

*Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science*



Hand Palm Recognition Using Wavelet Transform

*A Thesis Submitted to the College of Science, Al-Nahrain
University in Partial Fulfillment of the Requirements for
The Degree of Master of Science in Computer
Science*

by
Aunss Sinan Maki
(B.Sc. 2004)

Supervised By
Dr. Loay E. George

November 2007

Dhu'l-Qa'da 1428

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

وَيَسْأَلُونَكَ عَنِ الرُّوحِ قُلِ الرُّوحُ مِنْ أَمْرِ رَبِّي وَمَا

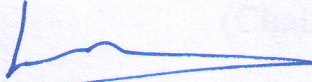
أُوتِيتُمْ مِّنَ الْعِلْمِ إِلَّا قَلِيلًا

صدق الله العظيم
سورة الأسراء (الآية 85)

Supervisor Certification

I certify that this thesis was prepared under ~~my~~ supervision at the Department of Computer Science/College of Science/Al-Nahrain University, by **Aunss Sinan Maki Kubba** as partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature:



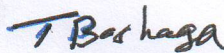
Name : **Dr. Loay E. George**

Title : **Senior Research**

Date : **13 / 9 / 2007**

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:



Name : **Dr. Taha S. Bashaga**

Title : **Head of the department of Computer Science,
Al-Nahrain University.**

Date : **13 / 9 / 2007**

Certification of the examination Committee

We chairman and members of the examination committees, certify that we have studied this thesis "*Hand Palm Recognition Using Wavelet Transform*" presented by the student *Aunss Sinan Maki* and examined her in its contents and that we have found it worthy to be accepted for the degree of Master of Science in computer science with excellent degree.

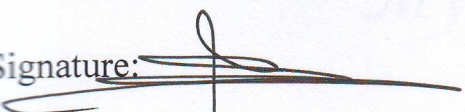
Signature: 

Name: **Dr. Ayad A. Al-Ani**

Title: **Assistant Professor**

Date: *14/ 11/ 2007*

(Chairman)

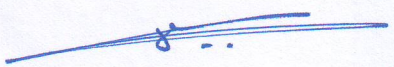
Signature: 

Name: **Dr. Amir S. Al-Malah**

Title: **Assistant Professor**

Date: *1 / / 2007*

(Member)

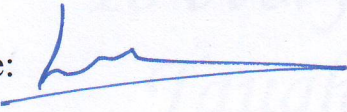
Signature: 

Name: **Dr. Sawsan Kamal Thamer**

Title: **Lecture**

Date: *18/ 11/ 2007*

(Member)

Signature: 

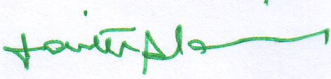
Name: **Dr. Loay E. George**

Title: **Senior Research**

Date: *13 / 11 / 2007*

(Supervisor)

Approved by the Dean of the Collage of Science, Al-Nahrain University.

Signature: 

Name: **Dr. LAITH ABDUL AZIZ AL-ANI**

Title: **Assistant Professor**

Date: *9 / 14 2007*

(Dean of collage of science)



DEDICATED TO

My Parents ...

My Sisters ...

My friends...

To everyone

Taught me a letter

Aunss



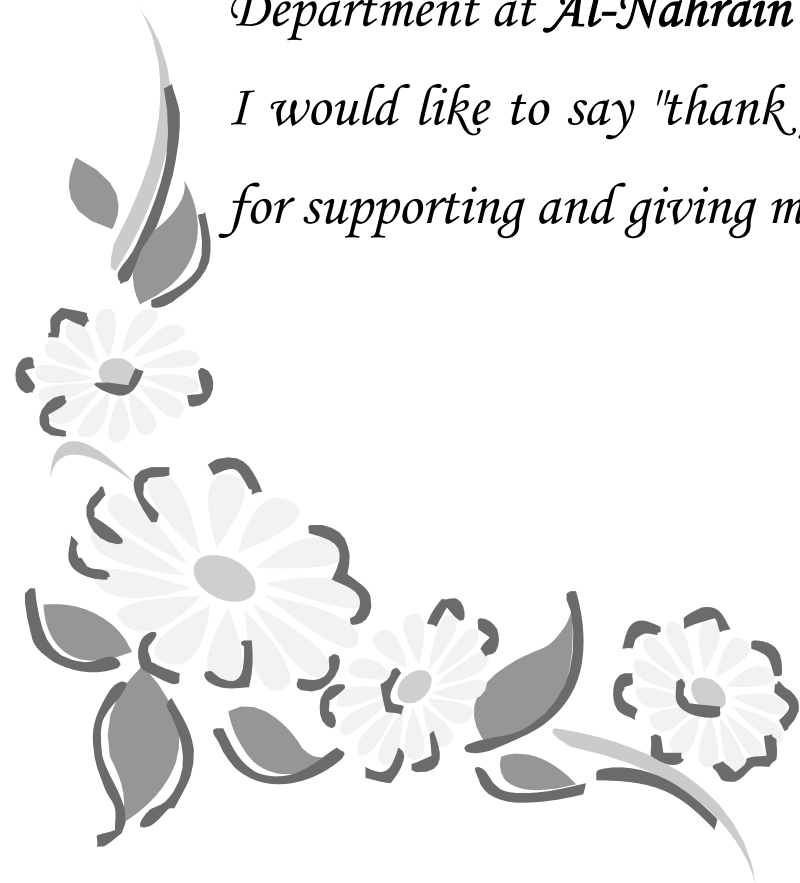
Acknowledgment

I would like to express my sincere appreciation to my supervisor, Dr. Loay E. George, for giving me the major steps to go on to explore the subject, sharing with me the ideas in my research "Hand Palm Recognition Using Wavelet Transform" and discuss the points that I felt they are important.

Grateful thanks for the Head of Department of Computer Science, Dr. Taha S. Bashaga,

Also, I wish to thank the staff of Computer Science Department at Al-Nahrain University for their help.

I would like to say "thank you" to my faithful friends for supporting and giving me advises.



