Interactive Imaging via Hand Gesture Recognition

Jia JIA

A thesis submitted for the Degree of

Master of Philosophy

Department of Electronic Imaging and Media Communications

School of Informatics

University of Bradford

May 2009

Abstract

With the growth of computer power, Digital Image Processing plays a more and more important role in the modern world, including the field of industry, medical, communications, spaceflight technology etc. As a sub-field, Interactive Image Processing emphasizes particularly on the communications between machine and human. The basic flowchart is definition of object, analysis and training phase, recognition and feedback. Generally speaking, the core issue is how we define the interesting object and track them more accurately in order to complete the interaction process successfully.

This thesis proposes a novel dynamic simulation scheme for interactive image processing. The work consists of two main parts: Hand Motion Detection and Hand Gesture recognition. Within a hand motion detection processing, movement of hand will be identified and extracted. In a specific detection period, the current image is compared with the previous image in order to generate the difference between them. If the generated difference exceeds predefined threshold alarm, a typical hand motion movement is detected. Furthermore, in some particular situations, changes of hand gesture are also desired to be detected and classified. This task requires features extraction and feature comparison among each type of gestures. The essentials of hand gesture are including some low level features such as color, shape etc. Another important feature is orientation histogram. Each type of hand gestures has its particular representation in the domain of orientation histogram. Because Gaussian Mixture Model has great advantages to represent the object with essential feature elements and the Expectation-Maximization is the efficient procedure to compute the maximum likelihood between testing images and predefined standard sample of each different gesture, the comparability between testing image and samples of each type of gestures will be estimated by Expectation-Maximization algorithm in Gaussian Mixture Model. The performance of this approach in experiments shows the proposed method works well and accurately.

Keywords: interactive imaging, hand gesture recognition, feature extraction, segmentation, motion detection, Gaussian Mixture model, Expectation Maximization algorithm.

Acknowledgement

The research work presented in this thesis has been carried out at the department of EIMC, University of Bradford. The path towards to this thesis spans two years with a great progress I have obtained. The author acknowledges his debt to those who have helped along the way and influenced the formation of the understanding and representation of the approaches presented in this thesis. Without them, the way would be much more rugged.

In particular, I would express my gratitude to my supervisor, Professor Jiangmin Jiang, for his continued encouragement and invaluable guidance during this work. From the elaborate education of Professor Jiang, I made a great progress in the field of Digital Imaging Science field and the methodology of research activity. These are precious treasure for me and will benefit myself in the future.

Furthermore, I am also deeply indebted to Dr Ren. He has shared the essential programming environment and his experiences. Dr Ren also provided lots of help and suggestions to resolve many issues within the segmentation-based algorithm design and the revision part of formal thesis contribution. I would also like to extend my thanks to all the colleagues in Digital Imaging Research Group for their generous helps and concerns.

At last, I would give my deep thanks to my parents, my family. Their encouragement, love and never-ending support are the most important and powerful source of energy.

Jia Jia in May, 2009

TABLE OF CONTENTS

List of Figures

List of Table

List of Abbreviations

CBIR	Content-based image retrieval
GMM	Gaussian Mixture Model
EM	Expectation Maximization algorithm
MPEG	Moving Picture Experts Group
RF	Relevance Feedback
BDA	biased discriminant analysis
RGB	Red Green and Blue
bpp	bits per pixel
1 D/2 D/3 D	One-Dimensional/Two-Dimensional/Three-Dimensio
HSV	Hue, Saturation and Value
CMY	Cyan, Magenta, and Yellow
CCV	color coherence vectors
SPCA	shift-invariant principal component analysis
QBIC	query by image and video content
DC	Discrete Cosine
HCI	human computer interaction
HMM	Hidden Markov Model
ZMs	Zernike Moments
FFT	Fast Fourier Transform
PDF	probability density function
ML	Maximum Likelihood

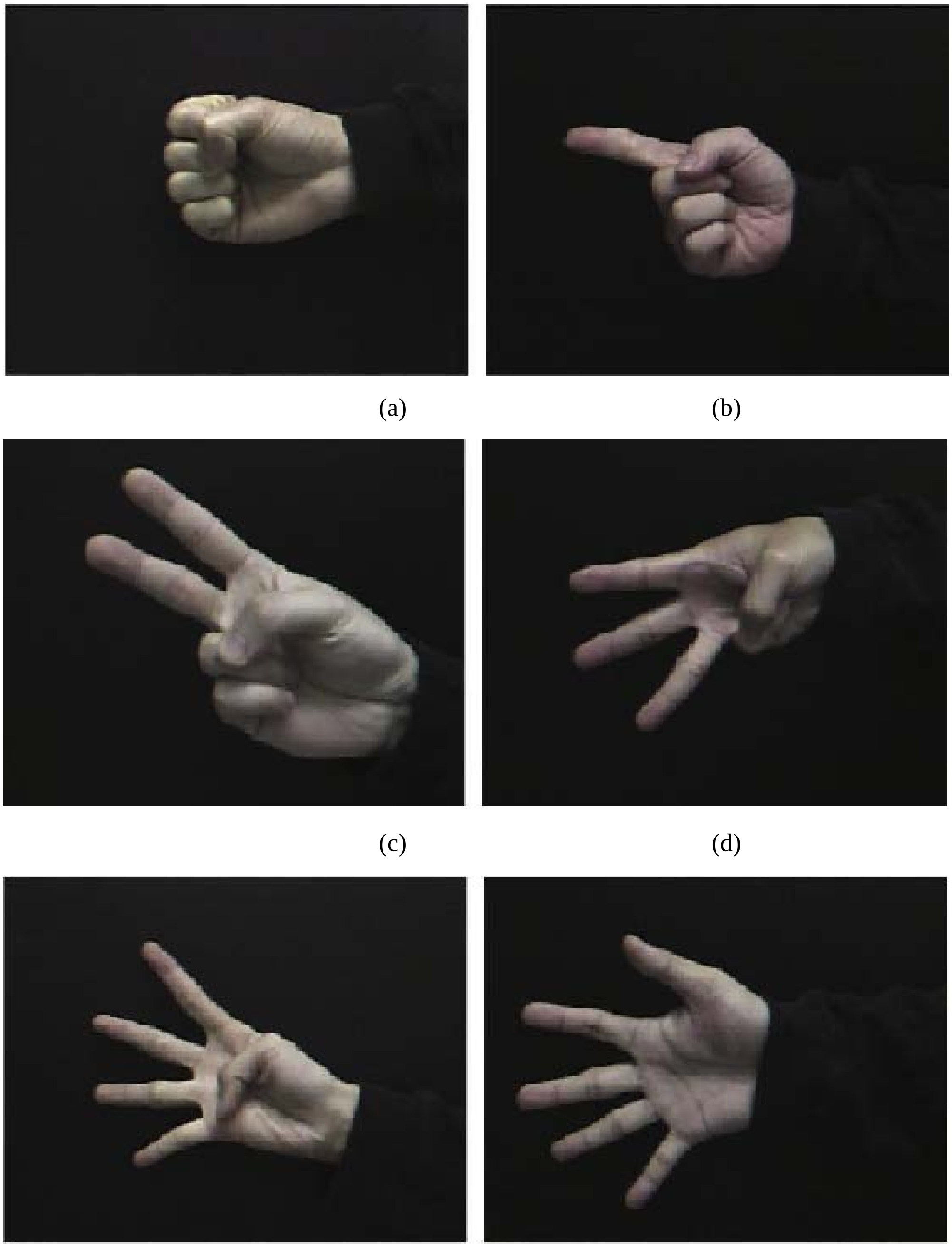
Chapter 1

Introduction

1.1 Introduction to Hand Gesture Recognition

Human hand gestures, for example as shown in Fig 1.1, have their specific meanings and are widely used for communications between deaf people. Actually, gesturing is so deeply rooted in communications that people often continue gesturing when speaking. Recently, hand gesture recognition has gained a lot of interests, which plays a crucial role in a wide range of applications including automatic sign language understanding, entertainment, and human computer interaction (HCI). Because hand gestures are natural and intuitive in providing rich information to computers without extra cumbersome devices, they can offer a great potential for next generation user interfaces, being especially suitable for large scale displays, 3D volumetric displays or wearable devices.

For human–computer interaction, vision-based recognition of hand gestures can provide a natural and modest solution [1]. To achieve this, three steps are required, including (1) analyzing signals acquired by imaging sensors such as video, infrared or ultrasonic; (2) inferring the geometry and motion of the hand; (3) mapping to a set of predefined gestures. An important potential application of this technology is to develop advanced interfaces for the interaction with virtual objects. These objects can be images on a computer screen and the user can manipulate the objects by moving his/her hand and performing actions like ‘‘grasping’’ and ‘‘releasing’’. Using gesture recognition, the user actions on the virtual object will be reproduced by the computer and the operational result is shown in the graphical interface so as to provide feedback to the user. Another important application is to provide computing devices that can interpret gestures from the sign-language alphabet and aid natural interaction of hearing impaired people [2].



(a) (b)

(c) (d)

Figure 1.1 Hand Gestures

Recently, there has been a growing interest in gesture-recognition systems, and quite a few novel approaches have been provided since 1990. Two main challenges here are summarized below. Firstly, the system needs to be personal independent where it must be able to deal with geometric distortions (due to different hand anatomy or different performance of gestures by different persons). Second, the system needs to cope with complex and cluttered background, where segmentation of the gesturing hand becomes difficult.

In [3], Gaussian Mixture Model (GMM) is adopted to detect the human body's gestures and Hidden Markov Model (HMM) is then applied for tracking. The key features are polygonal vertices extracted from body shapes, and the final accuracy is close to 98%. In [4], shape and depth information is integrated for robust hand tracking and the recognition rate is between 70% and 92% under various numbers of samples. As the primary measurement, shape builds an important function describing areas of state-space and contains critical information about the posture. In [5], a hand gesture recognition algorithm is presented based on input/output HMMs, which achieves a recognition rate between 90% and 100%.

In the following, features extracted for hand gesture recognition are analyzed by researchers. From [6], the Zernike Moments (ZMs) [7] of hand silhouettes is proposed for gesture estimation and recognition. Although this method effectively separates the rough posture estimate, it lacks reliable local support as ZMs are globally computed. In [8], a static hand gesture recognition system is presented based on the flex angles of the 10 fingers using a composite neural network. It is efficient and elegant, but the control flowchart is too complicated and also a training phase is needed to extract the classification rules. In [9], a neural network is employed for static gesture recognition where 2D plane cells are used as features. The network performs well, though it costs too much in extracting entire plane cells.

In [10], a view-independent hand gesture recognition approach is presented, enabling natural interaction in virtual environments, in which hand gestures are represented using Fourier descriptors. Although being easy to extend FFT from 1D to 2D, it would be inaccurate for recovering the transformed parameters and corresponding points on the initial contours. In [11], a bottom-up algorithm is presented for static hand gesture recognition using local orientation histogram features. Although having higher recognition accuracy and faster speed, the features used are found sensible to rotations. In [12], a segment-based algorithm for hand gesture estimation is presented, where shape information is utilized whilst weakness in edge detection is avoided. However, it has difficulty in correctly determining the finger-segment protrusion.

In [13], a real-time hand gesture recognition algorithm is proposed using low-resolution depth images. The main drawback here is it requires a Canesta camera for the acquisition of depth images. In [14], the modified census transform is used for hand gesture classification and recognition. However, it requires that in each image the gesture should be accurately centered and all images must have the same size. In [15], a shape descriptor is presented for the recognition of static gestures. However, the accuracy is sensitive to the polygonal approximation of the static gesture contour.

1.2 Research Objectives

With advances in the computer technologies and the multimedia format, there is a continuative explosion in the amount and complexity of digital data being generated, stored, transmitted, analyzed and accessed. Much of this information is multimedia in nature, including digital images, videos, audios, graphics, and text data. In order to make use of this vast amount of data, effective and efficient techniques need to be developed and deployed to retrieve multimedia information based on its content. Among the various types of media, images are of prime importance, and are not only the most widely used media type apart from text, but also one of the most widely used methods for representing and retrieving videos and other multimedia information.

Among numerous multimedia data, retrieving the interesting objects which user needs is a very important task. In this way, the interaction between human and machine that helps to answer user's need to be designed. Human interactive systems have already attracted a lot of research interests in recent years for this reason, especially for content-based image retrieval systems (CBIR). And contrary to the early systems that focused on fully automatic strategies, recent approaches also introduced human-computer interaction [16], [17]. This is a strong stimulation for doing my research.

In a system of CBIR, normally the primary work-"Search task" may be initiated using a query as an example. The similar images within a top rank are presented to the user. Then, the interactive process allows the user to refine his/her request as much as necessary in a relevance feedback loop. Many kinds of interaction between the user and the system have been proposed [18], but most of the time, user information consists of binary labels indicating whether or not the image belongs to the desired concept. The positive labels indicate relevant images for the current concept, and the negative labels irrelevant images.

As a specific area of interactive image processing, the considerable interest in hand gestures detection and recognition has been led by the wish to provide a more natural means of interacting with computers. Recently, many researchers have devoted themselves to developing communication aids for the deaf people [19]–[25]. Deaf persons use sign language or hand gestures to express themselves, however, most of hearing people do not have special sign language expertise. Therefore, conversing between them is troublesome and sometimes even causes misunderstandings. Due to this situation, this thesis research is to work on detection and recognition of hand gestures to help resolving many existing issues in this area.

1.3 The Organization of the Thesis

The remainder of this thesis is organised as follows. In Chapter 2, literature is reviewed. We reviewed the basic principles of digital image processing, feature extraction and descriptors of image features. One of the aims of literature review is to learn from others' experience and to make us better aware of what other research has taken place in the similar areas. Also, the knowledge gained from this review allows us to select the most appropriate algorithms and puts our study in proper perspective and context.

In Chapter 3, current research activities of interactive image processing have been outlined. And an approach of hand motion detection based on segmentation is also introduced. Frame differencing is used to generate the difference between the current frame and its previous frame. Background modelling is employed to reduce noises and generate improved frame difference over the temporal window.

Chapter 4 proposes a novel approach to hand gesture recognition, in which the Gaussian Mixture Model and Expectation Maximization algorithm are utilized to improve the recognition accuracy. Experiment design is also explained including feature extraction, training phase and testing for the recognition task of three hand gestures, and results from 2788 images are discussed in details to show its superior performance in comparison with other existing algorithms.

