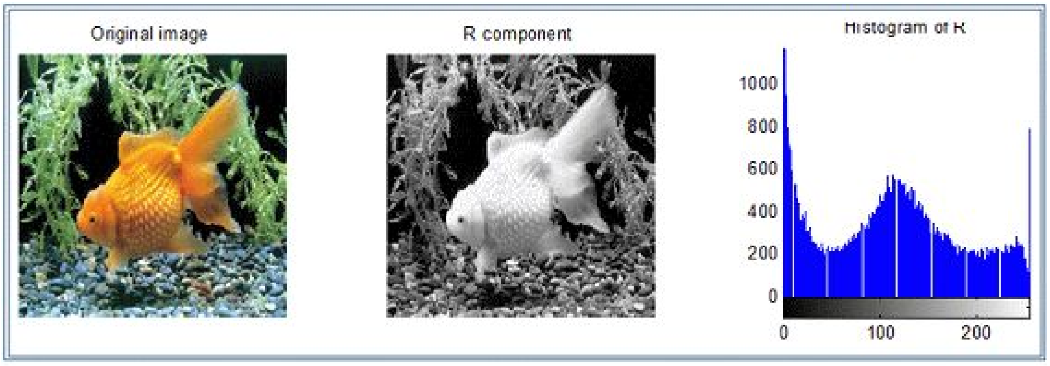


(c) (d)

Figure 3.10: An example of histogram equalization
(a) original grayscale image. (b) histogram of a.
(c) histogram equalization of a. (d) histogram of c.

2. Procedure of color feature extraction

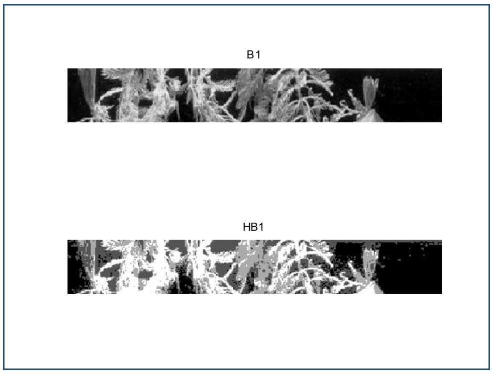
After three components of R, G and B are obtained from the color image, getting 8 blocks using the proposed image block method for each component. The color feature is obtained from each block. Figure 3.11 shows the image of R component and histogram of an original color image. And Figure 3.12 shows the 8 blocks obtained using the proposed method and equalized image of these blocks.

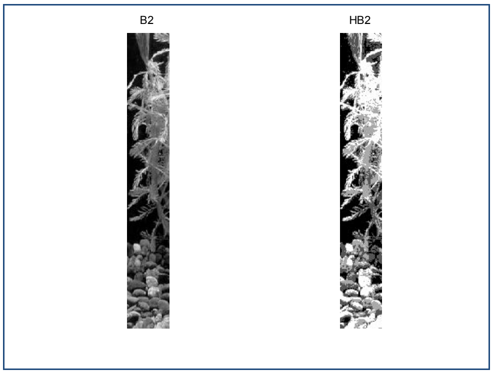


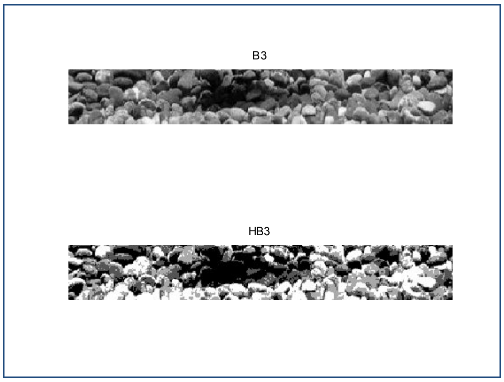
(a) (b) (c)

Figure 3.11: R component and histogram

(a) Original image. (b) Image of R component. (c) Histogram of R component.