Abstract

Biometric identification is an emerging technology that can solve security problems in our networked society. Biometrics traits such as fingerprints, hand geometry, face and voice verification provide a reliable alternative for identity verification / identification and are gaining commercial and high user acceptability rate. In this research, a hand palm identification system has been introduced. The work flow of the introduced system consists of two stages. The first stage is called the preprocessing stage and the second stage is called matching stage. In the preprocessing stage a number of color images samples of the right hand for some persons have been captured. In both phases (preprocessing and matching) each color input image was converted to binary image by using thresholding method, then the boundary image extracted by using Laplace edge detector. In order to extract the hand palm area the sequence of the hand boundary point was collected by using chain coding algorithm. The last step in the preprocessing stage is the determination of the joint points which are required to separate the palm region from the overall hand area. After the joint points have been determined then Bresenham algorithm was implemented to connect the joint points by lines. Then seed filling algorithm are used to mark the palm pixels as white colored pixels, and finally the white color of the marked pixels was replaced with the original color of these pixels, the output of this stage was color palm region. Haar wavelet transform had been applied to determine the Haar wavelet coefficients. Then features of the palm area were extracted by using two types of discriminating features (i.e., complex moment and invariant moment which are stored as feature vectors in a database file. Also, the mean and standard deviation of the feature vectors had been determined to be used as template in the matching stage. In the matching stage the identification process was based on using Euclidean distance measure as a similarity criterion. The test results indicate that the identification rates when using complex moments features as discriminating features are much better than when using invariant moments features, and the best identification rates when using one pass Haar wavelet transform have reached 88% using complex moments features and 78% using invariant moments features, but when using two pass Haar wavelet transform the best identification rate have reached 96% when using complex moments features and 76% when using invariant moments features.

List of Abbreviations

Abbreviation	Meaning
BMP	Bitmap Pixel
DCT	Discreet Cosine Transform
DWT	Discrete Wavelet Transform
HH	High – High band coefficient
HL	High – Low band coefficient
HWT	Haar Wavelet Transform
ID	Identity
LH	Low – High band coefficient
LL	Low – Low band coefficient
PIN	Personal Identification Numbers
RGB	Red Green Blue
WT	Wavelet Transform

Table of Contents

Chapter One

General Introduction

1.1 Introduction.....	1
1.2 Definition of Biometric.....	2
1.3 Biometric Technologies	2
1.4 The Difference between Identification and Verification	4
1.5 Aim of Thesis	6
1.6 Literature Survey	6
1.7 Thesis Layout.....	9

Chapter Two

Theoretical Overview

2.1 Introduction.....	10
2.2 Pattern Recognition	10
2.3 Type of Images	11
2.3.1 Binary image.....	12
2.3.2 Gray Image	12
2.3.3 Color Image	12
2.3.4 Multi-Spectral Image	13
2.4 Hand Palm Recognition / Verification System.....	13
2.4.1 Image Capture.....	13
2.4.2 The Preprocessing Stage	14
A. Thresholding	14
B. Enhancement	15
C. Edge Detection	15
D. Chain Coding	16
E. Edge Linking Algorithm	18
2.4.3 Feature Extraction.....	18
2.4.4 Feature Analysis and Matching	19
2.5 Wavelet Transform	20
2.6 Haar Wavelet Transform	21
2.7 Moments	23

Chapter Three

Design and Implementation

3.1 Introduction.....	26
3.2 The Overall System Stages	26
3.3 Preprocessing Stage	28
3.3.1 Image Binarization.....	28
3.3.2 Smoothing and Erosion.....	31
3.3.3 Edge Detection.....	32
3.3.4 Chain Coding for Boundary Tracing	34
3.4 Extraction of Palm Area	36
3.4.1 Joint Points Detection	36
3.4.2 Bresenham Algorithm.....	41
3.4.3 Seed Filling	43
3.4.4 Hand Span Detection	45
3.5 Haar Wavelet Transform	46
3.6 Moments	48
3.7 Feature Analysis	52

Chapter Four

Tests and Results

4.1 Introduction.....	61
4.2 Test Material	61
4.3 Experimental Results	63
4.3.1 The Results of the Preprocessing Stage.....	63
4.3.2 The Results of Hand Palm Area Extraction.....	64
4.3.3 The Templates Features Vectors	66
4.3.4 Identification Rates for one level Haar Wavelet Transform.....	77

Chapter Five

Conclusions and Future Work

5.1 Conclusions.....	78
5.2 Future Work	79

Chapter One

Chapter One

General Introduction

1.1 Introduction

User authentication is vitally important for the security of computer networks. Designing a high-security user-authentication system still remains an unsolved problem. The conventional means of identification, (e.g. passwords and Personal Identification Numbers (PINs)), can be easily guessed, observed, or forgotten. However, several of the biometric characteristics of an individual are unique, easily observed and time invariant. These properties make biometrics well suited for the purposes of user identification [Rho04].

Hand geometry-based verification systems are not new and have been available since 1970. Hand geometry refers to the geometric structure of the hand; this structure includes width of the fingers at various locations, width of the palm, thickness of the palm, length of the fingers, etc [Jai99].

The hand geometry offers a good balance of performance characteristics and is relatively easy to use. The hand images can be obtained using a simple setup including a low-cost web cam. However, other biometric traits require specialized, high-cost scanners to acquire the data. User acceptability for hand geometry based biometrics is very high as it does not extract detail features of the individual. Thus, for applications where the biometric features are needed to be distinctive enough for verification, hand geometry can be used. Further, hand geometry features can be easily combined with other biometric traits [Roy98].

1.2 Definition of Biometric

Biometric technologies become the foundation of an extensive array of highly secure identification and personal verification solutions instead of the previous methods of identification and verification.

A biometric system is a pattern recognition system that recognizes a person based on a feature vector derived from a specific physiological or behavioral characteristic that the person possesses. Examples of physiological characteristics that are used in biometric devices include fingerprints, hand-geometry, face features, palm prints and patterns within the iris or retina, or in the layout of the veins. Behavioral characteristics include voice pattern, gait (the manner in which a person walks), and the dynamics of handwriting (like signature) or keystrokes [Roy98].

1.3 Biometric Technologies

There are many kinds of biometric technologies; the most familiar ones are shown below: [Woo03]

1. **Fingerprint:** This analyzes the patterns of lines, known as friction ridges and the discontinuity in the ridge flow that give rise to feature points called minutiae. fingerprint recognition can achieve good accuracy sufficient for both verification and identification. It is low cost and compact and is getting popular as consumer products. However, not everyone has fingerprints that can be recognized. The sensor is also not able to capture acceptable quality fingerprint images for people with very wet and very dry skin. In addition, the sensor needs to be maintained properly in order to get consistent performance
2. **Face:** This analyzes facial features such as an individual's eyes, nose, mouth and cheekbones, and map their proportions and locations relative to each other. The disadvantage is that the accuracy achievable, it is only suitable for verification, but is still insufficient for identification. The

performance will also be affected by variation in face due to aging, make-up, hair-style, glasses, pose and lighting condition in addition to not being able to separate twins.