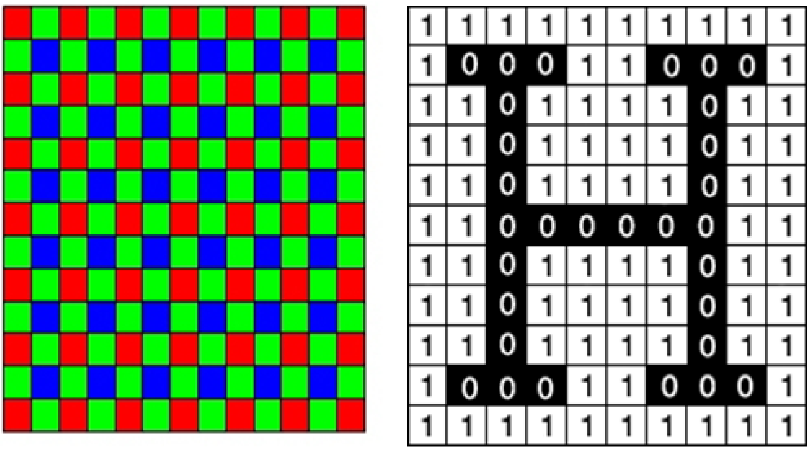
Finally, Chapter 5 summarizes the thesis work with both conclusions and future directions.

Chapter 2

Review of Literature

2.1 Definition of Digital Image Processing

A digital image is defined as a two-dimensional function $f(x, y)$, where x, y and f are all finite and discrete, and the amplitude f at coordinates (x, y) is called the intensity or gray level of the image at that spatial point. Typically, a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as pixel. Each pixel is assigned a value (black, white, shades of gray or color), which is represented in binary code (zeros and ones). The binary digits for each pixel are stored in a sequence by a computer and often reduced to a mathematical representation. The bits are then interpreted and read by the computer to produce an analog version for display or printing. The pictures below show how a digital image is built.



1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	0	1	1	1	1	0	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Figure 2.1 Pixels form a digital image

In general, digital image processing refers to a procedure to apply computer algorithms and process digital images to achieve some expected targets, such as enhancement, compression, etc. Here, three levels of computerized processes are always used namely low-level, mid-level, and high-level processes. Low-level processing is pre-processing for further analysis, and it aims to reduce noise and increase the contrast in the image. Traditional methods for low-level image processing include low-pass filtering for noise suppression, grey-level operations such as histogram equalization for contrast improvement, etc. Mid-level processing aims at extracting valuable features from images, such as region/object segmentation and edge detection. Accurate region segmentation facilitates subsequent higher level processing. High-level image processing is the intelligent part, which attempts to identify the regions or features previously detected. Techniques such as fuzzy logic, neural network, mathematics model and other artificial intelligent techniques can be applied for recognition and classification.

The relationships and differences among these three levels of processing are illustrated in the figure below, where a low-level process is characterized by the fact that both its inputs and outputs are images. As for mid-level and high-level image processing, their outputs are features and recognition results, respectively.

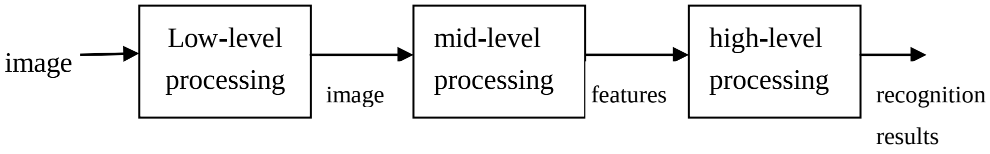


Figure 2.2 Three levels of digital image processing

2.2 Principles and Main Topics in Digital Image Processing

2.2.1 Principles

As mentioned above, digital image processing includes at least three levels, and these levels need to be completed stage by stage. In addition, another two additional stages for image processing are digital image acquisition and image output. Acquisition here means conversion and quantisation from analogue mode to digital format via a special device such as scanners, cameras, etc. As for image output, it is an inverse procedure of acquisition to convert digital image to analogue one for printing, etc.

Operations of Image processing can be roughly divided into three major categories, including image compression, image enhancement and restoration, and measurement extraction. Image compression is familiar to most people which involves reducing the amount of memory needed to store a digital image. Since there may exist defects introduced during image acquisition, image enhancement is used to remove these defects for improved image quality. Once the image is in good condition, the measurement extraction operation can be used to obtain useful information from the image for recognition and decision, etc.

2.2.2 Examples in Image Processing

To better understand the process of digital image processing, some examples on image enhancement and measurement extraction are given below. The examples shown all operate on 8-bit grey-level images. This means that each pixel in the image is stored as a number between 0 and 255 where 0 represents a black pixel, 255 represents a white pixel and values in-between represent shades of grey. These operations can be naturally extended to operate on color images.

Figure 2.3 shows an example on image enhancement and restoration. The image at its top left has a corrugated effect due to a fault in the acquisition process. This corrugated effect can be removed by frequency-domain filtering, where bright spots are removed from the Fourier Transform of the image (top right of Figure 2.3) and the results is also given (bottom left of Figure 2.3). After an inverse Fourier Transform, enhanced image without the corrugated background is obtained (bottom right of Figure 2.3).

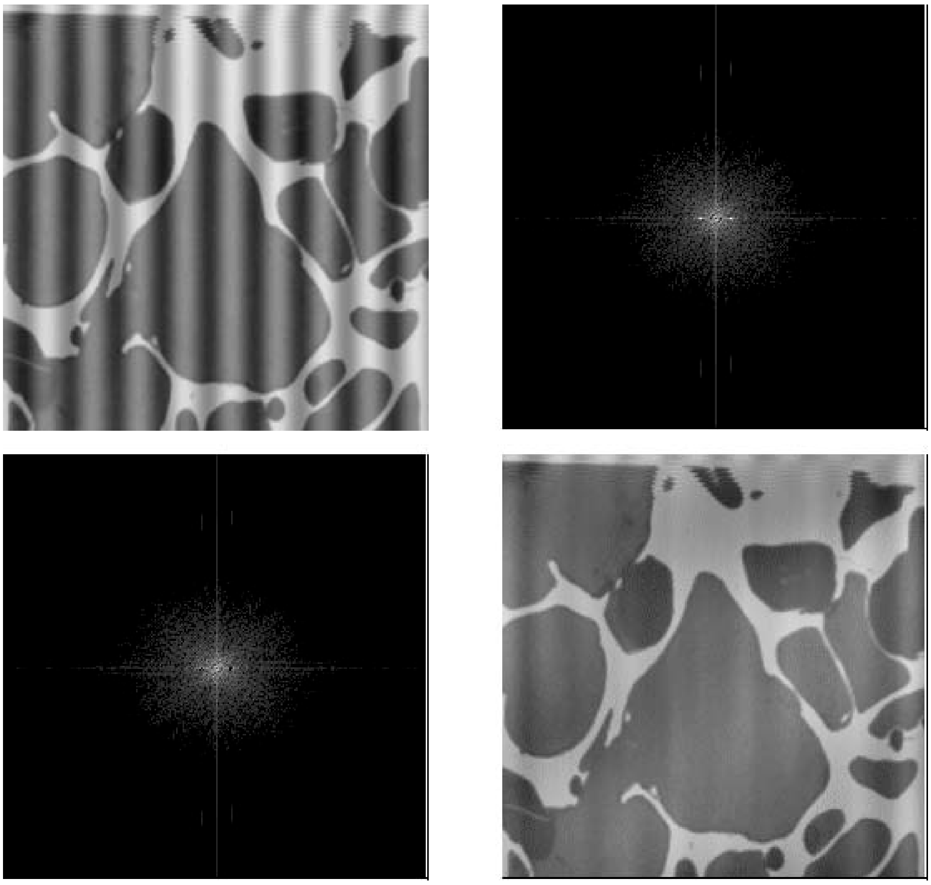


Figure 2.3 Image Enhancement via Frequency-domain filtering

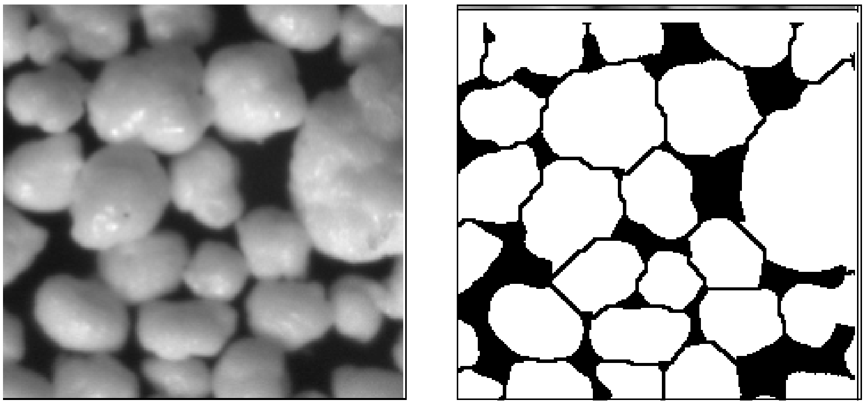


Figure 2.4 Image Segmentation

Figure 2.4 illustrates an example on image measurement extraction, where the original image (to the left) contains some objects. Through some processing such as edge detection and image segmentation, these objects are extracted with their outlines clearly identified in the right image for further classification, etc.

2.3 Feature Extraction

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

A good visual content descriptor should be invariant to the accidental variance. However, such invariance should not lead to loss of discriminative ability between essential differences. Consequently, there is a tradeoff between the invariance and the discriminative power of visual features. Invariant description has been largely investigated and employed in computer vision (like object recognition), but is relatively new in image retrieval [26].

A visual content descriptor can be defined as either global or local. A global descriptor is defined on the whole image, whereas a local descriptor uses the visual features of regions or objects within the image. To extract local visual descriptors, an image is often divided into parts first. The simplest way of dividing an image is to partition the image into tiles of equal size and shape. A simple partition does not generate perceptually meaningful regions but it helps to provide a finer resolution in representing the global features of the image. A better solution is to divide the image into homogenous regions according to some criterion using image segmentation algorithms. A more complex way of dividing an image is to extract semantically meaningful objects such as a ball, a car and a horse. Currently, automatic extraction of objects from images in broad domains is still unsolved.

2.4 Descriptions of Image Features

General visual features to be extracted from images for analysis purpose include color, texture, shape and spatial relationship, etc., and they are introduced below.

2.4.1 Color Feature

Color is the most important and extensively used visual content for image retrieval [27,28,29,30,31,32,33,34,35,36]. Its 3-D values make its discrimination potentiality superior to the 1-D gray values of images. To select an appropriate color description, color space must be determined first.

2.4.1.1 Color Space

In a color image, each pixel can be represented as a point in a 3D color space. Typical color spaces used for image retrieval include RGB, Munsell, CIELab, CIELuv, HSV, and opponent color space. There is no agreement on which color space is the best. However, uniformity is one of the desirable characteristics of an appropriate color space for image retrieval [37]. Uniformity here means that two color pairs that are perceived as equal by viewers should be equal in similarity distance in a color space. In other words, the measured distance or similarity among the colors must be directly related to the psychological similarity among them.

RGB space, being composed of three color components Red, Green, and Blue, is widely used for image display. In contrast, CMY space is a color space primarily used for printing in which the three color components are Cyan, Magenta, and Yellow. Both RGB and CMY spaces are device-dependent and perceptually non-uniform.

The CIELab and CIELuv spaces are device independent and also perceptually uniform. They consist of a luminance or lightness component and two chromatic components (a and b or u and v). How to transform RGB space to CIELuv or CIELab space can be found in [38].

HSV space is widely used in computer graphics and is more intuitive in describing color. The three color components are Hue, Saturation (Lightness) and

Value (Brightness). The hue component is invariant to illumination changes and camera directions and hence more suitable for object retrieval. Transformation from RGB space to HSV space can be easily completed [39].

2.4.1.2 Color Descriptors

In the following sections, some commonly used color descriptors will be introduced, including the color histogram, color coherence vector, color correlogram.

(1) Color Histogram

The color histogram is always an effective representation of the color content in an image if the color pattern is unique in comparison with the rest of the data set. The color histogram is easy to compute and effective in characterizing both the global and local color distribution within an image. In addition, it is invariant to translation and rotation and also insensitive (change slowly) to the scale, occlusion and viewing angle.

Since one pixel in an image can be described by three components, such as red, green, and blue components in RGB space, a histogram can be defined for each color component which represents the distribution of the number of pixels for each quantized bin. Obviously, more bins in a color histogram will provide more discrimination power. However, a large number of bins in the histogram will not only increase the computational cost, but will also be inappropriate for efficient indexing for image databases.

Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance hence it is possible to reduce the number of bins in calculating color histogram. One way to reduce the number of bins is to use the opponent color space in which the brightness of the histogram can be down sampled. Another way is to employ clustering methods to determine the K best colors in a given space for the given image set, and each of these best colors will be taken as a histogram bin. Since the clustering process takes into consideration of the color distribution of all images in the database, the results obtained will be robust. Another option is to use the bins that have the largest pixel numbers to enable the majority of pixels of an image being captured by a small number of histogram bins [40]. Such a reduction does not degrade the performance in histogram matching, but may even improve it since small histogram bins here are more likely to be noisy.

When a large number of images are contained in the image database, histogram comparison becomes saturating in terms of discrimination. To solve this problem, the joint histogram technique is introduced [33]. In addition, color histogram does not consider spatial information of pixels, thus similar histograms may refer to very different images. For large databases, this problem becomes especially serious. To increase discrimination power and solve this problem, several improvements have been proposed to incorporate spatial information. One simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. The division can be simple rectangular partition or complex region or even object based segmentation. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

(2) Color Coherence Vector

Color coherence vectors (CCV) is proposed to incorporate spatial information into the color histogram in a different way [41]. 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. The CCV is defined as the vector $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_N, \beta_N)$ where α_i and β_i denote the numbers of coherent and incoherent pixels in the i$^{\text{th}}$ color bin. As seen, $(\alpha_1 + \beta_1, \alpha_2 + \beta_2, \ldots, \alpha_N + \beta_N)$ is the color histogram of the image.

Due to spatial information considered, CCV provides better retrieval results than the color histogram, especially for images which have either mostly uniform color or mostly texture regions. In addition, for both the color histogram and CCV representation, the HSV color space is found better than CIELuv and CIELab spaces.

(3) Color Correlogram

The color correlogram [29] was proposed to characterize not only the color distributions of pixels, but also the spatial correlation of color pairs. A color correlogram is a table indexed by color pairs, where the k^{th} entry for (i, j) specifies the probability of finding a pixel of color j at a distance k from a pixel of color i in the image. Let I represent the entire set of image pixels and $I_{c(i)}$ represent the set of pixels whose colors are $c(i)$. Then, the color correlogram is defined as:

$$\gamma_{i,j}^{(k)} = \Pr_{p_1 \in I_{c(i)}, p_2 \in I} \left[p_2 \in I_{c(j)} \middle\| p_1 - p_2 \right| = k \Big] \tag{1}$$

where i, j $\in$ {1, 2, …, N}, k$\in$ {1, 2, …, d}, and | $p_1 - p_2$ | is the distance between pixels p_1 and p_2.

If all the possible combinations of color pairs are considered, the size of the color correlogram will be very large. A simplified version of color correlogram is often used namely the color autocorrelogram where only the spatial correlation between identical colors is captured and thus reduces the dimension. In comparison with the color histogram and CCV, the color autocorrelogram provides the best retrieval results, although it is also the most computational expensive due to its high dimensionality.