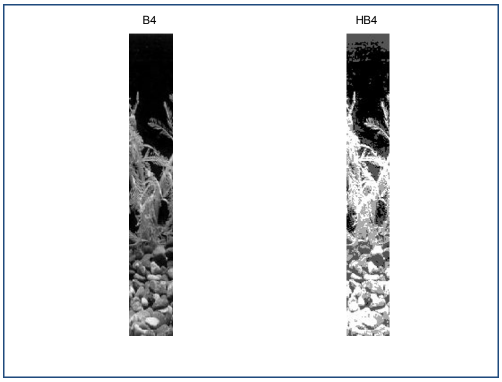


Figure 3.12: (B1~B8) Image blocks
(HB1~HB8) Histogram equalized image blocks.

After component segmentation of 8 blocks, the color feature is extracted by 3 steps:

1. Get the histogram of each block;
2. Histogram equalization of 4 gray-level for each block;
3. Calculate the statistical information of the gray values in every gray-level.

There are 4 gray-level. That is 0, 85, 170 and 255. Table 3.1 shows the number of the gray values in every gray-level of each block.

Table 3.1: Number of the gray values in every gray-level of each block.

Block \ Num \ Gray Value	0	85	170	255
1	4416	4340	4366	4378
2	4350	4427	4309	4414
3	4352	4355	4452	4341
4	4401	4336	4354	4409
5	9385	9387	9419	9309
6	9390	9257	9488	9365
7	9365	9326	9433	9376
8	9438	9316	9415	9331

D. Edge feature extraction

Edge detection is a terminology in image processing and computer vision, particularly in the areas of feature detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities.

The edge detection methods that have been published mainly differ in the types of smoothing filters that are applied and the way the measures of edge strength are computed. As many edge detection methods rely on the computation of image gradients, they also differ in the types of filters used for computing gradient estimates in the x- and y-directions.

1. Canny edge detection

The Canny edge detection operator was developed by John F. Canny in 1986 and uses a multi-stage algorithm to detect a wide range of edges in images. Most importantly, Canny also produced a computational theory of edge detection explaining why the technique works.

In order to implement the canny edge detector algorithm, a series of steps must be followed.

Step 1

The first step is to filter out any noise in the original image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask used in my implementation is shown as Figure 3.13.

$$\frac{1}{115}\begin{array}{|c|c|c|c|c|}\hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline \end{array}$$

Figure 3.13: Discrete approximation to Gaussian function with σ=1.4

Step 2

After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown as Figure 3.14.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure 3.14: Masks of Sobel operator

The magnitude, or EDGE STRENGTH, of the gradient is then approximated using the formula:

$$|G| = |Gx| + |Gy| \tag{3. 4}$$

Step 3

Finding the edge direction is trivial once the gradient in the x and y directions are known. However, you will generate an error whenever sumX is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If Gy has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is just:

$$\text{theta} = \text{invtan}\,(Gy / Gx) \tag{3. 5}$$

Step 4

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5x5 image are aligned as follows.

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

Then, it can be seen by looking at pixel "a", there are only four possible directions when describing the surrounding pixels - 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal), as shown in Figure 3.15. So now the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions.

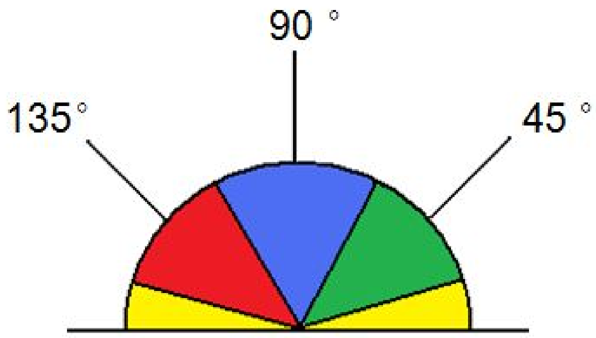


Figure 3.15: Possible directions

Therefore, any edge direction falling within the yellow range (0 to 22.5 & 157.5 to 180 degrees) is set to 0 degrees. Any edge direction falling in the green range (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling in the blue range (67.5 to 112.5 degrees) is set to 90 degrees. And finally, any edge direction falling within the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Step 5

After the edge directions are known, nonmaximum suppression now has to be applied. Nonmaximum suppression is used to trace along the edge in the edge direction and suppress any pixel value (sets it equal to 0) that is not considered to be an edge. This will give a thin line in the output image.