3. Eye: its consist of iris scan that analyzing features of colored ring of the eye and retinal scan that analyzing blood vessels in the eye. Iris recognition is very accurate with very low false acceptance rate and can be applied to both verification and identification. But it suffers from poor lighting, reflection and possibly glasses and may not be suitable for people with cataract and young children. In addition, some imaging system will require the user to be motionless for a while.
4. Voice: Speech contains information about the identity of the speaker. A speech signal reflect the voice features that depend on the spoken language, the presence and type of speech pathologies, the physical and emotional state of the speaker. Humans are able to extract the identity information when the speech comes from a speaker they are acquainted with. voice varies with age and there can be drastic change from childhood to adolescence. Also illness and emotion may affect the voice as well as room acoustics and environmental noise. Variation in microphones and channel mismatch (use of different type and quality of microphones) is also a major problem for the widespread use of this biometric technology.
5. Key stroke dynamics: Keystroke dynamics is an automated method of examining an individual's keystrokes on a keyboard. This technology examines such dynamics as speed and pressure, the total time taken to type particular words, and the time elapsed between hitting certain keys. This biometric could be used to verify the computer user's identity continuously. The disadvantage of key stroke is has low performance, low permanence, and low uniqueness.

6. Hand geometry: it measures the fingers' length, fingers' width, thickness, curvatures and their relative geometry measures. Hand geometry has acceptable accuracy for verification but not sufficient for identification. The major advantage is that most people can use it and as such, the acceptance rate is good. However, the system is rather bulky and may have problems with aging and health condition such as arthritis
7. Palm print: it's close to fingers and it's used a sensor or scanner to obtain a representation of the pattern Ridges, valleys and other minutiae data on the palm as with finger images.
8. Signature: it's an automated method of measuring an individual's signature. This technology examines such dynamics as speed, direction, and pressure of writing; the time that the stylus is in and out of contact with the paper, the total time taken to make the signature; and where the stylus is raised from and lowered onto the paper. The disadvantage of signature is has low performance, low permanence, low uniqueness and low circumvention.

1.4 The Difference between Identification and Verification

Pattern recognition consists of *identification* and *verification*. The terms differ significantly, *in verification systems*, the step after enrollment is to verify that a person is who he / or she claims to be (i.e., the person who enrolled). After the individual provides an identifier, the biometric is presented, where the biometric system captures the identifier features and generates a trial template that is based on the vendor's algorithm, Then the biometrics system compares the trial biometric template with the person's reference template, which should be stored in the system during enrollment phase, to determine whether the individual's trial and stored templates match.

Verification is often referred to as 1:1 (one-to-one) matching, verification systems can contain databases ranging from dozens to millions of enrolled templates but are always predicated on matching an individual's presented biometric against his or her reference template. Nearly all verification systems can render a match–no-match decision in less than a second. Systems that require employees to authenticate their claimed identities before granting them access to secure buildings or to computers are verification applications. [Rho04]

In identification systems, the step after enrollment is to identify who the person is. Unlike verification systems, no identifier is provided. To find a match, instead of locating and comparing the person's reference template against his or her presented biometric, the trial template is compared against the stored reference templates of all individuals enrolled in the system. Identification systems are referred to as 1: N (one-to-N, or one-to-many) matching because an individual's biometric is compared against multiple biometric templates in the system's database.

There are two types of identification systems: positive and negative.

Positive identification systems are designed to ensure that an individual's biometric is enrolled in the database. The anticipated result of the search is a match. A typical positive identification system controls access to a secure building or secure computer by checking anyone who seeks access against a database of enrolled employees. The goal is to determine whether a person seeking access can be identified as having been enrolled in the system.

Negative identification systems are designed to ensure that a person's biometric information is not present in a database. The anticipated result of the search is a non match. Comparing a person's biometric information against a database of all who are registered in a public benefits program, for

example, can ensure that this person is not “double dipping” by using fraudulent documentation to register under multiple identities. [Rih01]

1.5 Aim of Thesis

The aim of the thesis is to design a hand palm recognition system that uses wavelet transform and moments to extract the hand-palm features, these biometrics features are used to identify the enrolled persons, these features should be stored in the template system to be used later in the identification and verification systems to identifying unknown person who should previously enrolled by the established biometric system.

1.6 Literature Survey

- Zhang D. [Zha98], he presented two novel palm print characteristics, datum point invariance and line feature matching. The datum points of palm print, have a remarkable advantage of invariable location, they have been determined by using the developed directional projection algorithm. Then, he implemented the line feature extraction method, and presented the line matching method to detect whether a couple of palm prints are from the same palm. Various palm print images have been tested to illustrate the effectiveness of the palm print verification by using both characteristics.
- You, et al (2002), [You02], their paper describes a method to authenticate individuals based on palm print identification and verification. The concept of texture features and interesting point was introduced to define palm print features. A texture based dynamic selection scheme was proposed to facilitate the fast search for the best matching of the sample in the data base in a hierarchal fashion. The global texture energy, which is characterized with high convergence

of inner palm similarities and good dispersion of inter palm discrimination, was used to guide the dynamic selection of a small set of similar candidate from the database at coarse level for further processing. The referred tests results demonstrate the effectiveness and accuracy of the proposed method.

- Ekinici, et al (2003), [Eki03], they presented a new biometric approach for the personal identification based on palm print recognition. Palm print-based classification consists of two main tasks; image processing, and implementation tasks. The image processing task performs automated extraction of region of interest (i.e., palm print area while ignoring fingers, wrist, and background if any), and then detect of structural features (such as lines, in the palm) because such features can characterize a palm exactly. The enrollment and authentication modules use the principal features of the palm (obtained by the image processing task) to produce morphological model based on relational graph structures for identity of the person. Then the relational graph structures were implemented to have classification for palm print-based person identity. The experimental results on the image dataset from 100 users demonstrate the usefulness this approach for the wide range of personal identification.
- Kong W. K. [Kon03], he considered the palm print as a piece of texture and apply texture-based feature extraction techniques to palm print authentication. A 2-D Gabor filter was used to obtain texture information, and any two palm print images are compared in terms of their hamming distance. The mentioned Experimental results illustrate the effectiveness of the method.
- Zhang, et al (2003), [Zha03], in their paper they present a new biometric approach to online personal identification using palmprint technology. In contrast to the existing methods, their online palm

print identification system employs low-resolution palmprint images to achieve effective personal identification. The system consists of two parts: a novel device for online palmprint image acquisition and an efficient algorithm for fast palmprint recognition. A robust image coordinate system is defined to facilitate image alignment for feature extraction. In addition, a 2D Gabor phase encoding scheme is proposed for palmprint feature extraction and representation. The experimental results demonstrate the feasibility of the proposed system.