(4) Invariant Color Features

In general, color always varies considerably with the change of illumination, the orientation of the surface, and the viewing geometry of the camera. However, invariance to these environmental factors is not considered in most of the color features discussed above.

Recently, invariant color representation has been introduced to CBIR systems. In [42], a set of color invariants was derived for object-based retrieval on the base of the Schafer's object reflection model. In [43], specular reflection, shape and illumination invariant representation is presented based on blue ratio vector (r/b, g/b, 1). In [44], a surface geometry invariant color feature is provided for image retrieval.

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

2.4.2 Texture Feature

Texture is another important feature of images, and various texture features have been investigated in pattern recognition and computer vision. Basically, texture representation methods can be classified into two categories, i.e. structural and statistical. Structural methods include morphological operator and adjacency graph, which describe texture by using structural primitives and their placement rules. They are most effective for texture being very regular. Statistical methods are characterized by the statistical distribution of the image intensity, which include Fourier power spectra, cooccurrence matrices, shift-invariant principal component analysis (SPCA), Tamura feature, Wold decomposition, Markov random field, fractal model, and multi-resolution filtering techniques such as Gabor and wavelet transform. In this section, we introduce a number of texture representations [45]-[64], which have been frequently used and have found to be effective in CBIR systems.

In accordance with psychological studies on the human perception of texture, the Tamura features [62] are designed including coarseness, contrast, directionality, line-likeness, regularity, and roughness. The first three components of Tamura features have been widely used in some well-known CBIR systems, such as QBIC [65, 66] and Photobook [67] and their definitions are given as follows.

(1) Coarseness

Coarseness is used to measure the granularity of the texture. To calculate the coarseness F_{crs}, moving averages $A_k(x, y)$ are computed first using $2^k \times 2^k$ (k = 0, 1, …, 5) size windows at each pixel (x, y), i.e.,

$$A_k(x,y) = \sum_{i=x-2^{k-1}}^{x+2^{k-1}-1} \sum_{j=y-2^{k-1}}^{y+2^{k-1}-1} g(i,j)/2^{2k} \tag{2}$$

where $g(i,j)$ is the pixel intensity at (i,j).

Then, the differences between pairs of non-overlapping moving averages in the horizontal and vertical directions for each pixel are computed, i.e.,

$$E_{k,h}(x,y) = \left| A_k(x+2^{k-1},y) - A_k(x-2^{k-1},y) \right|$$

$$E_{k,v}(x,y) = \left| A_k(x,y+2^{k-1}) - A_k(x,y-2^{k-1}) \right| \tag{3}$$

After that, the value of *k* that maximizes E in either direction is used to set the best size for each pixel, i.e.

$$S_{best}(x,y) = 2^k \tag{4}$$

The coarseness is then computed by averaging S_{best} over the entire image, i.e.,

$$F_{crs} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{best}(i,j) \tag{5}$$

Instead of taking the average of S_{best}, an improved version of the coarseness feature can be obtained by using a histogram to characterize the distribution of S_{best}. Compared with a single value, histogram-based coarseness representation can greatly increase the retrieval performance. This improvement makes the feature capable of dealing with an image or region of multiple texture properties, and thus is more useful to CBIR applications.

(2) Contrast

The formula for the contrast F_{con} is as follows:

$$F_{con} = \frac{\sigma}{\alpha_4^{1/4}} \tag{6}$$

where $\alpha_4 = \mu_4 / \sigma^4$, μ_4 is the fourth moment about the mean, and σ^2 is the variance. This feature can be extracted from the whole image or regions, respectively.

(3) Directionality

To compute the directionality, image is convoluted with two 3x3 arrays (i.e. $\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$ and $\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$) and a gradient vector at each pixel is then computed.

The magnitude and angle of this vector are defined as:

$$|\Delta G| = (|\Delta_H| + |\Delta_V|)/2$$

$$\theta = \tan^{-1}(\Delta_V / \Delta_H) + \pi / 2 \qquad (7)$$

where Δ_H and Δ_V are the horizontal and vertical differences of the convolution.

Then, a histogram of θ can be constructed as H_D by quantizing θ and counting the pixels with the corresponding magnitude |ΔG| larger than a threshold. For highly directional images, this histogram will exhibit strong peaks. While for images without strong orientation, the histogram H_D will be relatively flat. Based on the sharpness of the peaks, an overall directionality measure F_{dir} is then obtained as below:

$$F_{dir} = \sum_{p}^{n_p} \sum_{\phi \in \omega_p} (\phi - \phi_p)^2 H_D(\phi) \qquad (8)$$

where p ranges over n_P peaks; and for each peak p, ω_P is the set of bins distributed over it; while ϕ_P is the bin that takes the peak value.

2.4.3 Shape Feature

Shape features of objects or regions have been widely applied in many digital image processing systems [68, 69, 70, 71]. Compared with color and texture features, shape features can be used after regions or objects have been segmented in images. Due to the inaccuracy and difficulty in image segmentation, shape features has limitations and only suitable for special applications in which objects or regions are readily available. Two main categories of methods used for shape description include boundary-based (rectilinear shapes [70], polygonal approximation [72], finite element models [73], and Fourier-based shape descriptors [74, 75, 76]) and region-based approaches (statistical moments [77, 78]). A good representation of shape feature should be invariant to translation, rotation and scaling. In this section, some of these shape features that have been commonly used in image retrieval applications will be described, and a concise comprehensive introductory overview can be found in [79].

（1） Moment Invariant

Moment invariants are classical shape representation. If the object R is represented as a binary image, then the central moments of order $p+q$ for the shape of R are defined as:

$$\mu_{p,q} = \sum_{(x,y)\in R} (x - x_c)^P (y - y_c)^q \tag{9}$$

where (x_c, y_c) denotes the center of object. This central moment can be normalized to scale invariant as follows [80]:

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}.. \quad \gamma = \frac{p+q+2}{2} \tag{10}$$

Based on these moments, a set of moment can be derived which are invariant to translation, rotation, and scale [77, 78]:

$$\begin{aligned}
&\phi_1 = \mu_{2,0} + \mu_{0,2} \\
&\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\
&\phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2 \\
&\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2 \\
&\phi_5 = (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})\left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2\right] + \\
&(\mu_{0,3} - 3\mu_{2,1})(\mu_{0,3} + \mu_{2,1})\left[(\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{3,0} + \mu_{1,2})^2\right] \\
&\phi_6 = (\mu_{2,0} - \mu_{0,2})\left[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2\right] + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1}) \\
&\phi_7 = (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2})\left[(\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{0,3} + \mu_{2,1})^2\right]
\end{aligned} \tag{11}$$

（2） Turning Angles

For a 2D object, its contour can be represented as a closed sequence of successive boundary pixels (x_s, y_s), where $0 \le s \le N-1$ and N is the total number of pixels on the boundary. To measure the angle of the counterclockwise tangents as a function of the arc-length s according to a reference point on the object's contour, the turning function or turning angle $\theta(s)$ is defined as:

$$\theta(s) = \tan^{-1}\left(\frac{y_s^{'}}{x'_s}\right)$$

$$y'_s = \frac{dy_s}{ds}, \quad x'_s = \frac{dx_s}{ds} \tag{12}$$

One major problem here is that $\theta(s)$ appears variant to the rotation of object and the choice of the reference point. If the reference point is shifted along the boundary of the object by t, the new turning function becomes $\theta(s+t)$. If the object is rotated by angle ω, the new function will become $\theta(s)+\omega$.

Consequently, to compare the shape similarity between two objects A and B using their turning functions, the minimum distance needs to be calculated over all possible shifts t and rotations ω, i.e.,

$$d_p(A,B) = \left(\min_{\omega\in R, t\in[0,1]} \int_0^1 \left| \theta_A(s+t) - \theta_B(s) + \omega \right|^p ds \right)^{\frac{1}{p}} \tag{13}$$

Here each object has been re-scaled with the total perimeter length being 1. The measure $d_p(A,B)$ is invariant under translation, rotation, and change of scale.

（3） Fourier Descriptors

Fourier descriptors describe the shape of an object using the Fourier transform of its boundary. Again, the contour of a 2D object is considered as a closed sequence of successive boundary pixels (x_s, y_s), where $0 \le s \le N-1$ and N is the total number of pixels on the boundary. Three types of contour representations can be defined below including curvature, centroid distance, and complex coordinate function.

The curvature $K(s)$ at a point s along the contour is defined as the change rate in tangent direction of the contour, i.e.

$$K(s) = \frac{d}{ds}\theta(s) \tag{14}$$

where $\theta(s)$ is the turning function of the contour, defined as （12）.

The centroid distance $R(s)$ is defined as the distance between boundary pixels and the object centroid (x_c, y_c), i.e.

$$R(s) = \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2} \tag{15}$$

The complex coordinate can be obtained by representing the coordinates as complex numbers:

$$Z(s) = (x_s - x_c) + j(y_s - y_c) \tag{16}$$

According to these three types of contour representations, their Fourier transforms generate three sets of complex coefficients which represent the shape of an object in the frequency domain. Coefficients of lower frequency describe the general shape property, while coefficients of higher frequency reflect shape details. To achieve rotation invariance, in the complex coefficients only the amplitudes are used and the phase components are discarded. To achieve scale invariance, the amplitudes of the coefficients can be divided by the amplitude of DC (Discrete Cosine) component or the first non-zero coefficient. The translation invariance is obtained directly from the contour representation.

The Fourier descriptor of the curvature is then obtained as:

$$f_K = \left[|F_1|, |F_2|, \ldots |F_{M/2}| \right] \tag{17}$$

The Fourier descriptor of the centroid distance is determined by

$$f_R = \left[\frac{|F_1|}{|F_0|}, \frac{|F_2|}{|F_0|}, \ldots, \frac{|F_{M/2}|}{|F_0|} \right] \tag{18}$$

where F_i in （17） and（18） denotes the i^{th} component of Fourier transform coefficients. As the curvature and centroid distance functions are real, only the positive frequency axes are considered which lead to symmetry on their Fourier transform, i.e., $| F_{-i} | = | F_i |$.

The Fourier descriptor of the complex coordinate is then attained as

$$f_Z = \left[\frac{|F_{-(M/2-1)}|}{|F_1|}, \ldots, \frac{|F_{-1}|}{|F_1|}, \frac{|F_2|}{|F_1|}, \ldots, \frac{|F_{M/2}|}{|F_1|} \right] \tag{19}$$

where F_1 is the first non-zero frequency component used for normalization. Here both negative and positive frequency components are considered, and the DC coefficient is discarded as it is dependent on the position of a shape.

To ensure that the resulting shape features of all objects have the same length, the boundary $((x_s, y_s), 0 \le s \le N-1)$ of each object is normalized by re-sampling into M samples before performing the Fourier transform. The selected M will enable fast implementation of Fourier transform for efficiency.

（4） Circularity, Eccentricity, and Major Axis Orientation

Circularity of a shape is computed as:

$$\alpha = \frac{4\pi S}{P^2} \tag{20}$$

where S is the size and P denotes the perimeter of an object. As seen, α ranges between 0 (corresponding to a perfect line segment) and 1 (corresponding to a perfect circle).

For each object or region, its major axis orientation is defined as the direction of the largest eigenvector of the second order covariance matrix object. The eccentricity is defined as the ratio of the smallest eigenvalue to the largest eigenvalue.

2.4.4 Spatial Information

Spatial constraints are useful in distinguishing regions or objects with similar color and texture properties. For example, it is difficult to discriminate regions of blue sky and ocean due to similar color histograms, but the differences of their spatial locations in images are useful for the discrimination. Consequently, the spatial location of regions (or objects) or the spatial relationship between multiple regions (or objects) in an image is very useful in such a context.

The 2D strings proposed by Chang [81] are the most widely used representation of spatial relationship, where projecting images along the x and y directions are used to construct the features. Two sets of symbols, V and A, are defined on the projection where V represents objects and A represent spatial relationships between objects. As its variants, the 2D G-string [82], 2D C-string [83] and 2D-B string [84] have been proposed. The 2D G-string cuts all the objects along their minimum bounding box and extends the spatial relationships into two sets of spatial operators including local and global spatial relationships. 2D C-string can minimize the number of cutting objects. 2D-B string represents an object by two symbols, i.e. the beginning and ending boundary of the object. All the methods above facilitate three types of query, including querying images containing object O_1, O_2, …., O_n, querying images containing objects of minor distance but having certain relationship between each other, and querying images having certain distance relationship with each other.

In addition to the 2D strings, some other methods are also proposed for spatial information representation including spatial quad-tree [85] and symbolic image [86]. However, searching images based on spatial relationships of regions remains difficult due to the fact that reliable segmentation of objects or regions is often infeasible and extremely application specific. Although in some systems regions are simply extracted by dividing the images into regular sub-blocks [87], such spatial division schemes achieve limited success since most natural images are not spatially constrained to regular sub-blocks. To solve this problem, a method based on the radon transform to exploit the spatial distribution of visual features without a sophisticated segmentation is proposed in [88, 89].

2.5 Summary

In this chapter, the basic principles of digital image processing are reviewed. As the most important components of a digital image, features and descriptors of features are introduced, including color, texture, shape and spatial information. Color is considered as a very important feature due to its various 3D values in different color spaces. Texture is another important feature because it represents the primitive structure or statistical distribution of image intensity. Compared with color and texture, shape feature is invariant to translation, rotation and scaling, and can be used after segmentation. At last, spatial information is introduced.

Chapter 3

Motion Detection for Hand Segmentation

3.1 Introduction to Interactive Image Processing

The whole world has witnessed the recent and rapid growth in the generation, processing, and sharing of multimedia data. This trend has resulted in the emergence of numerous multimedia repositories that require efficient methods for storage, sharing, and organization of large volumes of multimedia data. As data compression and management of network have become relatively mature, a subsequent shift of the research attention has changed from storage and bandwidth considerations to the management of information content in multimedia [90]. Therefore, interactive content-based image retrieval systems have received much interest for locating relevant information within image repositories. These systems rely on low-level representations of images in terms of their visual content such as color, shape, and texture in order to compare images. In devising a standard scheme for these low-level features, the MPEG-7 standard is proposed for the description of multimedia content [91], whose primary goal is to enable interoperable searching, indexing, filtering, and access to multimedia content [92]. This interoperability is essential in systems to allow retrieval and searching among distributed repositories.

The low-level representation of images used by the MPEG-7 standard as well as most content-based visual data retrieval systems leads to several shortcomings when comparing and retrieving images. The difficulty is from two sources [93]: First, the semantic gap between low-level image representations and higher level concepts by which humans interpret and understand images; Second, the perceptual subjectivity of the users' similarity judgment.