Step 6

Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, T1 is applied to an image, and an edge has an average strength equal to T1, then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you don't stop till you hit a gradient below T1.

2. Procedure of edge feature extraction

After three components of R, G and B are obtained from the color image, getting 8 blocks using the proposed image block method for each component. The edge feature is obtained from each block. The proposed algorithm utilized canny edge detection method.

The edge feature is extracted as below:

1. Edge detection using ‘canny’ operator of each block [17].
2. Get the binary edge image of each block;

3. Calculate the total number of edge in each block.

Figure 3.16 shows the 8 blocks obtained using the proposed method and edge image of these blocks.

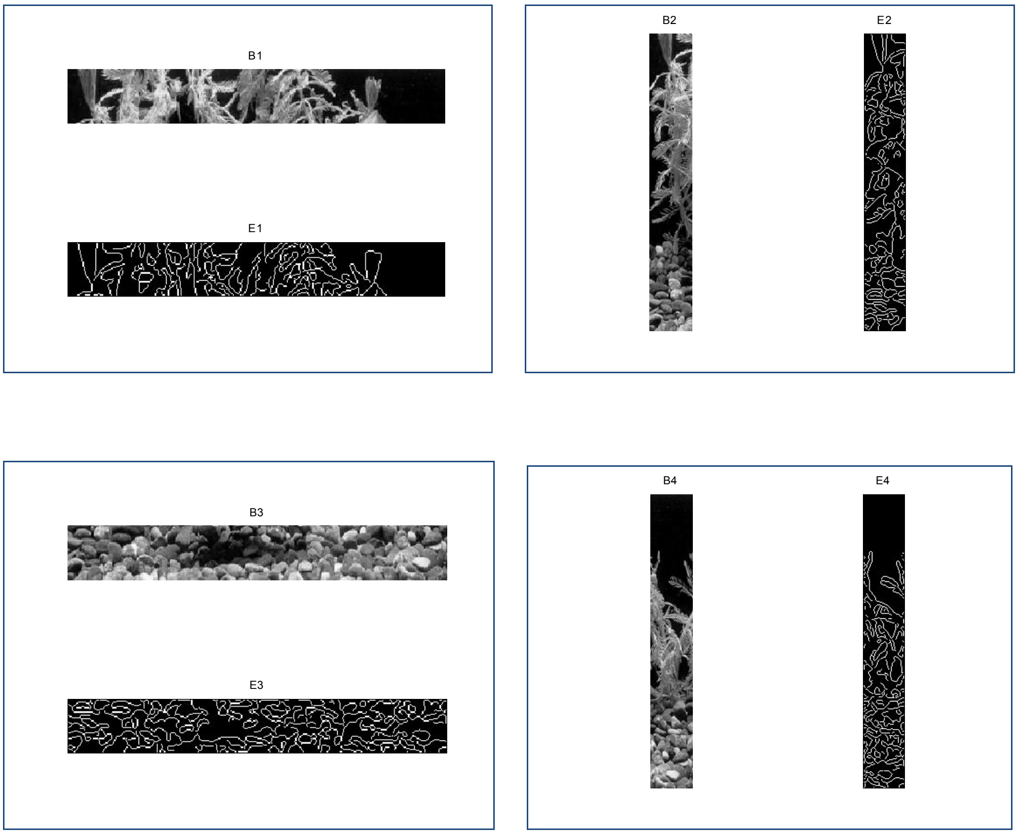


Figure 3.16: Edge detection use 'canny'

(B1~B8) Image blocks

(E1~E8) Edge image of blocks

E. Similarity measure

After having extracted features, our next task is to find a similarity measure. Usually Euclidean distance, histogram intersection, or cosine or quadratic distances are used for the calculation of the images' similarity ratio. Any of these values does not reflect the similarity rate of two images in itself. It is useful only with comparison to other similar values. This is the reason that all the practical implementations of content-based image retrieval must complete computation of all images from the database.