- Zhang, et al (2004), [Zha04], they had presented a palm print identification scheme that characterizes a palm print using a set of statistical signatures. The palm print is first transformed into the wavelet coefficients of its principle line and wrinkles. A set of statistical signatures (which includes gravity centre, density, spatial dispersivity and energy) is then defined to characterize the palm print with selected directional context values. A classification and identification scheme based on these signatures is subsequently developed. The listed test results indicate that this scheme exploits the features of the principle lines and prominent wrinkles sufficiently and achieves satisfactory results.
- Mohammed, A. M., [Moh05], he presented a hand identification using geometrical and non geometrical features to identify the unknown person. This system has two steps, first step is called preprocessing step and second step is called identification step. In the identification step fuzzy with different membership function and neural method (fuzzy self organization map) had been used. The test results shown that the trapezoidal membership function shows better performance in comparison with others.

1.7 Thesis Layout

Chapter two: “Theoretical Overview”, presents the basics of the image processing methods required to extract the geometrical and non geometrical features also an introduction to the wavelet transform is given.

Chapter three: “Design and Implementation”, in this chapter the design and implementation steps of the system are presented.

Chapter Four: “Experiment and Results”, in this chapter the test results are presented and discussed to evaluate the performance of the established system.

Chapter Five: “Conclusions and Suggestions”, in this chapter some of the derived conclusions are listed and a list suggestions for the future work are given.

Chapter Two

Chapter Two

Theoretical Overview

2.1 Introduction

The identification methods that have been developed in the last decade use various geometrical features such as (height, width, length of the finger and hand span, etc.). In addition to the above features, the moments of the palm have been used as additional features to identify the hand palm.

In this research both wavelet transform and moments have been used to extract the hand palm features for the purpose of identification and verification. So, in this chapter some related issues to both wavelets transform and moments are given beside some other digital image processing issues, which have been used to support the feature extraction process.

2.2 Pattern Recognition

Pattern recognition is a rigorous discipline aims to classify objects into a number of classes. The objects of interest are termed patterns (they may include: print characters, speech waveforms, textures, “states” of a system, and anything else need to be classified). The pattern recognition process is generally implemented in a manner that allows automatic recognition without human intervention [Tan95].

Recognition systems' solutions may used to verify the identity of a person against his/her transaction card, driving license, national ID card, or any other travel-related document. Hand geometry technology is well suited for these high volume verification applications, given the ease of use and low failure to enroll rates [Jai99].

Many image analysis tasks can be successfully performed by the classical pattern recognition system which is designed to classify an input image. For an $N \times N$ pixel block in which each pixel amplitude is quantized to B levels, the dimension of the pattern space $B^{N \cdot N}$ can be enormous even for relatively small blocks and coarse quantization. Therefore, it is necessary to perform a dimensional reduction by feature extraction. These features may include edge points, texture and Fourier components, etc. Any pattern recognition classifier is designed by measuring the representative features for the tested images. Usually the discriminating features tend to cluster into group according to the desired classes, if this does occur, then decision boundaries should be established to separate features of unknown data into proper classes [Pra78].

The typical pattern recognition process is shown in figure (2.1), the first module in recognition process is called preprocessing module which includes a number of preprocessing operations aim to transform the image into another form. Then after preprocessing stage, several image features are obtained, and plotted in the pattern space, by using feature extraction module, then as a final stage the extracted features are compared with those stored in a dedicated database to identify the pattern [Gon74].

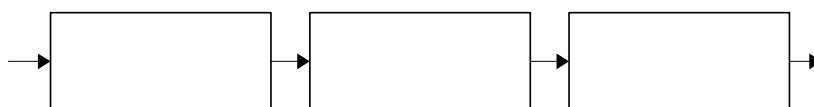


Figure (2.1) pattern recognition process

2.3 Type of Images

An image consists of a two-dimensional array of numbers, the color or gray shade displayed for a given picture element (pixel) depends on the number stored in the array for that pixel. There are many types of images

(such as binary image, gray image, color image and multi-spectral image) [Sac96].

2.3.1 Binary image

Binary image is the simplest type of the image whose pixels have only two possible intensity values. They are normally displayed as black and white. Numerically, the two values are often 0 for black, and either 1 or 255 for white. Binary images are often produced by thresholding a grayscale or color image. This type of images is frequently used in computer vision applications where the only information required for the task is the general shape, or outline information [Phi94].

2.3.2 Gray Image

A gray image is made up of pixels each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays (0...255). The eye can barely distinguish about 200 different gray levels [Sac96].

2.3.3 Color Image

Color image can be modeled as three band monochrome image data, where each band of data corresponding to different color component. The actual information stored in the digital image data is the brightness information in each spectral band. Typical color images are represented as red, green, blue or RGB. Using the 8-bit monochrome standard as a model, the corresponding color image would have 24 bits/pixel, 8 bits for each of the three color bands (red, green, and blue) [Phi94].

2.3.4 Multi-Spectral Image

A multi-spectral image is a collection of several monochrome images of the same scene, each of them taken with different sensor. Each image is referred to as a band. A well known multi-spectral (multi-band image or with different time or both) is a RGB color image, consisting of red, green and blue images, each of them taken by a sensor sensitive to a different wavelength. In image processing, multi-spectral images are most commonly used for remote sensing applications. Satellites usually take several images from different frequency bands, extend with the visual and non-visual range. For example, LANDSAT produces 7 band images with the wavelength of the bands extend from 450 to 1250 nm [Sac96].

2.4 Hand Palm Recognition / Verification System

In the following subsections some of the involved processes in the established hand-palm recognition / identification system are discussed, the way of their implementation is given in chapter 3.

2.4.1 Image Capture

The palm is the inner surface of the hand between the wrist and the fingers. The finger feature, the hand-geometry and palm features are extracted from the image of the right hand, which was placed on the flat, glass surface of a scanner [Rib02].

The user is asked to put his/her hand on the scanner using sufficient scanning quality (e.g. 150 dpi or little higher). The fingers must be clearly separated from each other in the image in order to obtain a complete hand shape [Ale02].

Figure (2.2) shows a typical image obtained from the scanner.