The existing semantic gap between the low-level representation of images and the high-level user concepts has many consequences in CBIR. First, the ideal query is always unknown to the system if it cannot be represented in terms of the low-level features. In addition, specification of a query to the system using an example results in many ambiguities in terms of the relevant features as well as importance of each of these features to the user. A second problem is that the mapping between low-level feature spaces and high-level user concepts is not known apriori and needs to be determined for each individual query. The overall effect of this gap is that images that may be similar to a query in terms of the low-level features are deemed similar by the system even if they do not contain the conceptual content. In the same light, images that contain the same high-level content may not have similar low-level representations.

Perceptual subjectivity of similarity judgment comes about as different users interpret the visual contents and the similarity between them differently. The main implication of this subjectivity is that the measure used to calculate similarity between images must be user and query dependent.

To alleviate the problems that come about because of the semantic gap and user subjectivity, interactive image processing systems are proposed that place the user in the loop during retrievals. Such relevance feedback approaches aim at learning intended high level query concepts and adjust for subjectivity in judgment by exploiting user input on successive iterations. Relevance feedback (RF) [94] is a very effective method to bridge this gap and to scale up the performance in CBIR systems.

RF focuses on the interactions between the search engine and the user by requiring the user to label semantically positive or negative feedbacks. Generally, the user provides quality assessment of the retrieval results to the system by indicating the rank of satisfaction with each of the retrieved results. The system then makes use of this feedback to adjust its query and/or the similarity measure in order to improve the next set of results.

Ishikawa [95] introduced both the query movement and the re-weighting techniques. Cox [96] formulated a minimization problem on the parameter estimation process. Zhou [97] proposed a stochastic comparison search. With the observation that all positive examples are alike and each negative example is negative in its own way, biased discriminant analysis (BDA) [98] and its enhanced version were developed.

Given the user feedback information, the key for a RF scheme is how to create a suitable classifier. But, RF is much different from the traditional classification problem because users would not like to provide a large number of feedbacks. Among various RF schemes, small sample learning methods, where the amount of the training samples is much smaller than the dimension of the descriptive features, are of the most promising.

3.2 Motion Detection for Moving Object Segmentation

In this section, we focus on motion detection and moving object segmentation to detect the change of hand gestures. Although there are many approaches proposed for motion detection in a continuous video stream, the basic principle is to compare the current video frame with its previous one or against a fixed/dynamic background. This is very useful in video compression applications when the changes need to be estimated to mark the changed part rather than the whole frame.

3.2.1 Frame Differencing

Frame differencing is a straightforward approach for motion detection, which uses the difference between two images as an indicator to show the changes caused by motion. Let $img(i)$ represent the i^{th} image in a sequence, the frame difference of this frame and its previous frame is defined as:

$$diff(i) = | img(i) - img(i-1) | \tag{21}$$

If the original image is a color one containing three or more color components, the difference above will also generate a color image. For simplicity, the input image is usually converted to grey one before differencing. Consequently, the resulted difference $diff(i)$ will also be an 8-bit grey image. Fig. 3.1 shows two inputted color images and their associated grey images for motion detection.

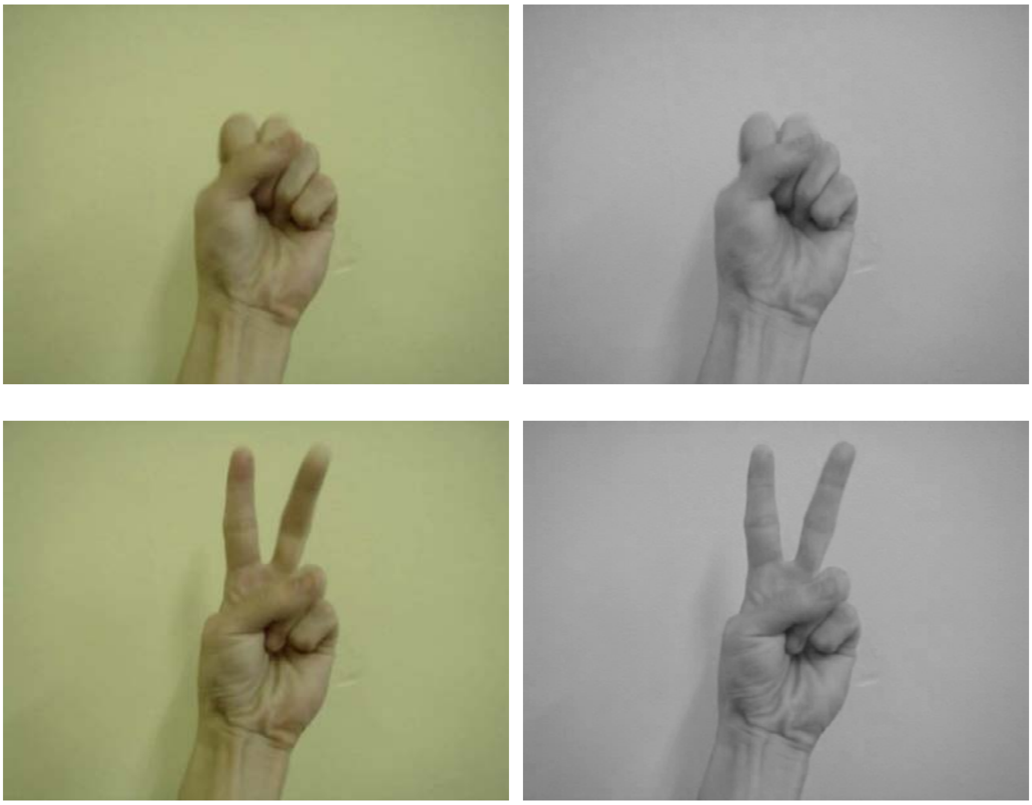


Figure 3.1: Two inputted color images (left) and their corresponding gray ones (right).

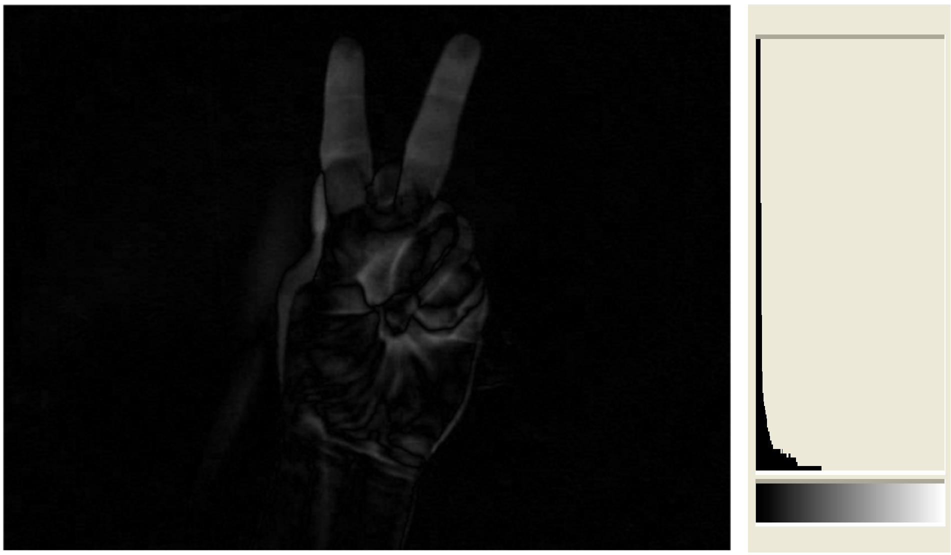


Figure 3.2: $diff(i)$ **and its histogram.**

With the extracted $diff(i)$, a simple thresholding th is then employed to classify the pixels as changed or unchanged ones. As a result, a binary image mask

$mask\_diff(i)$ can then be obtained in which white and black pixels represent those having been changed or remaining unchanged by motion.

$$mask\_diff(i) = \begin{cases} 1, & if \quad diff(i) > th \\ 0 & otherwise \end{cases} \tag{22}$$

Rather than using a fixed threshold, the threshold for each $diff(i)$ is determined automatically as follows, where μ and σ denote respectively the mean and standard deviation of $diff(i)$.

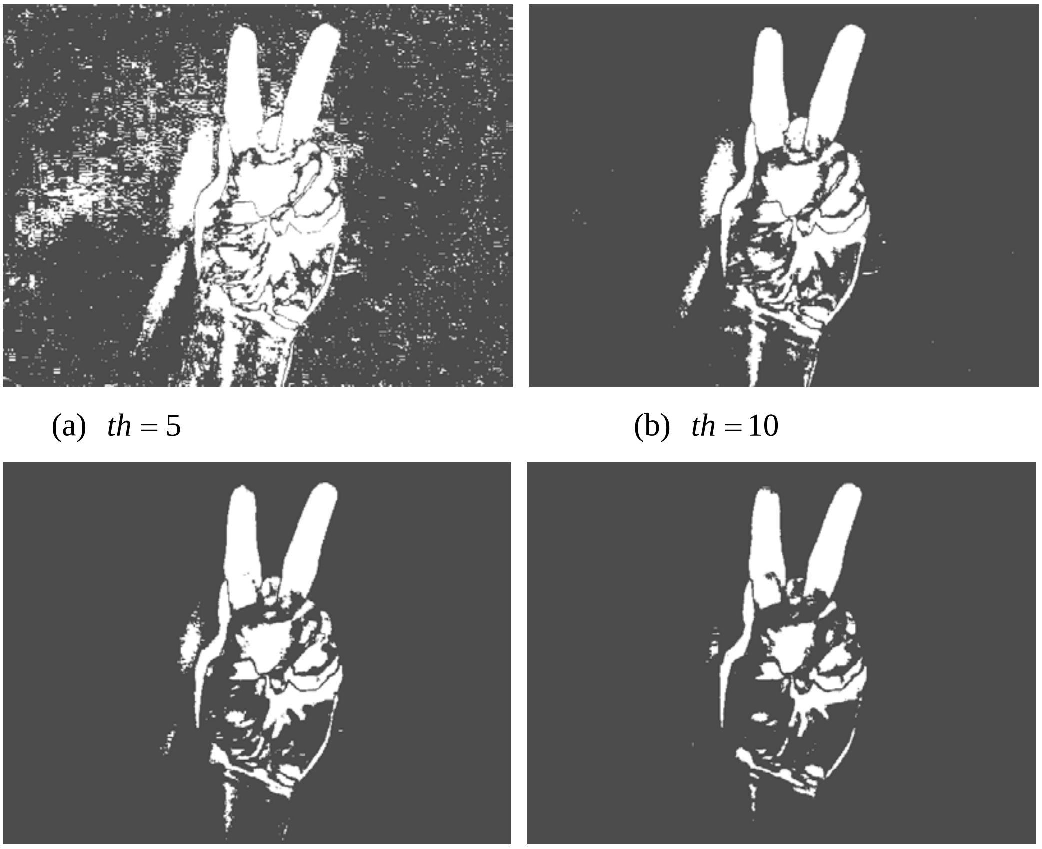
$$th(i) = \mu(i) + \sigma(i) \tag{23}$$

$diff(i)$ in Fig. 3.2 has $\mu = 5.47$ and $\sigma = 10.02$, hence we have $th(i) = 15.49$. The detected mask images under different thresholds are compared in Figure 3.3. When the threshold is set as 5, there are too many false pixels. On the contrary, a large threshold as 20 produces few false detected pixels, but it yields holes in the mask, such as the one on the left finger. Consequently, the results from the threshold of 15, or the one determined above, appear to be accurate and robust.

To further eliminate the false detected pixels and small moving areas, post-processing using a morphological filter is then applied. Firstly, erosion is employed to the detected binary mask $mask\_diff(i)$, followed by a dilation processing, both using 3*3 rectangle structure. If the new obtained mask image is donated as $mask\_diff1(i)$, we have

$$mask\_diff1(i) = dilation(erosion(mask\_diff(i))) \tag{24}$$

The result of processing is shown in Fig 3.4. A binary mask $mask\_diff1(i)$ is obtained after erosion filter and dilation processing, and shown at the left. Then, the mask is attached on the original image to represent the frame differencing clearly as shown at the right.



(a) $th = 5$

(b) $th = 10$

(c) $th = 15$

(d) $th = 20$

Figure 3.3: Extracted binary masks of motion using different thresholds.

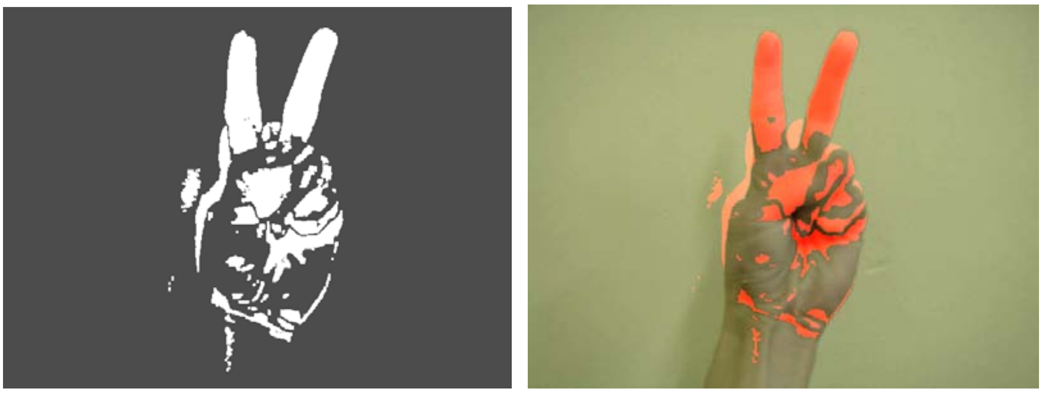


Fig. 3.4: Binary masks obtained after morphological erosion and followed by dilation (left) as post-processing and attached the mask on the original image (right).

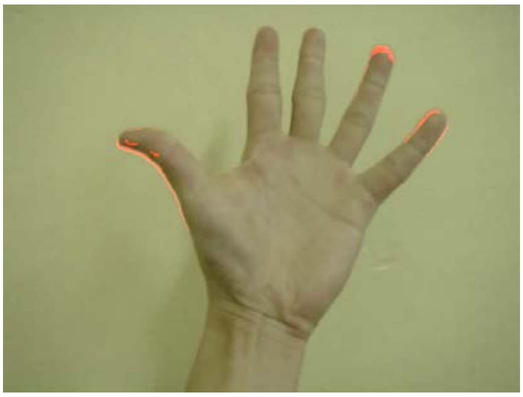


Fig. 3.5: Smooth or slow movement causes motion regions inconspicuous

From above picture Fig. 3.5 we can find the disadvantages of the approach. If the object is moving smoothly we'll only receive small changes from frame to frame. So, it's impossible to get the whole moving object. Things could become worse, when the object is moving so slowly. And sometimes the algorithms even do not generate any result at all. Therefore, a new approach using background modeling is presented in the next section for more accuracy.

3.2.2 Background Modelling

Background modeling is an important part for robust moving object detection as it can help to dynamically extract a "uniform" background against which the frame differencing is carried out in a more consistent way. This has been successfully applied in many vision applications when motion detection is required, such as visual surveillance. Moving object detection algorithm will be simple when a clean background image is available. Method of background extraction during training sequence and updating it during input frame sequence is called background modeling.