The simplest way to calculate the distance between the two color histograms is L1 distance [18]. It calculates the absolute value of the difference between two features. The proposed algorithm utilized Euclidean distance for calculation of the images' similarity ratio. As an example, to explain different distance metrics, we assume two images, A and B, H_j^A and H_j^B are the normalized feature values for image A and image B respectively.

$$d(A,B) = \sum_{j=1}^{n} | H_j^A - H_j^B | \tag{3. 6}$$

IV. Simulation and results

The validation of the algorithms proposed is verified through the computer simulation. Figure 4.1 shows the universal CBIR system. It contained two parts: Creation of image database and image retrieval.

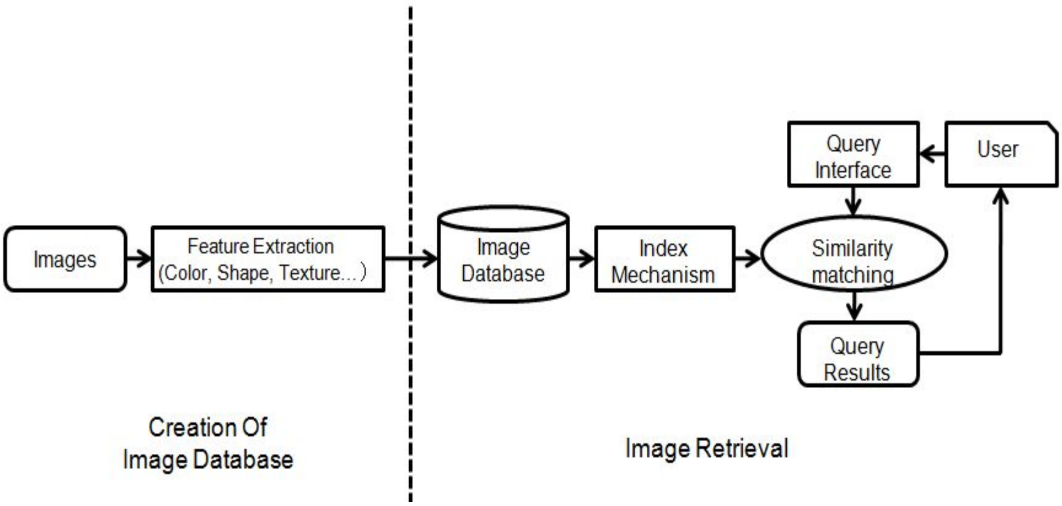


Figure 4.1: Image retrieval system.

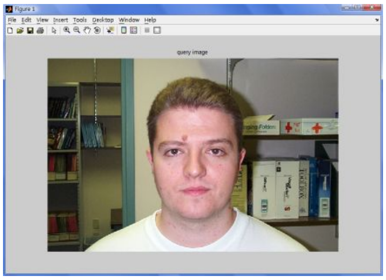
In the creation of image database part, images are acquired from a collection one after another. And then feature extraction process is applied to them using such algorithm. Any of the values does not reflect the similarity rate of two images in itself. It is useful only with comparison to other similar values. This is the reason that all the practical implementations of content-based image retrieval must complete computation of all images from the database. At last the extracted feature vector is saved in a database against the image name under consideration.

After database creation, the image retrieval system could start by user specified query image. This is called Query By Example (QBE). It is one of the most popular methodologies used in CBIR systems, in which images are selected from an image database similar to a given image presented by users. The feature extraction process runs on the query image and extracts the feature information in the form of a vector. And this vector is used to calculate similarity of query image against the feature vectors saved in the feature database of the image collection.

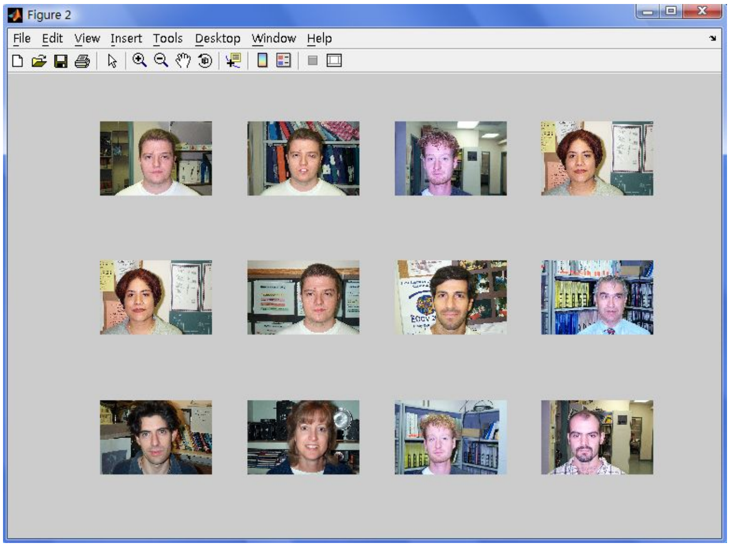
The simulation is achieved by MATLAB 7.0. The proposed algorithm is tested using the image database downloaded and selected from image database of Washington State University. The database contained 300 images of 3 different classes:

- Human face
- Airplane
- Nature scenery

The following figures show some of the simulation results that generated the closest results using proposed algorithm:

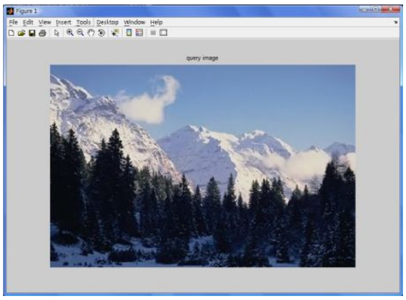


(a)

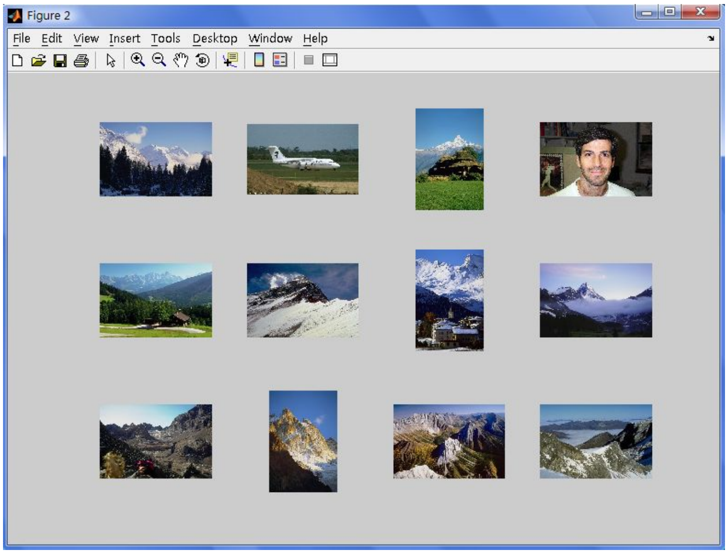


(b)

Figure 4. 2: Simulation result of human face
(a) Query image. (b) Query results.



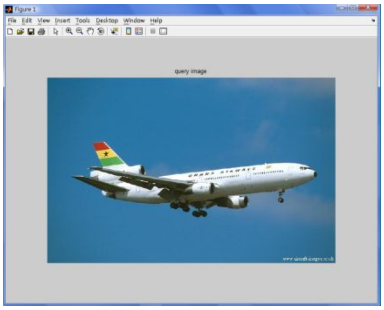
(a)



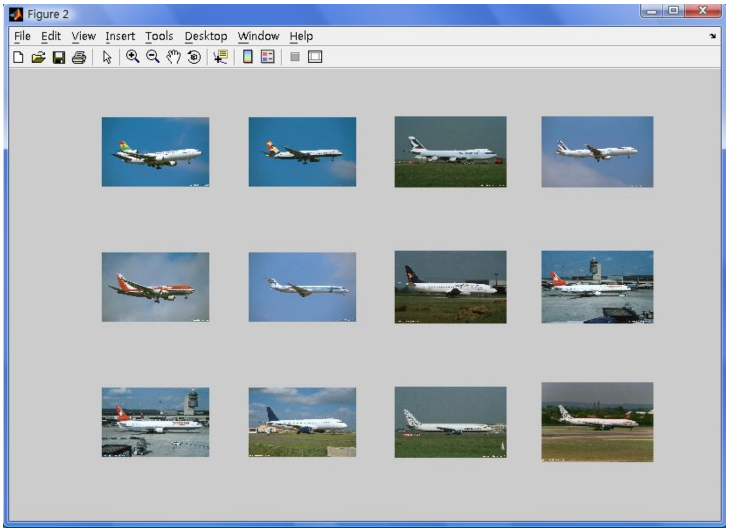
(b)

Figure 4. 3: Simulation result of nature scenery

(a) Query image. (b) Query results.