Figure (2.2) Typical Image

2.4.2 The Preprocessing Stage

After the hand's image is captured, the preprocessing stage should be required. The preprocessing stage consist of many steps, it discussed in the following sub section:

A. Thresholding

Thresholding is the starting step in most of the document image analysis systems; it refers to the conversion of the gray-scale image to a binary image. Thresholding is a key step in document (foreground-background) image processing modules, since a good thresholding sets the base for successful segmentation and recognition of objects [Gat04].

The simplest way of thresholding is done by replacing the intensity profile by a step function, in this method any pixel in the input image with a grey level below a predefined threshold (T) is mapped to 0 in the output image. Other pixels are mapped to 1(or 255) [Gon87].

$$G(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (2.1)$$

Figure (2.2) shows the input and output images of the thresholding process.

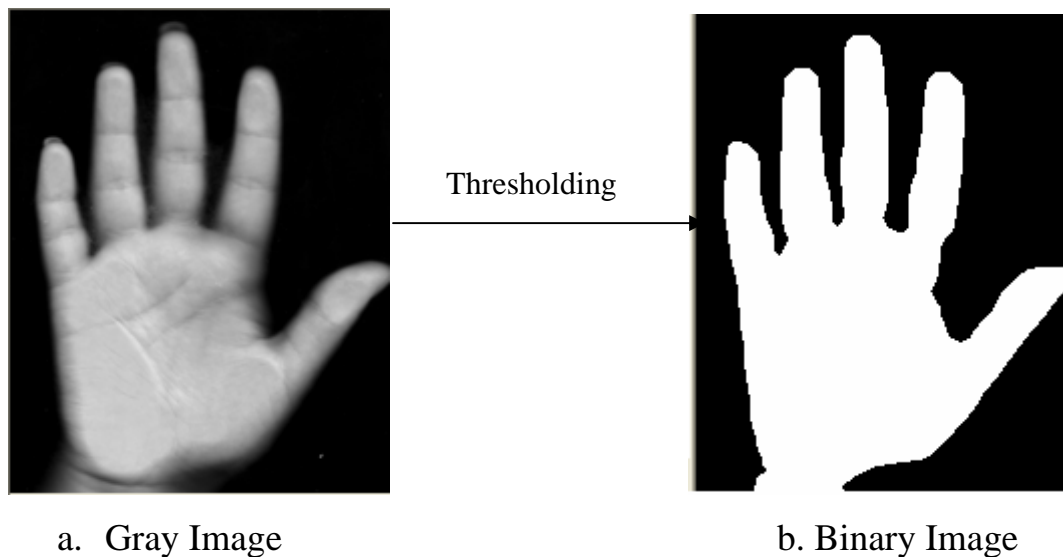


Figure (2.2) The Result of Thresholding Process

B. Enhancement

Image enhancement techniques are used as pre-processing tools in most of the image processing systems. One of these techniques is smoothing, the aim of smoothing is to suppress noise or spurious pixel values, or estimate the missing pixels values. The simplest way of noise reduction is based on analyzing the value of a few nearest neighbor pixels surrounding each pixel in the image to see if the difference between the gray value of the pixel and the average of gray values of its nearest neighbors is essential. Often, smoothing by averaging is applied; here each pixel is replaced by the average of its neighbor pixels. Another effective way of noise reduction is median filtering, where each pixel is replaced by the median of its neighbor pixels [Gon74].

C. Edge Detection

Edge detection task is based on one of the discrete differentiation forms, it is the foundation for many applications in computer vision. Edge detection is an important task in image processing. It is a main tool in

pattern recognition, image segmentation, and scene analysis. An edge is loosely defined as an extended region in the image that undergoes a rapid directional change in intensity [Gon87].

There are many types of filters could be used as edge detector (such as, Sobal, Prewit, Gradient, and Laplacian filter). Laplacian filter is used to compute the second derivatives of an image, which measures the rate at which the first derivatives change. The Laplacian is an edge detection operator; it is an approximation to the mathematical Laplacian $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial^2 y^2}$.

The set of Laplace (3x3) masks is shown in figure (2.3) [Kar00].

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure (2.3) Laplacian Masks

D. Chain Coding

The purpose of chain-coding is defining the trace of the boundary of each segment. When regions are detected and analyzed, the direction code of each pixel relative to its previously scanned neighbor is stored. In chain-coding the extra direction codes are removed and the index of the next pixel is left. The preliminary step in chain coding process is to find out a starting (or terminal) point. Once the starting points are determined then the chain-coding stage begins. Once all the pixels in the edge segments have the index of their next neighbor, then these pixels are chain coded and became ready to be traced. The indexing of chain coding direction is shown in figure (2.3) [Val99].

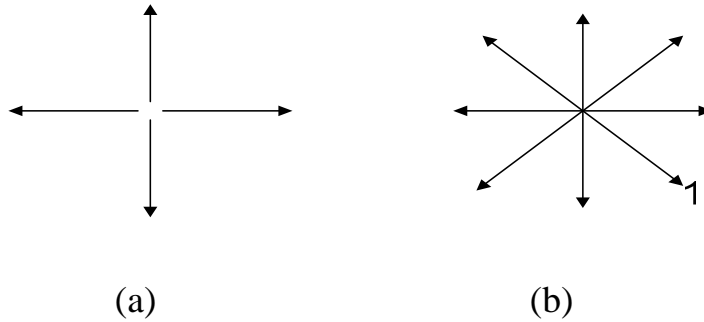


Figure (2.3) Directions indexing for (a. 4- direction chain coding;
b. 8- direction chain coding)

O

It's important to note that the chain code of a given boundary depends on the starting point, it is possible to normalize the code by a straightforward procedure, which treats it as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of the numbers must form an integer of minimum magnitude.