The main challenges in moving object detection is the extraction of a clean background and its updating. There are various methods for background modeling.

The commonly fastest and the most memory compact background modeling is the running average method. In this method, background extraction is done by arithmetic averaging on a training sequence. The method is introduced as following:

From the k^{th} image I_k in the image sequence, in total a temporal window is defined as ($I_{k-n0}, I_{k-n0+1},...., I_k,....., I_{k+n0}$), where there are $2n_0+1$ frames used, including the current frame, its previous and subsequent n_0 neighbouring frames, respectively. The average image over a temporal window is then extracted as follows:

$$av_k = \frac{1}{2n_0+1}\sum_{m=k-n_0}^{k+n_0} I_m \tag{25}$$

Consequently, each pixel value in av_k represents the mean value of that pixel over the temporal window. Meanwhile, the standard derivation for each pixel over the temporal window std_k is also attained as follows:

$$std_k(i,j) = \sqrt{\frac{1}{2n_0+1}\sum_{m=k-n_0}^{k+n_0}[I_m(i,j) - av_k(i,j)]^2} \tag{26}$$

For background pixels, limited changes in the temporal window are expected, i.e. smaller values in std_k. Therefore, a simple thresholding of std_k image can help to extract the background, where the threshold is also determined adaptively using the same strategy as introduced in (23).

$$bg_k(i,j) = \begin{cases} 1 & if \quad std_k(i,j) < th_2 \\ 0 & otherwise. \end{cases} \tag{27}$$

With the extracted background, the simple frame differencing is modified as follows.

$$diff_k(i,j) = \begin{cases} |\,av_k(i,j) - I_k(i,j)\,| & if \quad bg_k(i,j) = 1 \\ std_k(i,j) & otherwise. \end{cases} \tag{28}$$

And the mask of changed pixels is then obtained using the same way as applied to the difference from frame differencing, i.e.

$$mask\_diff_k(i,j) = \begin{cases} 1 & if \quad diff_k(i,j) > th_3 \\ 0 & otherwise. \end{cases} \tag{29}$$

where th_3 is a threshold determined from $diff_k$ using the same strategy in (23).

From a sequence of frames which begins with fist, then transforms into “Victory” sign and ends with fist, the original images are shown as below:

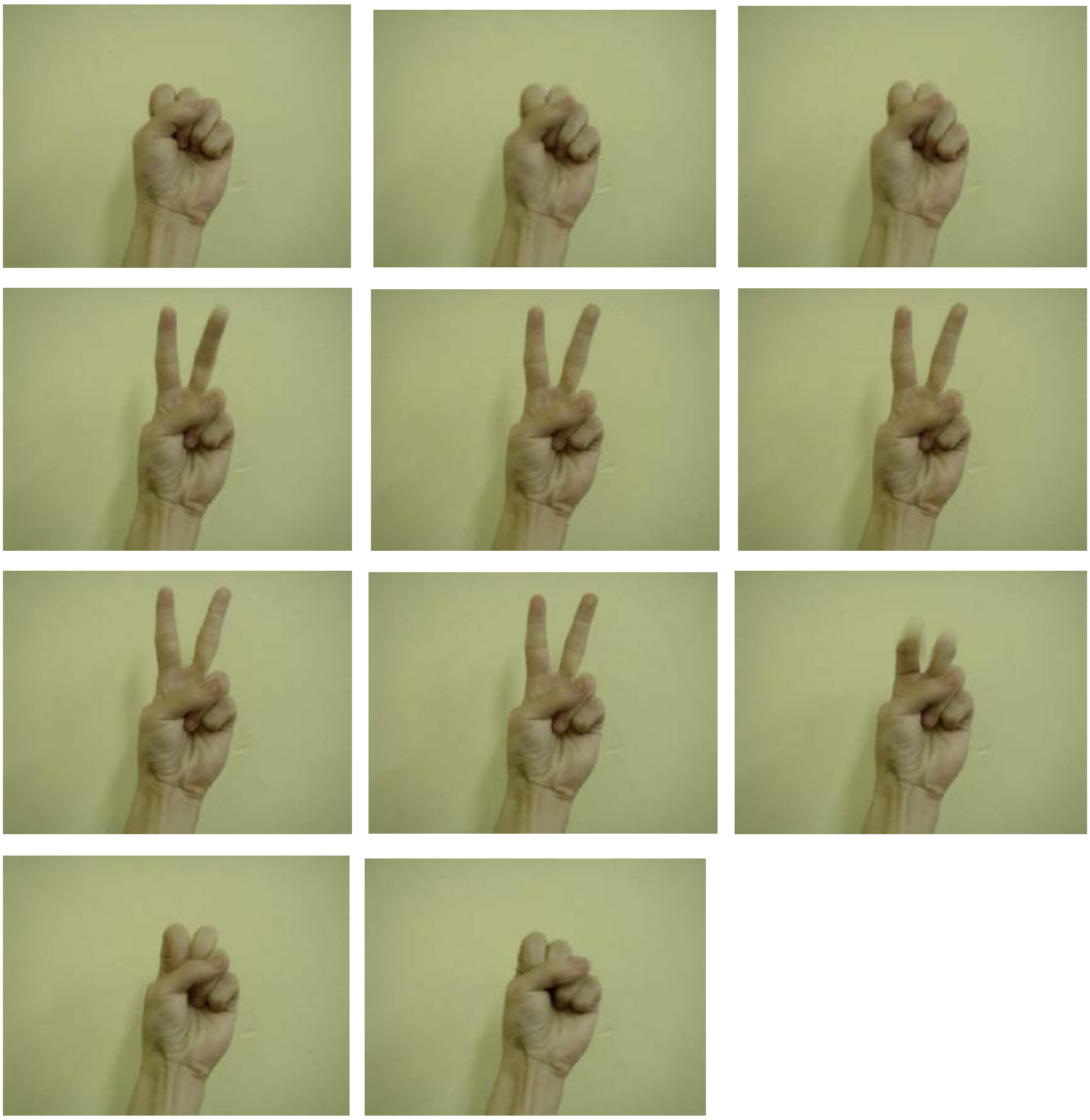


Fig 3.6 Sequence of 11 frames: from left to right, top to below, the hand gesture begins with fist, then turn to "Victory" sign and ends with fist.

By applying the approach we described above, each pixel is classified as a significant change or as a non-significant change. After removing noises, we extract the binary mask from the frame difference which is shown in Fig 3.4 with removing the background. The result is shown in Fig 3.7 in which the white pixel area shows the foreground or an object, and the background is turned into black:

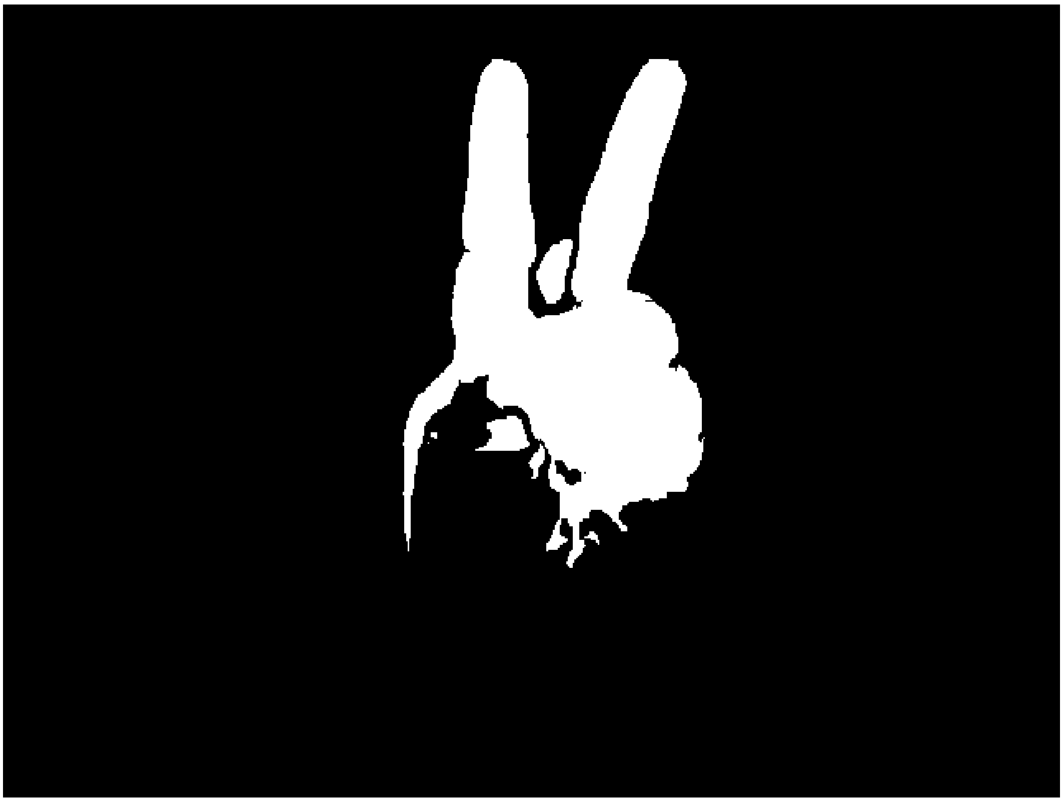


Fig 3.7 Binary mask of frame difference modified with background modelling

Apparently, background modeling can overcome the problem caused by directly frame differencing and help to extract the real moving parts for further recognition of hand gestures.

3.3 Summary

The segmentation-based approach compares the current frame with the previous frame in order to generate a difference value. When the value becomes more than predefined threshold value, motion event will be identified. Two problems have been issued: (1) if the movement takes place very slow, the difference value we calculated will not trigger the identification alarm, and this will cause a slow movement event to be ignored. In this case, we introduce the method which will compare the current frame with the average frame. (2) Background noise will sometimes result to decrease the performance of system. In this case, we introduce a new idea that can be simply defined as Temporal Window over Images Sequence. This new method will generate the standard derivation for each pixel over the temporal window in order to reduce noise and calculate improved frame difference.

Chapter 4

Hand Gesture Recognition with Gaussian Mixture Model

4.1 Generation of Proposed Approach

Due to the variation of gestures appearance, it is still a difficult problem to recognize and track human hand gestures. Also, it is hard to extract information from gesture images for 3D hand reconstruction. In most cases, we need model the hand gestures and adjust the parameters until they match the observations. These parameters should provide the desired information from the captured images. Since there is a huge amount of information in the captured images, one important issue here is to process the images to obtain the raw information that match the output model. In our system, the core issue addressed for gesture recognition is effective and robust feature extraction. Our proposed approach is different from existing ones as it focuses on estimating the gesture contained in an image by analyzing different complex features including shape, color and orientation histogram quantized in Gaussian Mixture Model (GMM).

GMM is a widely used statistical model in many applications of pattern recognition, which is often regarded as a versatile modeling tool as it can be used to approximate any probability density function (PDF) given a sufficient number of components, and impose only minimal assumptions about the modeled random variables. The advantage is including a rigorous statistical basis, the possibility of encoding spatial, color, texture and motion features in a unified system, and the ability to trade off accuracy of representation against data volume. Due to such advantages, our proposed technique builds upon the GMM to estimate the mutative meaning of human gestures in a compact and precise manner.

4.2 Description of Gaussian Mixture Model

GMM is one of the most widely used mixture modeling techniques. It's a simple model and is reasonably accurate when data are generated from a set of Gaussian distributions [99, 100]. Let $X_i = \{x_t, 1 \le t \le T^i\}$ denote the feature vectors for data points from the i^{th} class, they are modeled by a total number of J Gaussians as follows:

$$P\left(X_i \middle| \theta^i_{GMM}\right) = \prod_{t=1}^{T^i} \sum_{j=1}^{J} P(z_j) P_{z_j}\left(x_t \middle| u_j, \Sigma_j\right) \tag{30}$$

where θ^i_{GMM} refers to the model parameters, including $\{P(z_j), \mu_j, \Sigma_j, 1 \le j \le J\}$, and $P_{z_j}\left(x_t \middle| \mu_j, \Sigma_j\right)$ is the Gaussian distribution for the j^{th} class, with a mean vector μ_j and a covariance matrix $\sum_j$, i.e.:

$$\begin{aligned} &P_{z_j}\left(x_t \middle| \mu_j, \Sigma_j\right) = \\ &\frac{1}{(2\pi)^{D/2}\left|\Sigma_j\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(x_t - \mu_j\right)^T \sum\nolimits_j^{-1} \left(x_t - \mu_j\right)\right\} \end{aligned} \tag{31}$$

where D is the dimension of the feature vector x_t. Usually, Σ_j is set to be a diagonal matrix as $diag\{\sigma^2_{jd} : 1 \le d \le D\}$ in order to reduce the size of parameter space.

It can be seen from Eq. (30) that the data points of a specific class are generated from multiple Gaussian models with an identical weight $P(z_j)$. We define $\omega_j = P(z_j)$. In other words, an integrated Gaussian mixture model contains three basic parameters: Mixture weight, Mean vector and Covariance matrix, which can be represented as λ:

$$\lambda = \{\omega_j, \mu_j, \Sigma_j\} \tag{32}$$

where ω_j is the mixture weight, μ_j is the mean vector, and Σ_j is the covariance matrix. We use λ to stand for every single image. Additionally, we use

$$b_j(x) = P_{z_j}\left(x_t \middle| \mu_j, \Sigma_j\right) \text{ and } \sum_{j=1}^{J} \omega_j = 1. \tag{33}$$

in order to simplify the expression of $P_{z_j}\left(x_t \middle| \mu_j, \Sigma_j\right)$ in the following training phase.

4.3 The Expectation Maximization Algorithm

The EM algorithm is an efficient iterative procedure to compute the Maximum Likelihood (ML) [99, 101, 102] estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely. Before we use EM to compute ML in GMM in our experiment, the basic procedure of EM is detailed below:

In the EM algorithm, each iteration consists of two processes: the E-step, and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step is used in lieu of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration.

Let X be random vector which results from a parameterized family. We wish to find θ so that $P(X|\theta)$ is a maximum. This is known as the Maximum Likelihood (ML) estimate for θ. In order to estimate θ, it is typical to introduce the “log likelihood function” defined as,

$$L(\theta) = \ln P(X|\theta) \tag{34}$$

The likelihood function is considered to be a function of the parameter θ given the data X. Since $\ln(x)$ is a strictly increasing function, the value of θ which maximizes $P(X|\theta)$ also maximizes $L(\theta)$.