(a)

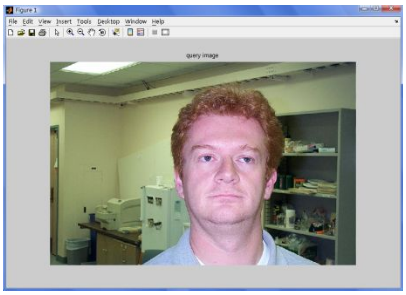


(b)

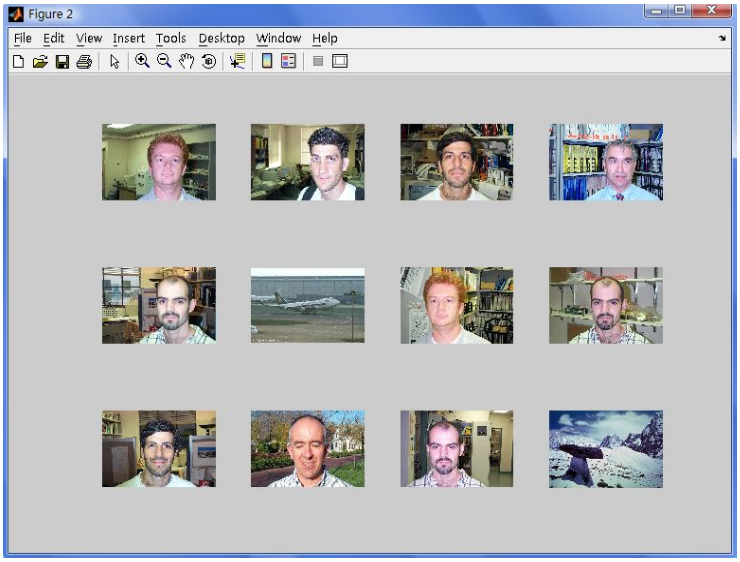
Figure 4. 4: Simulation result of airplane

(a) Query image. (b) Query results.

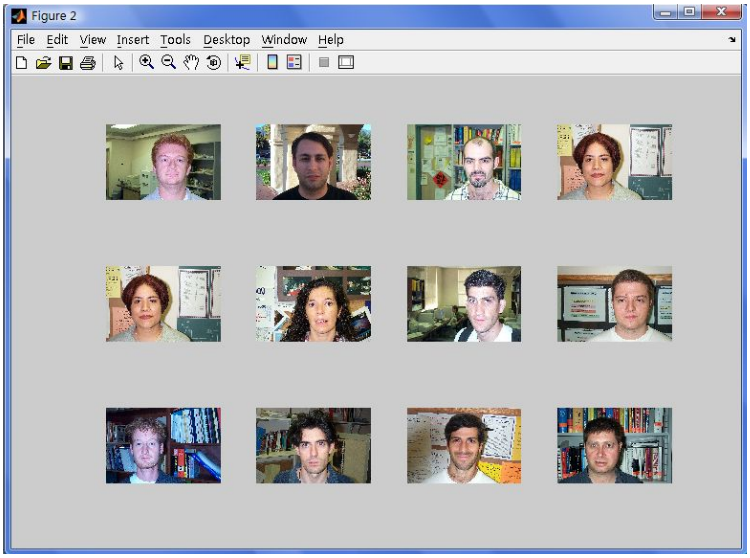
For comparing the effectively and search precision of the algorithm, we use traditional block method for feature extraction and generate retrieval results using the same database. Figure 4. 5 shows the simulation results of human face's case based on the same query image using two algorithms.



(a)



(b)



(c)

Figure 4. 5: Simulation results comparation

(a) Query image. (b) Simulation result of human face using traditional block method.

(c) Simulation result of human face using proposed method.

The following Table 4.1 shows the precision comparison using two algorithms based on different outputs. The precision is the average value of every tested image in the database.