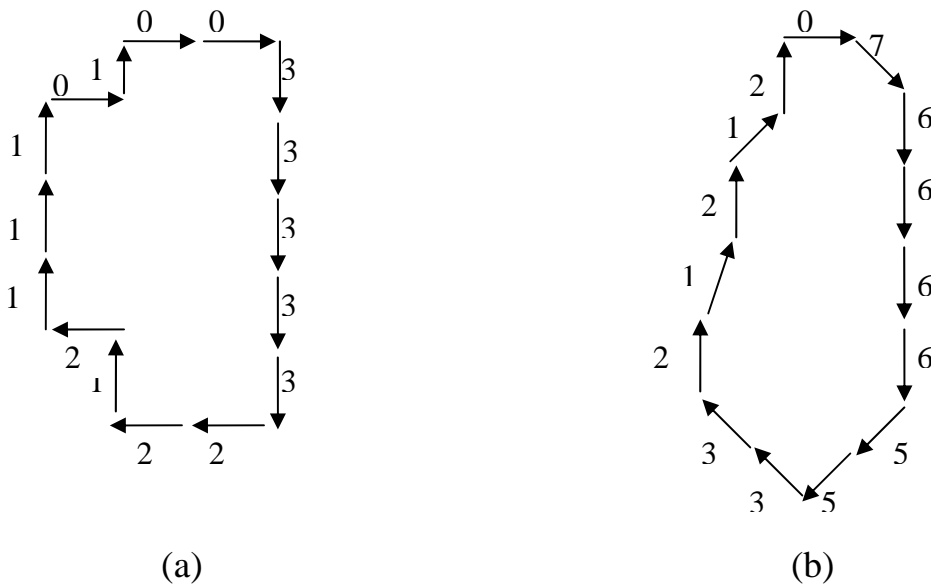


Figure (2.4) Chain Code (a. 4- directional chain code;
b. 8- directional chain code)

For example, the boundary shown in figure (2.4a) has the chain code (0033333221211101), while that shown in figure (2.4b) has the chain code (076666553321212) [Gon87].

E. Edge Linking Algorithm

The outputs of edge detectors are sometime broken edges, and in order to improve the quality of the detected edges, the use of edge linking method is required. The simplest way of connecting two edge segments is by drawing a line connects the nearest terminals of the two edge segments. The most common method used to establish line between two points is Bresenham algorithm. Bresenham line algorithm is a classic computer graphics method whose basic justification is its lack of reliance on floating-point operations. The Bresenham algorithm can be thought of as interpolating pixels, and sometimes it can be an excellent basis for other interpolation operations. It takes as an input the coordinates of the endpoints of a line segment (i.e. closest terminal points of the edge segments) and identifying the inter pixels to be set to represent the line segment itself. The key to the algorithm is that it steps one unit at a time through the more quickly varying coordinate, and each time it does so [Cun90].

2.4.3 Feature Extraction

The goal of feature extraction is to be able to describe the objects to an extent such that the classification becomes possible. A good feature will yields the same (or nearly same) value for all instances of one class and something different for instances belong to other classes. The value of the feature should not change if the object is displaced or rotated. The features should be invariant toward different appearances of an object, which is caused by perspective transformation. The purpose of a feature is to create an abstract symbolic representation, and to make it possible to discriminate between classes [Due04].

Feature extraction is always an important, and still it is difficult step in image verification. Many feature detection methods have been proposed.

Features should be easily computed, robust, insensitive to various distortions and variations in the images, and should be rotationally invariant. Two kinds of features have been used in pattern recognition problems. The first kind of features has clear physical meaning, such as geometric or structural and statistical features. The second kind of features has no physical meaning such as moments, these features are called mapping features. The advantage of physical features is that they don't deal with irrelevant features. The advantage of the mapping features is that they make classification easier because clear boundaries could be obtained between classes, but they require high computational complexity [Zhe05].

Palm prints are constituted from ridges and valleys, as like in the fingerprints. In the recognition stage, first, the features of the palm lines should be extracted, and then the classification of the palm lines could be accomplished. The classification should be supplied with fast matching processes, and require less time consuming for obtaining the results [Eki03].

2.4.4 Feature Analysis and Matching

In matching stage, a template vector is compared with the extracted feature vector, in order to determine the degree of similarity between the features vectors with the template vector, this similarity could be determined by calculating the correlation between two features vectors one of them is the template vector and the second is the extracted feature vector [Due04].

During verification, a template with the claimed identity is presented to the system. The system validates a person's identity by comparing the captured biometric characteristic with the individual's biometric template, which stored in the database. In identification mode, the system searches for a match in the whole database for the template presented to it and returns the identity of the person, or it fails if the person is not enrolled in the system database [Roy98].

No match is ever perfect in either biometric verification or an identification system, because every time a biometric is captured, the template is likely to be unique. Therefore, biometric systems can be configured to make a match or no-match decision, based on a predefined number, referred to as a threshold that establishes the acceptable degree of similarity between the trial template and the enrolled reference template [Rho04].

2.5 Wavelet Transform

Wavelets are functions defined over a finite interval. The basic idea of the wavelet transform is to represent an arbitrary function $f(x)$ as a linear combination of a set of wavelets or basis functions. The purpose of wavelet transform is to change the data from time-space domain to time-frequency domain which is useful in digital signal processing and image compression. In signal processing wavelet is helpful to recover weak signals from noise. But in image compression wavelet have been used to compress images to an extent greater than it is generally possible with the other methods [Xia01].

The Discrete Wavelet Transform (DWT) is a special case of the WT, it provides a compact representation of a signal in time and frequency [Tza01]. The standard wavelet transform algorithm is the discrete wavelet transform (DWT), which uses the set of dyadic scales and translates of the mother wavelet to form an orthogonal basis for signal analysis [Hug04].

The simplest possible orthogonal wavelet system is generated from the Haar scaling function and wavelet, this transform have two important features: linear phase and finite support. There is only one two-band orthonormal sub-band transform has these two properties, it is called Haar wavelet. The corresponding filters to the Haar transform have two-taps, so this wavelet has short length [Roj06].

The wavelet transform can be viewed as a special case of sub-band decomposition. It was extensively studied in image and video coding. Wavelets transform provide an efficient means for approximating source signal with a small number of basis elements. Wavelet bases have a number of useful properties: [Bur98]

1. They can represent smooth function.
2. They can represent singularities.
3. The basic functions are local, this makes the transform coefficients naturally adaptive to inhomogeneities in the signal.
4. They are computationally inexpensive, perhaps it is one of the few really useful linear transforms with a complexity $O(N)$.

Wavelet transform is suitable for the analysis of natural images because it provides a good representation of the image data in terms of both spatial and frequency localization. Natural images are locally smooth, and their frequency characteristics vary in space. The Fourier transform and DCT are capable to capture the global frequency characteristics of a signal, but they do not take into account the fact that different regions in an image may have different frequency characteristics. The scaling property of the wavelet transform provides information about frequency characteristics and where these characteristics are localized in space [Wan05].