The EM algorithm is an iterative procedure for maximizing $L(\theta)$. Assume that after the n^{th} iteration the current estimate for θ is given by θ_n. Since the objective is to maximize $L(\theta)$, we wish to compute an updated estimate θ so that,

$$L(\theta) \;>\; L(\theta_n) \tag{35}$$

Equivalently we want to maximize the difference,

$$L(\theta) - L(\theta_n) = \ln P(X|\theta) - \ln P(X|\theta_n) \tag{36}$$

So far, we have not considered any unobserved or missing variables, and the EM algorithm provides a natural framework for the problems where such data exist. Alternately, hidden variables may be introduced purely as an artifice for making the maximum likelihood estimation of θ tractable. In this case, it is assumed that knowledge of the hidden variables will make the maximization of the likelihood function easier. Either way, denote the hidden random vector by Z and a given realization by z. The total probability $P(X|\theta)$ may be written in terms of the hidden variables z as,

$$P(X|\theta) = \sum_{Z} P(X|z,\theta)P(z|\theta) \tag{37}$$

We may then rewrite Equation (36) as,

$$L(\theta) - L(\theta_n) = \ln(\sum_{Z} P(X|z,\theta)P(z|\theta)) - \ln P(X|\theta_n) \tag{38}$$

Notice that this expression involves the logarithm of a sum. Because

$$\ln\sum_{i=1}^{n}\lambda_i x_i \geq \sum_{i=1}^{n}\lambda_i \ln(x_i) \tag{39}$$

, for constants $\lambda_i \geq 0$ with $\sum_{i=1}^{n}\lambda_i = 1$. This result (39) may be applied to Equation (38) provided that the constants λ_i can be identified. Consider letting the constants be of the form $P(z|X,\theta_n)$. Since $P(z|X,\theta_n)$ is a probability measure, we have that $P(z|X,\theta_n) \geq 0$ and that $\sum_z P(z|X,\theta_n) = 1$ as required.

Then starting with Equation (38) the constants $P(z|X,\theta_n)$ are introduced as,

$$\begin{aligned}
L(\theta) - L(\theta_n) &= \ln\left(\sum_Z P(X|z,\theta)P(z|\theta)\right) - \ln P(X|\theta_n) \\
&= \ln\left(\sum_Z P(X|z,\theta)P(z|\theta) \bullet \frac{P(z|X,\theta_n)}{P(z|X,\theta_n)}\right) - \ln P(X|\theta_n) \\
&= \ln\left(\sum_Z P(z|X,\theta_n) \bullet \frac{P(X|z,\theta\ )P(z|\theta)}{P(z|X,\theta_n)}\right) - \ln P(X|\theta_n) \\
&\geq \sum_Z P(z|X,\theta_n)\ln\left(\frac{P(X|z,\theta\ )P(z|\theta)}{P(z|X,\theta_n)}\right) - \ln P(X|\theta_n) \\
&= \sum_Z P(z|X,\theta_n)\ln\left(\frac{P(X|z,\theta\ )P(z|\theta)}{P(z|X,\theta_n)P(X|\theta_n)}\right) \\
&\overset{\Delta}{=} \Delta(\theta|\theta_n)
\end{aligned} \tag{40}$$

From Equation (40), we made use of the fact that $\sum_Z P(z|X,\theta_n) = 1$ so that $\ln P(X|\theta_n) = \sum_Z P(z|X,\theta_n)\ \ln P(X|\theta_n)$ which allows the term $\ln P(X|\theta_n)$ to be brought into the summation.

We continue by writing

$$\mathrm{L}(\theta) > \mathrm{L}(\theta_n) + \Delta(\theta|\theta_n) \tag{41}$$

and for convenience we also define,

$$\mathrm{l}(\theta|\theta_n) \ \underline{\underline{\Delta}}\ \mathrm{L}(\theta_n) + \Delta(\theta|\theta_n) \tag{42}$$

so that the relationship in Equation (41) can be made explicit as,

$$\mathrm{L}(\theta) \;\geq\; \mathrm{l}(\theta|\theta_n) \tag{43}$$

We have now a function, $\mathrm{l}(\theta|\theta_n)$ which is bounded above by the likelihood function $\mathrm{L}(\theta)$. Additionally, observe that,

$$\begin{aligned} l(\theta_n|\theta_n) &= L(\theta_n) + \Delta(\theta_n|\theta_n) \\ &= L(\theta_n) + \sum_Z P(z|X,\theta_n)\ln\frac{P(X|z,\theta_n)P(z|\theta_n)}{P(z|X,\theta_n)P(X|\theta_n)} \\ &= L(\theta_n) + \sum_Z P(z|X,\theta_n)\ln\frac{P(X,z|\theta_n)}{P(X,z|\theta_n)} \\ &= L(\theta_n) + \sum_Z P(z|X,\theta_n)\ln 1 \\ &= L(\theta_n) \end{aligned} \tag{44}$$

so for $\theta = \theta_n$ the functions $\mathrm{l}(\theta|\theta_n)$ and $\mathrm{L}(\theta)$ are equal.

Our objective is to choose a value of θ so that $\mathrm{L}(\theta)$ is maximized. We have shown that the function $\mathrm{l}(\theta|\theta_n)$ is bounded above by the likelihood function $\mathrm{L}(\theta)$ and that the value of the functions $\mathrm{l}(\theta|\theta_n)$ and $\mathrm{L}(\theta)$ are equal at the current estimate for $\theta = \theta_n$. Therefore, any θ which increases $\mathrm{l}(\theta|\theta_n)$ in turn increase the $\mathrm{L}(\theta)$. In order to achieve the greatest possible increase in the value of $\mathrm{L}(\theta)$, the EM algorithm calls for selecting θ so that $\mathrm{l}(\theta|\theta_n)$ is maximized. We denote this updated value as θ_{n+1}. This process is illustrated in Figure 4.1.

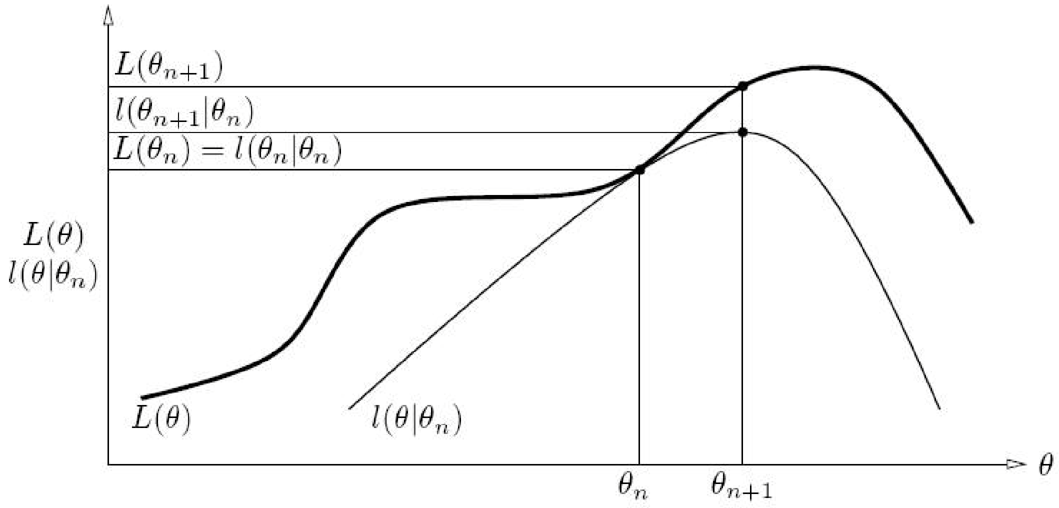


Figure 4.1: Graphical interpretation of a single iteration of the EM algorithm.

Formally we have,

$$\begin{aligned} \theta_{n+1} &= \arg\max_{\theta} \{ l(\theta|\theta_n) \} \\ &= \arg\max_{\theta} \left\{ L(\theta_n) + \sum_z P(z|X,\theta_n) \ln \frac{P(X|z,\theta)P(z|\theta)}{P(X|\theta_n)P(z|X,\theta_n)} \right\} \end{aligned} \tag{45}$$

Now drop terms which are constant w.r.t. θ

$$\begin{aligned} &= \arg\max_{\theta} \left\{ \sum_z P(z|X,\theta_n) \ln P(X|z,\theta)P(z|\theta) \right\} \\ &= \arg\max_{\theta} \left\{ \sum_z P(z|X,\theta_n) \ln \frac{P(X,z,\theta)}{P(z,\theta)} \frac{P(z,\theta)}{P(\theta)} \right\} \\ &= \arg\max_{\theta} \left\{ \sum_z P(z|X,\theta_n) \ln P(X,z|\theta) \right\} \\ &= \arg\max_{\theta} \left\{ E_{Z|X,\theta_n} \{ \ln P(X,z|\theta) \} \right\} \end{aligned} \tag{46}$$

In Equation (46) the expectation and maximization steps are apparent. The EM algorithm thus consists of iterating the:

1) E-step: Determine the conditional expectation $E_{Z|X,\theta_n} \{ \ln P(X,z|\theta) \}$

2) M-step: Maximize this expression with respect to θ.

At this point it is fair to ask what has been gained given that we have simply traded the maximization of L(θ) for the maximization of l($\theta | \theta_n$). The answer lies in the fact that l($\theta | \theta_n$) takes into account the unobserved or missing data Z. In the case where we wish to estimate these variables the EM algorithms provides a framework for doing so. Also, as alluded to earlier, it may be convenient to introduce such hidden variables so that the maximization of l($\theta | \theta_n$) is simplified given knowledge of the hidden variables. (as compared with a direct maximization of L(θ))

4.4 Training Phase

For training purposes, our primary work is to find the mixture model λ which can stand for the feature vector of every certain image. The maximum likelihood means attempting to find the certain λ from the image which is used for training purposes in order to obtain the maximum likelihood.

For example, we extract the feature vectors $X = \{x_1, \cdots, x_T\}$ from an image by selecting the feature where the T is the number of features and the likelihood of GMM is defined as:

$$P(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \tag{47}$$

According to the fact that the function $P(X|\lambda)$ is nonlinear, ML is used to estimate the parameter of GMM until the $P(X|\lambda)$ is convergent.

Firstly, the estimation algorithm starts from an initial guess λ for the new model parameters $\overline{\lambda}$ to be estimated, in order to obtain a relationship of $P(X|\overline{\lambda}) \geq P(X|\lambda)$. Then transform the $\overline{\lambda}$ into the initial model parameter λ. This step will be repeated until $P(X|\lambda)$ is convergent. During the iteration, the following estimation calculated ensures that the approximation of GMM is achieved via the nature of monotonic increase:

$$P(i|x_i,\lambda)=\frac{\omega_i b_i(x_t)}{\sum_{k=1}^{T}\omega_k b_k(x_k)} \tag{48}$$

where the mixture weight is estimated as $\omega_i=\frac{1}{T}\sum_{t=1}^{T}p(i|x_t,\lambda)$ (49)

The estimation of mean vector is:

$$\mu_i=\frac{\sum_{t=1}^{T}p(i|x_t,\lambda)x_t}{\sum_{t=1}^{T}p(i|x_t,\lambda)} \tag{50}$$

The estimation of covariance is:

$$\Sigma_i^2=\frac{\sum_{t=1}^{T}p(i|x_t,\lambda)x_t^2}{\sum_{t=1}^{T}p(i|x_t,\lambda)}-\mu^2 \tag{51}$$

4.5 Methods of Testing

We use the maximum of a posterior criterion to differentiate all images, which means that the likelihood between testing images and pre-assigned images of each different type are calculated in order to compare the results and select the maximum numerical value. Accordingly, the testing image is ranged to a certain type, in which it has the maximum numerical value of likelihood compared with other images. In this way, we can use the equation below to describe the proposed process:

$$\hat{S}=\arg\max_{1\le k\le S}\Pr(\lambda_k|X) \tag{52}$$

where S is the total of all pre-assigned different types, $\hat{S}$ is the certain type which the testing image is classified to, λ_k is the model of pre-assigned type *K*, and *X* is the vector of features of the testing image.

This equation can be transformed to another by the Bayesian rule below:

$$\hat{S} = \arg\max_{1 \le k \le S} \frac{p(X|\lambda_k)\Pr(\lambda_k)}{P(X)} \tag{53}$$

for $\Pr(\lambda_k) = 1/S$, we have: $\hat{S} = \arg\max_{1 \le k \le S} P(X|\lambda_k)$. By calculating their logarithms, we have:

$$\hat{S} = \arg\max_{1 \le k \le S} \sum_{t=1}^{T} \log p(x_t|\lambda_k) \tag{54}$$

4.6 Extraction of Hand Features

Following the described algorithm, three significant features are extracted, which include color, shape, and the orientation histogram from all the images. Details on feature extraction are presented below.

4.6.1 Color Feature

Color is a powerful feature that often simplifies object identification and extraction from a scene. In our system, skin color detection is employed for hand detection due to the fact that human skin has a special color distribution that differs significantly from those of the background objects. The obvious advantage here is simplicity of skin detection rules that leads to construction of a very rapid classifier to identify hands in images.

Previous studies have found that skin pixels show similar chrominance components and insensitive to human races [103, 104]. Literature survey also shows that YCbCr color space is useful in segmenting skin color more accurately than RGB as the latter is sensitive to the variation of intensity [105]. In YCbCr space, luminance information is represented by a single component Y, and color information is stored as two color-difference component, Cb and Cr. The conversion from RGB to YCrCb is:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{55}$$

Recent research has also proved that the chrominance values in the skin region are narrowly distributed, which implies that the skin color is fairly uniform [106]. To this end, the narrow distribution of chrominance values are simply regulated by the presence of Cr values within [133, 173] and Cb values within [77, 127].

4.6.2 Shape

The shape information contained in an image is described using its significant edges, in which a histogram of the edge directions is employed to represent the shape attribute. The edges are extracted using the Canny edge operator, and a histogram intersection technique is used for the retrieval of corresponding histograms.

The Shape Histogram module is a type of histogram transform and can be used as part of an object classifier. A binary image is used to generate a histogram that represents the run-length values of the given image in each of 4 directions. The advantage of creating a histogram based on a shape's pixel-length span in many directions is that it reduces orientation dependency and produces a similar histogram regardless of the shape's orientation.