Table 4.1: Precision comparison

	Method	10 outputs	30 outputs
Human face	Traditional Block Method	0.80	0.68
	Proposed Method	0.92	0.76
Air Plane	Traditional Block Method	0.57	0.50
	Proposed Method	0.75	0.67
Nature Scenery	Traditional Block Method	0.58	0.52
	Proposed Method	0.69	0.61

In the experimental results, we are able to notice that the proposed image block method have better ability of characterizing the image contents than traditional block method, especially for the images that the target object located in the middle part of the image.

I. Conclusion

A content based image retrieval algorithm which was based on image block method that combined both color and edge feature was proposed. In consideration of the main drawback of global histogram's representation is dependent of the color without spatial or shape information, a new image block method that divided the image to 8 related blocks which contained more spatial information is utilized to extract image feature. Based on these 8 blocks, histogram equalization and edge detection techniques are also used for image retrieval.

From the experiment results of the proposed algorithm that verified through the computer simulation, we could notice that the method which put emphasis on the middle part of the image is more consistent with human subjective vision. And the proposed image block method could represent the information of image more efficiently. From the precision of results we could also see that this method is more suited for searching those images with target object in the middle of the image.

References

1. R. N. Bracewell, “Two-Dimensional Imaging”, 1995.

2. Pedram Azad, Tilo Gockel, and Rüdiger Dillmann, “Computer Vision-Principles and Practice”, 2008.

3. Michael Lew, “Content-based Multimedia Information Retrieval: State of the Art and Challenges”, ACM Transactions on Multimedia Computing, Communications, and Applications, pp. 1-19, 2006.

4. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glassman, D. Petkovic, and P. Yanker, “The QBIC project: Querying image by content using colour, texture and shape”, SPIE, pp.115-132, Feb. 1993.

5. P.G.B. Enser, “Progress in documentation: Pictorial information retrieval”, Journal of Documentation, pp.126-170, 1995.

6. A. D. Bimbo, “Visual Information Retrieval”, Morgan Kaufmann Publishers, 1999.

7. Y. Rui, A. She, and T. Huang, "Modified fourier descriptors for shape representation-a practical approach”, In Proceedings of the First International Workshop on Image Databases and Multi Media Search, pp. 22-23, Aug. 1996.

8. L. Yang and F. Albregtsen, “Fast computation of invariant geometric moments: A new method giving correct results”, In Proceeding of the International Conference on Pattern Recognition (ICPR) 94, pp.201-204, 1994.

9. Y. Rui, T. S. Huang, and S. F. Chang, “Image retrieval: Past present and future”, Journal of Visual Communication and Image Representation, pp.1-23, 1999.

10. W. Y. Ma and B. S. Manjunath, “A toolbox for navigating large image databases”. Multimedia systems, pp.184–198, 1999.

11. Bashkov, E.A., Kostyukova, N.S., “Effectiveness estimation of image retrieval by 2D color histogram”, Journal of Automation and Information Sciences, 2006.

12. DIMAI A, “Spatial encoding using differences of global features”, SPIE, pp.352-360, 1997.

13. MALKI J., BOUJEMAA N., and NASTER C., “Region queries without segmentation for image retrieval by content”, Proceeding of the Third International Conference on Visual Information and Information Systems, London, UK: SpingePVerlag, pp.115-122, 1997.

14. Stephen Johnson, “Stephen Johnson on Digital Photography”, 2006.

15. Acharya and Ray, “Image Processing: Principles and Applications”, Wiley-Interscience, ISBN 0-471-71998-6, 2005.

16. Russ, “The Image Processing Handbook: Fourth Edition”, CRC, ISBN 0-8493-2532-3, 2002 .

17. Lan-Yan Xue and Jian-Jia Pan, “Edge detection combining wavelet transform and canny operator based on fusion rules”, Wavelet Analysis and Pattern Recognition, ICWAPR 2009, International Conference on 12-15, pp.324-328, Jul. 2009.

18. de A.T. de Carvalho F., Pimentel J.T., and Bezerra, L.X.T, “Clustering of symbolic interval data based on a single adaptive L1 distance”, Neural Networks, IJCNN 2007, International Joint Conference on 12-17, pp.224-229, Aug. 2007.