2.6 Haar Wavelet Transform

Haar wavelet is probably the simplest type of all wavelets, and it is the oldest. It has been used in image analysis for many years as the Haar transform. It's illustrating that the power of wavelet transform. The disadvantage of the Haar wavelet is that it is not continuous and therefore not differentiable. Haar wavelet is a step function taking the values +1 and -1 as below:

$$\Psi(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ -1 & 0.5 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

Haar Wavelet Transform (HWT) consists of both: low pass and high pass filters. It is preferred because it can be readily implemented in hardware. The one level classical wavelet transform decomposes the original image into four sub images:

- (1) The approximation band (LL): it is the result of applying low pass filter in vertical and horizontal directions.
- (2) The (LH) band: it is the result of applying horizontal low pass filter and vertical high pass filter.
- (3) The (HL) band: it is the result of horizontal high pass filter and vertical low pass filter.
- (4) The (HH) band: it is the result of horizontal and vertical high pass filter.

In Haar wavelet transform each (2x2) adjacent pixels are picked as group and passed simultaneously through four filters (i.e., LL, HL, LH, and HH), this operation can be repeated for obtaining a multi-level wavelet transform. Figure (2.5) illustrates the decomposition stages for a three level wavelet transform [Geo05].

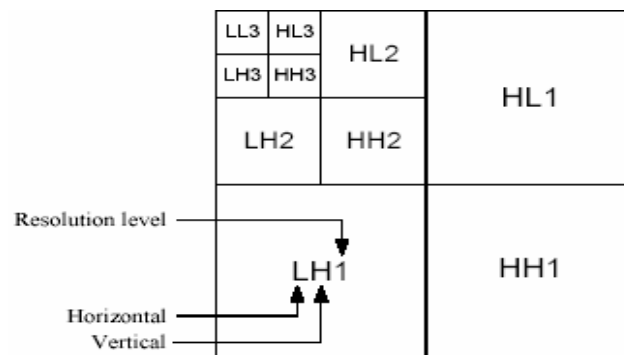


Figure (2.5) Three level wavelet transform

To obtain the four wavelet coefficients (LL, HL, LH, HH), the bases of these 4-filters could be written as follows:

$$1. \text{ LL Pass Filter: } LL(x, y) = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$2. \text{ HL Pass Filter: } HL(x, y) = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$3. \text{ LH Pass Filter: } LH(x, y) = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$4. \text{ HH Pass Filter: } HH(x, y) = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Therefore, the image could be decomposed by applying the following Haar wavelet equations:

$$LL(x, y) = \frac{f(2x, 2y) + f(2x+1, 2y) + f(2x, 2y+1) + f(2x+1, 2y+1)}{2} \quad (2.3)$$

$$LH(x, y) = \frac{f(2x, 2y) + f(2x+1, 2y) - f(2x, 2y+1) - f(2x+1, 2y+1)}{2} \quad (2.4)$$

$$HL(x, y) = \frac{f(2x, 2y) - f(2x+1, 2y) + f(2x, 2y+1) - f(2x+1, 2y+1)}{2} \quad (2.5)$$

$$HH(x, y) = \frac{f(2x, 2y) - f(2x+1, 2y) - f(2x, 2y+1) + f(2x+1, 2y+1)}{2} \quad (2.6)$$

2.7 Moments

Global geometric moments and their invariants are widely used in many areas of image analysis, including pattern recognition, image reconstruction, and shape identification. In addition to geometric moments, which are also known as regular or ordinary moments, a number of other moments have been proposed in the literature. Moment characteristics can be used for shape description even if the region is represented by its boundary. [Mic04]

For 2D continuous function $f(x, y)$, where x and y are the spatial coordinate, and the moment of order $(p+q)$ “ m_{pq} ” is defined as [Tan98]:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q=0,1,2, \quad (2.7)$$

For digital images the integrals are replaced by summations and the equation of M_{pq} becomes:

$$m_{pq} = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} x^p y^q f(x, y) \quad (2.8)$$

The central moments of a digital image can be expressed as shown below:

$$M_p = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} (x + iy)^p f(x, y) \quad (2.9)$$

Where

P is the moment order

$$x' = \frac{x - x_c}{x_c}, \quad y' = \frac{y - y_c}{y_c} \quad (2.10)$$

$$x_c = \frac{W - 1}{2}, \quad y_c = \frac{H - 1}{2} \quad (2.11)$$

Where

W is the width of the image

H is the height of the image

There is another type of moments called moment invariant, the concept of moments as invariant image features was first introduced by Hu (1962), and later revised by Reiss (1991). Moment invariants are properties of connected regions in binary images that are invariant to translation, rotation and scaling. They are useful because they define a simply calculated set of region properties that can be used for shape classification and recognition. Moments and functions of moments have been used as pattern features in many applications. In this research, the seven invariant moments

described by Hu (1962), and Gonzalez (1987) have been used. Hu (1962) had derived seven invariant moments, ($i = 1, 2, \dots, 7$), they are nonlinear combinations of the second and third normalized central moments. These invariant moments possess the desirable properties of being translation, rotation, and scale invariant. Hu published the following seven rotation invariants, they consist of the second and third order moments [Tan98]:

$$\Phi_1 = m_{20} + m_{02} \quad (2.12)$$

$$\Phi_2 = (m_{20} - m_{02})^2 + 4m_{11}^2 \quad (2.13)$$

$$\Phi_3 = (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \quad (2.14)$$

$$\Phi_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2 \quad (2.15)$$

$$\begin{aligned} \Phi_5 = & (m_{30} - 3m_{12})(m_{30} + m_{12})((m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2) + \\ & (3m_{21} - m_{03})(m_{21} + m_{03})(3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2) \end{aligned} \quad (2.16)$$

$$\Phi_6 = (m_{20} - m_{02})((m_{30} + m_{12})^2 - (m_{21} + m_{03})^2) + 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03}) \quad (2.17)$$

$$\begin{aligned} \Phi_7 = & (3m_{21} - m_{03})(m_{30} + m_{12})((m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2) - \\ & (m_{30} - 3m_{12})(m_{21} + m_{03})(3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2) \end{aligned} \quad (2.18)$$

Chapter Three

Chapter Three

Design and Implementation

3.1 Introduction

This chapter presents the implementation of the hand palm identification system. It describes the implemented steps in both enrollment and recognition/verification modules. Also, the method of extracting the moments features from the wavelet sub-bands is illustrated.

3.2 The Overall System Stages

Figure (3.1) shows the main system stages, it consists of two main stages:

1. Enrollment stage: in this module the input is the color hand image, the preprocessing operations are performed which include:
 - A. image binarization to convert the color image to binary image (black and white).
 - B. edge detection, where Laplace edge detector has been used.
 - C. chain coding.