4.6.3 Orientation Histogram

The orientation histogram is employed as it helps provide invariant representation under changes in lighting and positions [107, 108]. As shown in Figure 4.2, comparisons between pixel representation and orientation representation are illustrated for a picture of a hand under two different lighting conditions. Though the pixel values of the hand change considerably, the orientation values remain fairly constant. The local orientation can be calculated using image gradients, and the local orientation angle θ, is defined by horizontal and vertical image pixel differences as follows

$$\theta(x, y) = \arctan\left[I(x, y) - I(x-1, y), I(x, y) - I(x, y-1)\right] \tag{56}$$

For gesture recognition, shift-invariant is a basic principle. To achieve this, we simply measure how often each orientation element occurred in the histogram. Therefore, we form a vector Φ of N elements, with the i^{th} element showing the number of orientation elements $\theta(x, y)$ between the angles $\frac{360°}{N}(i - \frac{1}{2})$ and $\frac{360°}{N}(i + \frac{1}{2})$:

$$\Phi(i) = \sum_{x,y} \begin{cases} 1, if \left|\theta(x, y) - \frac{360°}{N} i\right| < \frac{360°}{N} \\ 0, otherwise \end{cases} \tag{57}$$

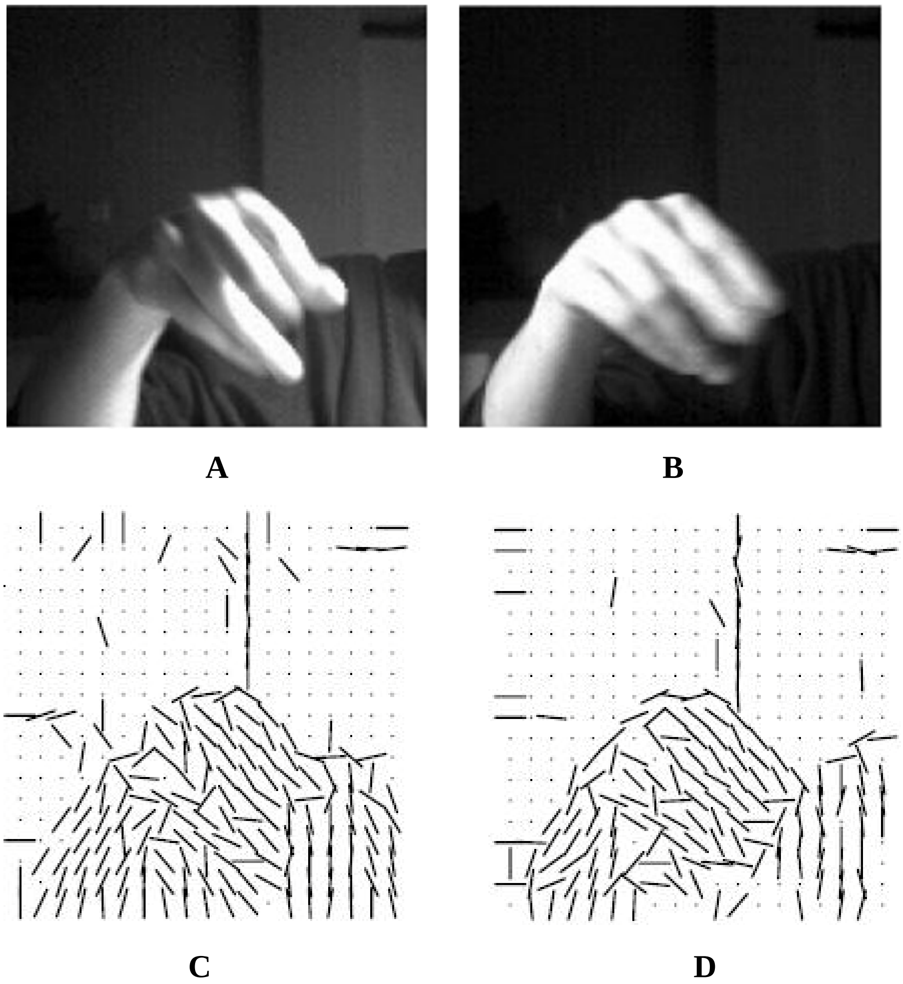


Figure 4.2: Orientation maps (C) and (D) extracted from (A) and (B) are generally more robust to lighting changes than pixel intensities.

The advantages of such representation are:

1) Very Fast: The orientation histogram can be calculated and compared in real time. Since the feature dimension is only 36, Euclidian distance can be used for the comparison of orientation histograms;
2) Robust: Since the feature considers only edge directions, it is robust to illumination changes. Even light condition changes, the edges are about the same.
3) Translation Invariant: The feature vector is insensitive to the place of the object in the image, and such translation-invariant is important in such applications.

The disadvantages here are:

1) Rotation dependent: Obviously, the orientation of edges considered in the "Orientation Histogram" is sensitive to the rotation of the object in the image.
2) Limited Vocabulary: Since two different hand positions may produce the same orientation histogram, this representation has limited the vocabulary of the recognition system. Consequently, we should first select hand gestures with different histograms. In my experiments, a vocabulary size composing more than 8 gestures may produce wrong results.
3) Object Size: The object should occupy a considerable amount of area in the image as small objects have little impact on the orientation histogram.

4.7 Experiment Results

To evaluate the proposed algorithm, a group of hand gestures are collected as the test data set, which is then classified into 3 basic classes as shown in Fig. 4.3(a). The first one is a hand with all fingers outstretched. The second one is considered as a fist, and the third one involves only 2 outstretched fingers (forefinger and middle finger) symbolizing a victory.

Following the described algorithm, from each image three significant features are extracted, including color, shape, and orientation histogram. The color feature is extracted in the YCbCr Color Space which forms a color mask as shown in Fig. 4.3(b). The shape feature, shown in Fig. 4.3(c), is extracted by Canny Edge Detector from the extracted color masks of skin pixels. Finally, the orientation histograms for each gesture calculated from the edges in Fig. 4.3(c) are illustrated in Fig. 4.3(d).

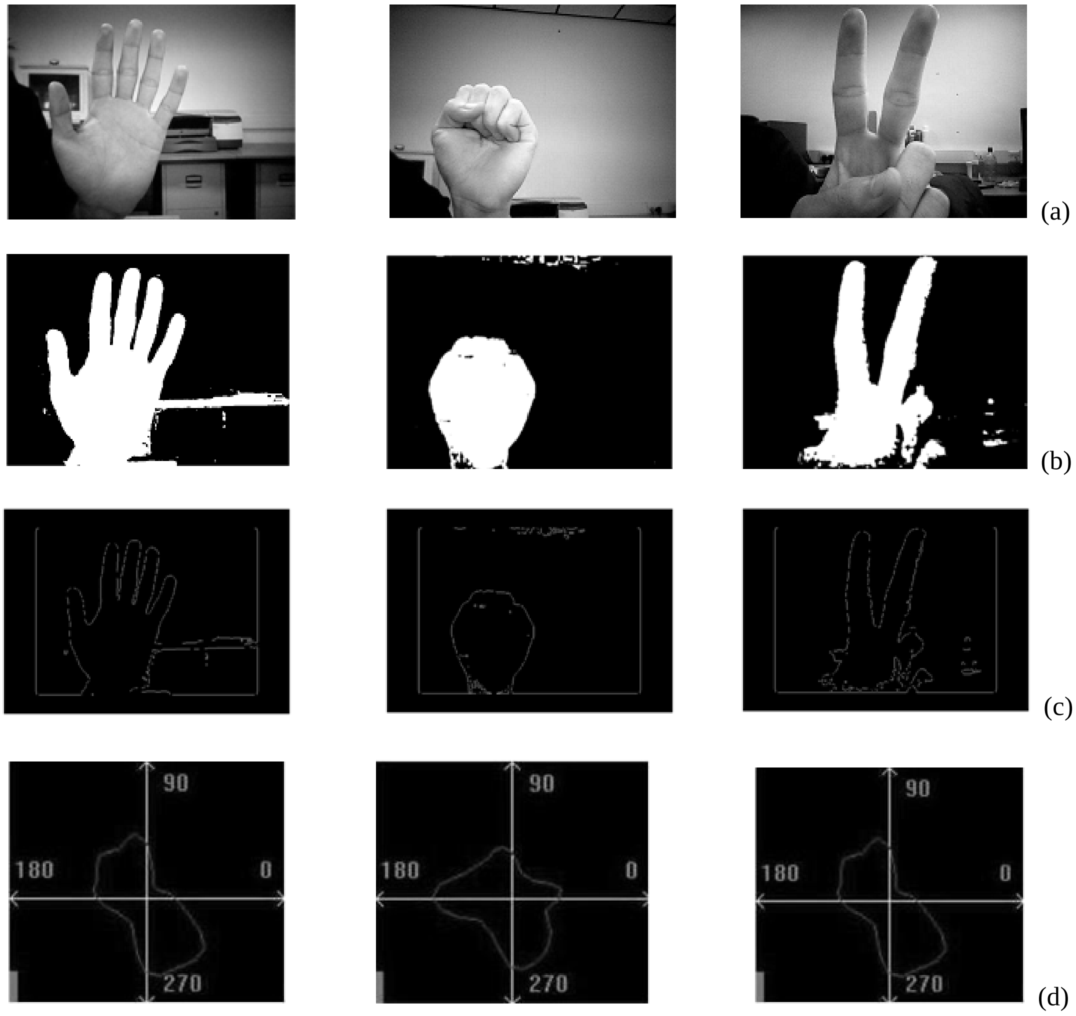


Fig 4.3 Classification of Hand Gestures: (a) Three basic classes of hand gestures; (b) color extraction of each class of hand gestures; (c) shape extraction of each class of hand gestures; (d) orientation histogram of each class of hand gestures

The performance of our algorithm is evaluated by using a total number of 2968 hand gesture images, derived from the "Sebastien Marcel Static Hand Posture Database" [109]. Since these images are captured against different backgrounds, it is useful to test the robustness of the proposed algorithm if the images in the data set are randomly organized. Each gesture is trained by more than 50 images in the database, the proposed technique is evaluated by extensive experiments and the results are measured by recognition rate as summarized in Table 1. As seen, our algorithm has successfully recognized different gestures under various lighting conditions.

Table 1: Summary of Experimental Results

	TRAINING	**TESTING**	**RECOGNITION RATE**
OUTSTRETCHED	**67**	**776**	**98.37%**
FIST	**54**	**1107**	**94.42%**
VICTORY	**59**	**905**	**90.03%**
TOTAL	***180***	***2788***	***94.27% (AVERAGE)***

4.8 Summary

We have proposed an algorithm for human gesture recognition and demonstrated its discriminative ability for recognition of gestures on a large image database. By using GMM, we have shown that multiple features extracted from gesture images could be organized to formulate new discriminating vector for classification and recognition of human gestures. The application of GMM illustrates the advantage that it provides improved performance over other existing methods, yet requiring only modest computational cost to complete the recognition. Further research can focus on the issue of extendibility and selection of primary features for improved gesture recognition, especially inside digital videos.

Chapter 5

Conclusion and Future Work

5.1 Research So Far

With rapid growth of applications within science research and business fields, interactive image processing has attracted more and more researchers. As a part of interactive imaging, object recognition such as hand gesture detection and recognition also have been greatly developed in the last decade. For human being, a multitude of familiar and novel objects can be recognized with little effort, despite the fact that these objects may vary in form, color, texture, etc, and even with partial occlusion. For machine learning algorithms, this is still a serious challenge since all of the aspects cannot be defined as the way we identify objects in the real world. Recent research in this area focuses on two main aspects: Features Extraction and Similarity Measurements. Based on the two main aspects of research, this thesis research is carried out in the specific area of hand gesture detection and recognition.

At first, Introduction to hand gestures recognition systems and the basic principles of digital image processing are reviewed in Chapter 1 and Chapter 2.

Then, my primary work is introduced in the next two chapters. In Chapter 3, design of hand motion detection based on segmentation is represented. To detect the motion of hand gestures conversion accurately, our proposed system contains two parts: Segmentation of hand gestures and extraction of background. Proposed segmentation method compares current frame with previous frame in order to generate difference between them. Background modelling greatly reduces the interference from complex background.

In Chapter 4, a competitive methodology is proposed on the basis of Gaussian

Mixture Model and Expectation Maximization Algorithm. GMM takes its advantage in describing density distribution of any structure. In order to calculate the similarity value in GMM, EM Algorithm is introduced. In our GMM model, the missing or hidden data could be considered as limited features and the limitation of presentation of features by proposed descriptors. An excellence of EM Algorithm is divided into two main steps: Expectation Step and Maximization Step. On the E-Step, we expect the target should be closed to a guess parameter and calculate the likelihood. On the M-Step, the guess parameter is modified iteratively in order to maximize the possible likelihood. At last, the target will be classified to one of the guess parameters which has the maximum likelihood with a target. Afterwards, an experiment procedure is introduced. Within the training phase, each GMM of hand gestures is acquired. This step ensures that each type of hand gestures obtains unique parameters standing for its Gaussian Mixture Model. At last, testing section begins with calculating the value of likelihood between testing images and pre-assigned images.

5.2 Thesis Contributions

At first, in the thesis work, segmentation based motion detection is proposed. Frame differencing is employed to compare the current frame and previous frame in order to generate the value of difference. As background noises will reduce the performance of frame differencing and slow motion will not be detected during the processing of frame differencing, background modelling is proposed. With background modelling, noises are extracted and the current frame is compared with average image which is calculated from the sequence of neighbouring images. Results show that background modelling greatly increases the performance of motion detection.

Secondly, Gaussian Mixture Model is a mature approach which is widely used in voice recognition, but this is the first time that GMM is successfully employed to recognize hand gestures. Compared with other current hand gesture recognition systems, GMM has its advantages in calculating the likelihood probability of gestures using various mixed features. One research paper has been published that describes the hand gesture recognition based on GMM and EM Algorithm [110].

5.3 Further Work

Although the thesis implemented a system for hand gesture recognition, there are still many other challenges in this area that have not been addressed, which will motivate our future work as summarized below.

1) In the proposed system, color, shape and edge directions form the features. As the edge orientation histogram is sensitive to rotations and has less discriminative ability, investigation on new features are desired to solve these problems and also increasing vocabulary in the system under new features;
2) The images have comparatively pure background for easy segmenting hands in images. In a further stage, complicated background in real scenes need to be tested hence a fine segmentation method is desired;
3) Currently, the approach only works fine in images, which can be extended to work in digital videos and possibly in the compressed domain for more applications such as content-based video annotation, indexing and retrieval.

References

[1] V. Athitsos, S. Sclaroff, “Estimating 3d hand pose from a cluttered image”, Proceeding of Conference on Computer Vision and Pattern Recognition (CVPR '03), vol. 1, Madison, WI, 2003.

[2] P. H. S. Torr, B. Stenger, A. Thayananthan and R. Cipolla, “Hand pose estimation using hierarchical detection”, Proceeding of International Workshop on Human-Computer Interaction, 2004.

[3] C. Gurrapu and V. Chandran, “Gesture Classification Using A GMM Front End and Hidden Markov Models”, Proceedings of the 3rd IASTED International Conference on Visualization, Imaging, and Image Processing, pp. 609-612, Spain, September, 2003.

[4] Y. Liu and Y. Jia, “A Robust Hand Tracking and Gesture Recognition Method for Wearable Visual Interfaces and Its Applications”, Proceedings of the Third International Conference on Image and Graphics, IEEE Computer Society, pp. 472-475, USA, December 2004.

[5] S. Marcel, O. Bernier, J. Viallet and D. Collobert, “Hand Gesture Recognition using Input-Output Hidden Markov Models”, Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, pp. 456-461, USA, 2000.

[6] E. Hunter, J. Schlenzig and R. Jain, “Posture estimation in reduced-model gesture input systems”, Proceedings of the International Workshop on Automatic Face and Gesture Recognition, pp. 290-295.

[7] H. K. Kim, J. D. Kim, D. G. Sim and D. I. Oh, “A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval”, Proceeding of 2000 IEEE International Conference on Multimedia and Expo, vol. 1, pp. 307-310.

[8] M. C. Su, W. F. Jean and H. T. Chang, “A static hand gesture recognition system using a composite neural network”, Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, vol. 2, pp. 786-792.

[9] D. S. Banarse and A. W. G. Duller, “Deformation invariant pattern classification for recognizing hand gestures”, Proceeding of IEEE International Conference on Neural Networks, vol. 3, pp. 1812-1817.

[10] Y. Wu and T. S. Huang, “View-independent recognition of hand postures”, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2000), vol. 2, pp. 88-94.

[11] H. Zhou, D. J. Lin and T. S. Huang, “Static Hand Gesture Recognition based on Local Orientation Histogram Feature Distribution Model”, Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04),

Volume 10, pp. 161, June 27-July 02, 2004.

[12] C. Schwarz and N. da V. Lobo, “Segment-Based Hand Pose Estimation”, Proceedings of the 2nd Canadian conference on Computer and Robot Vision, pp. 42-49, May 09-11, 2005.

[13] Z. Y. Mo and U. Neumann, “Real-time Hand Pose Recognition Using Low-Resolution Depth Images”, Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1499-1505, June 17-22, 2006.

[14] A. Just, Y. Rodriguez and S. Marcel, “Hand Posture Classification and Recognition using the Modified Census Transform”, Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, pp. 351-356, April 10-12, 2006.

[15] X. Wang and G. Dai, “A new invariant descriptor for shape representation and recognition”, Proceedings of IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007), pp. 48-51.

[16] R. Veltkamp, “Content-based image retrieval system: A survey” Technical Report, University of Utrecht, Utrecht, The Netherlands, 2002.

[17] Y. Rui, T. Huang, S. Mehrotra and M. Ortega, “A relevance feedback architecture for content-based multimedia information retrieval systems”, in Proceeding of IEEE Workshop Content-Based Access of Image and Video Libraries, pp. 92–89, 1997.

[18] E. Chang, B. T. Li, G. Wu and K. Goh, “Statistical learning for effective visual information retrieval”, Proceeding of IEEE International Conference on Image Processing, pp. 609–612, Barcelona, Spain, September 2003.

[19] J. Kramer and L. Leifer, “The talking glove: An expressive and receptive verbal communication aid for the deaf, deaf-blind, and nonvocal”, Proceeding of 3rd Annual Conference on Computer Technology, Special Education, Rehabilitation, pp. 335–340, Northridge, CA, October 1987.

[20] J. Kramer and L. Leifer, “The talking glove: A speaking aid for nonvocal deaf and deafblind individuals”, Proceeding of RESNA 12th Annual Conference, pp. 471–472, New Orleans, LA, 1989.

[21] S. S. Fels and G. E. Hinton, “Glove-Talk: A neural network interface between a Data-Glove and a speech synthesizer”, IEEE Transactions on Neural Networks, vol. 1, No. 1, pp. 2–8, 1993.

[22] K. Kamata, T. Yoshida, M. Watanabe and Y. Usui, “An approach to Japanese sign language translation system”, in IEEE International Conference on Systems, Man, Cybernetics, pp. 1089–1090, 1990.

[23] K. Murakami and H. Taguchi, “Gesture recognition using recurrent neural networks”, Proceeding of CHI’91, pp. 237–242, 1991.

[24] K. Morimoto, T. Izuchi, E. Fujishige, S. Watanabe, T. Morichi and T. Kurokawa, “Analysis of spatio-temporal structure of sign language for its machine translation”, 8th Symposium Human Interface, pp. 621–626, 1992.

[25] M. C. Su, “A speaking aid using neural networks for the deaf,” Biomedical, Engineering Application, Basis, Communications, Vol. 8, No. 4, pp. 33–39, 1996.

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

[27] J. D. Foley, A. V. Dam, S. K. Feiner and J. F. Hughes, “Computer graphics: principles and practice”, 2nd edition, Reading, Mass, Addison-Wesley, 1990.

[28] J. Huang, S. R. Kumar, M. Metra, W. J. Zhu and R. Zabith, "Spatial color indexing and applications", International Journal of Computer Vision, Vol.35, No.3, pp. 245-268, 1999.

[29] J. Huang, "Image indexing using color correlogram", IEEE International Conference on Computer Vision and Pattern Recognition, pp. 762-768, Puerto Rico, June 1997.

[30] M. Ioka, "A method of defining the similarity of images on the basis of color information", Technical Report RT-0030, IBM Tokyo Research Laboratory, Tokyo, Japan, November 1989.

[31] A. K. Jain, “Fundamental of Digital Image Processing”, Englewood Cliffs, Prentice Hall, 1989.

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

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

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

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

[36] H. J. Zhang, "Image retrieval based on color features: An evaluation study", SPIE Conference on Digital Storage and Archival, Pennsylvania, October 25-27, 1995.

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

[38] A. K. Jain, “Fundamental of Digital Image Processing”, Englewood Cliffs, Prentice Hall, 1989.

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

[40] Y. Gong, H. J. Zhang and T. C. Chua, "An image database system with content capturing and fast image indexing abilities", IEEE International Conference Proceeding on Multimedia Computing and Systems, pp. 121-130, Boston, 14-19 May 1994.

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

[42] T. Gevers and A. W. M. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval", IEEE Transactions on Image Processing, Vol.9, No.1, pp. 102-119, 2000.

[43] G. D. Finlayson, "Color in perspective", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.8, No. 10, pp. 1034-1038, October 1996.

[44] T. Gevers and A. W. M. Smeulders, "Content-based image retrieval by viewpoint-invariant image indexing", Image and Vision Computing, Vol.17, No.7, pp. 475-488, 1999.

[45] T. Chang, and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform", IEEE Transactions on Image Processing, vol. 2, No. 4, pp. 429-441, October 1993.

[46] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis", IEEE Transactions on Information Theory, Vol.36, pp. 961-1005, September 1990.

[47] J. M. Francos, "Orthogonal decompositions of 2D random fields and their applications in 2D spectral estimation", N. K. Bose and C. R. Rao, editors, Signal Processing and its Application, pp. 20-227, North Holland, 1993.

[48] J. M. Francos, A. A. Meiri and B. Porat, "A unified texture model based on a 2d Wold like decomposition", IEEE Transactions on Signal Processing, pp. 2665-2678, August 1993.

[49] J. M. Francos, A. Narasimhan and J. W. Woods, "Maximum likelihood parameter estimation of textures using a Wold-decomposition based model", IEEE Transactions on Image Processing, pp. 1655-1666, December 1995.

[50] T. Gevers and A. W. M. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval", IEEE Transactions on Image Processing, Vol. 9, No. 1, pp. 102-119, 2000.

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

[52] A. Kankanhalli, H. J. Zhang and C. Y. Low, "Using texture for image retrieval", Third International Conference on Automation, Robotics and Computer Vision, pp.

935-939, Singapore, November 1994.

[53] W. J. Krzanowski, “Recent Advances in Descriptive Multivariate Analysis”, Chapter 2, Oxford Science Publications, 1995.

[54] A. Laine and J. Fan, "Texture classification by wavelet packet signatures", IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1186-1191, November 1993.

[55] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval", IEEE Transactions on Pattern Analysis and Machine Learning, Vol. 18, No. 7, July 1996.

[56] W. Y. Ma and B. S. Manjunath, "A comparison of wavelet features for texture annotation", Proceeding of IEEE International Conference on Image Processing, Vol. II, pp. 256-259, Washington D. C., October 1995.

[57] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, pp. 674-693, July 1989.

[58] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, pp. 837-842, August 1996.

[59] J. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models", Pattern Recognition, Vol. 25, No. 2, pp. 173-188, 1992.

[60] T. Ojala, M. Pietikainen and D. Harwood, "A comparative study of texture measures with classification based feature distributions", Pattern Recognition, Vol. 29, No. 1, pp. 51-59, 1996.

[61] R. W. Picard, T. Kabir and F. Liu, "Real-time recognition with the entire Brodatz texture database", Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition, pp. 638-639, New York, June 1993.

[62] H. Tamura, S. Mori and T. Yamawaki, "Texture features corresponding to visual perception", IEEE Transactions on Systems, Man, and Cybernetics, vol. Smc-8, No. 6, June 1978.

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

[64] P. Brodatz, "Textures: A photographic album for artists & designers", Dover, NY, 1966.

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

[66] W. Niblack et al., "Querying images by content, using color, texture, and shape", SPIE Conference on Storage and Retrieval for Image and Video Database, Vol. 1908, pp. 173-187, April 1993.

[67] A. Pentland, R. W. Picard and S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases", Proceeding of Storage and Retrieval for Image and Video Databases II, Vol. 2185, San Jose, CA, USA, February 1994.

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

[69] W. I. Grosky and R. Mehrotra, "Index based object recognition in pictorial data management", CVGIP, Vol. 52, No. 3, pp. 416-436, 1990.

[70] H. V. Jagadish, "A retrieval technique for similar shapes", Proceeding of International Conference on Management of Data, SIGMOID'91, pp. 208-217, Denver, CO, May 1991.

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

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

[73] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, No. 6, pp. 545-561, June 1995.

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

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

[76] E. Persoon and K. Fu, "Shape discrimination using Fourier descriptors", IEEE Transactions on System, Man, and Cybernation, Vol. 7, pp. 170-179, 1977.

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

[78] L. Yang and F. Algregtsen, "Fast computation of invariant geometric moments: A new method giving correct results", Proceeding of IEEE International Conference on Image Processing, 1994.

[79] R. C. Veltkamp and M. Hagedoorn, "State-of-the-art in shape matching", Technical Report UU-CS-1999-27, Utrecht University, Department of Computer Science, September 1999.

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

[81] S. K. Chang, Q. Y. Shi and C. Y. Yan, "Iconic indexing by 2-D strings", IEEE Transactions on Pattern Analysis, Machine Intelligence, Vol. 9, No. 3, pp. 413-428, May 1987.

[82] S. K. Chang, E. Jungert and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2D string", Technical Report, University of Pittsburgh, 1988.

[83] S. Y. Lee and F. H. Hsu, "2D C-string: a new spatial knowledge representation for image database systems", Pattern Recognition, Vol. 23, pp. 1077-1087, 1990.

[84] S. Y. Lee, M.C. Yang and J. W. Chen, "2D B-string: a spatial knowledge representation for image database system", Proceeding of ICSC'92 Second International computer Science Conference, pp. 609-615, 1992.

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

[86] V. N. Gudivada and V. V. Raghavan, "Design and evaluation of algorithms for image retrieval by spatial similarity", ACM Transactions on Information Systems, Vol. 13, No. 2, pp. 115-144, April 1995.

[87] M. Stricker and M. Orengo, "Color indexing with weak spatial constraint," Proceeding of SPIE Conference on Visual Communications, 1996.

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

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

[90] R. J. Lopes, A. T. Lindsay and D. Hutchison, “The utility of MPEG-7 systems in audio-visual applications with multiple streams”, IEEE Transactions of Circuits System and Video Technology, vol. 13, pp. 16–25, January 2003.

[91] “Multimedia content description Interface – Part 3: Visual”, ISO/IEC, Technical Report 15 938-3:2001, 1st edition, 2001.

[92] P. Salembier and J. R. Smith, “MPEG-7 multimedia description schemes”, IEEE Transactions Circuits System and Video Technology, vol. 11, pp. 748–759, June 2001.

[93] C. Y. Chiu, H. C. Lin and S. N. Yang, “Learning user preference in a personalized CBIR system” in Proceeding of 16th International Conference on Pattern Recognition,

vol. 2, pp. 532–535, 2002.

[94] Y. Rui, T. S. Huang and S. Mehrotra, “Content-based image retrieval with relevance feedback in MARS” in Proceeding of IEEE International Conference on Image Processing, vol. 2, pp. 815–818, 1997.

[95]Y. Ishikawa, R. Subramanya and C. Faloutsos, “Mindreader: Querying databases through multiple examples” in Proceeding of International Conference on Very Large Data Bases, pp. 218–227, New York, 1998.

[96] I. J. Cox, L. Miller, P. Minka, V. Papthomas and P. Yianilos, “The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments”, IEEE Transactions on Image Process, vol. 9, no. 1, pp. 20–37, January 2000.

[97] X. Zhou and T. S. Huang, “Small sample learning during multimedia retrieval using Biasmap” in Proceeding of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 11–17, 2001.

[98] D. Tao, X. Tang, X. Li and Y. Rui, “Direct kernel biased discriminant analysis: A new content-based image retrieval relevance feedback algorithm”, IEEE Transactions on Multimedia, vol. 8, no. 4, pp. 716–727, April 2006.

[99] C. M. Bishop, “Neural Networks for Pattern Recognition”, Oxford University Press, 1995.

[100] R. O. Duda, P. E. Hart and D. G. Stork, “Pattern Classification”, volume 2, Wiley-Interscience Publication, 2000.

[101] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via EM algorithm”, Journal of the Royal Statistical Society, Series B (Methodological), Vol. 39, No.1, pp. 1-38, 1977.

[102] J. Bilmes, “A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models”, Technical Report ICSI-TR-97-021, University of California Berkeley, 1998.

[103] D. Chai and K. N. Ngan, “Face segmentation using skin-color map in videophone applications”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 4, pp. 551-564, June 1999.

[104] D. Chai and K. N. Ngan, “Locating facial region of a head-and-shoulders color image”, in Third IEEE International Conference on Automatic Face and Gesture Recognition (FG’98), pp. 124-129, Nara, Japan, April 1998.

[105] Y. M. Hsuan, D. J. Kriegman and N. Ahuja, "Detecting faces in images: a survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 34-58, 2002.

[106] D. Chai and K. Ngan, “Face segmentation using skin-color map in videophone applications”, IEEE Transactions on Circuits and Systems for Video Technology, Volume 9, Issue 4, pp. 551 – 564, June 1999.

[107] W. T. Freeman and M. Roth, “Orientation Histograms for Hand Gesture Recognition”, IEEE International Workshop on Automatic Face and Gesture Recognition, pp. 296-301, IEEE Computer Society, Switzerland, June 1995.

[108] H. J. Lee and J. H. Chung, “Hand gesture recognition using orientation histogram”, Tencon 99 Proceedings of the IEEE Region 10 Conference, Vol. 2, pp. 1355-1358, South Korea, December 1999.

[109] S. Marcel, “Hand posture recognition in a body-face centered space”, Proceedings of the Conference on Human Factors in Computer Systems, pp.302-303, ACM, USA, 1999.

[110] J. Jia, J. Jiang, “Recognition of Hand Gesture Based on Gaussian Mixture Model”, the Sixth International Workshop on Content-Based Multimedia Indexing, pp. 353-356, London, June 2008.