The next stage in this module is the feature extraction. In this proposed identification system the non geometrical features set, which consist of the invariant and central moments of the sub bands of Haar wavelet transform are used as hand palm discrimination features. These sets of extracted features are stored in the database system. Figure (3.2) shows the implemented steps of the enrollment module.

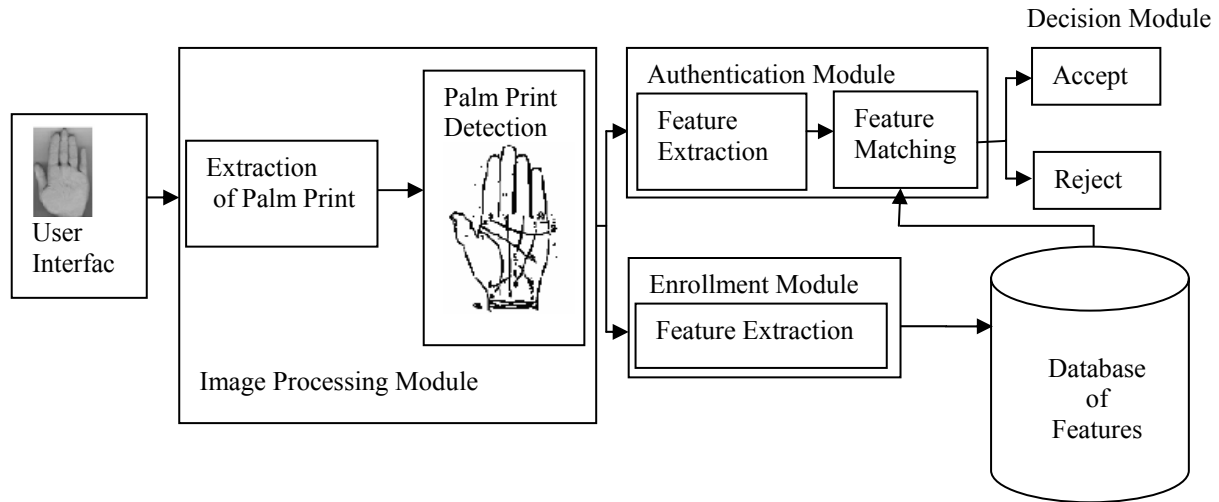


Figure (3.1) The layout of the established hand palm identification system

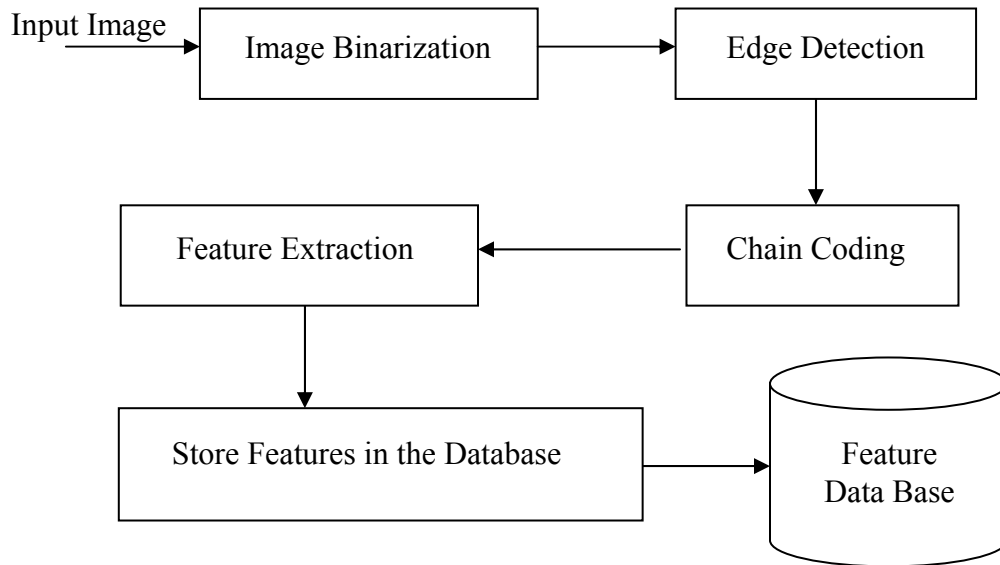


Figure (3.2) Stage of the Enrollment Module

2. Authentication stage: in this stage the input is the color image, and the same preprocessing operations and features extraction steps that performed in enrollment module are implemented here. After the computation of the moment features vector of the palm, finally it is compared with the

template features vectors stored in the database system to find out the best matched features vector registered in the database, and consequently to assign the ID of tested palm.

3.3 Preprocessing Stage

The processes included in this stage are required to make the data of the hand image more suitable to extract the discriminating features which are required in the identification and verification process. The first step in this stage, after capturing the hand image, is the conversion of color image to binary image. Then in the second step the image of hand boundary edge is extracted from the binary image, and finally the chain code of the detected edge points is determined.

3.3.1 Image Binarization

The first step in the image binarization process is the conversion of color image to gray image (where its gray scale is 256 levels). Then the second step is converting the gray image to a binary image which consists of two colors (black and white). The implementation of these two steps is illustrated in the following:

1. Conversion of the color image to gray image: each pixel in the image consists of three principal colors (red, green, and blue). Each color component is represented by one byte. The gray value is obtained by using the following equation:

$$GrayValue = \frac{Red + Green + Blue}{3} \quad (3.1)$$

The above equation is applied on all image pixels in order to convert the color image to gray image.

2. Conversion of the gray image to binary image: the first step in this process is the computation of the histogram of the gray image. Histogram shows the number of pixels for each gray value within the range of the image gray scale. If the minimum value of the image is 0 and the maximum value of the image is 255, then the histogram of the image shows the number of pixels for each gray value ranging between 0 and 255. Peaks in the histogram represent more common gray values within the image that usually consists of nearly uniform regions. Valleys in the histogram represent less common values. Empty regions within the histogram indicate that no pixels within the image contain gray values belong to these empty regions. The conversion from gray image to binary image could be done by using thresholding method. In threshold method, a threshold value should be defined. The simplest way to determine the threshold value is dividing the histogram into two regions (i.e., low gray and high gray regions). For each region find the maximum peak value of the histogram, then the average of the two gray values correspond to these two peaks is considered as the threshold. Figure (3.3) shows the curve of the hand gray image. After the computation of threshold, the gray image is converted to binary image by testing each image pixel in the image alone; if it is greater than the threshold then the tested pixel is converted to white pixel (255) otherwise, it is converted to black (0).

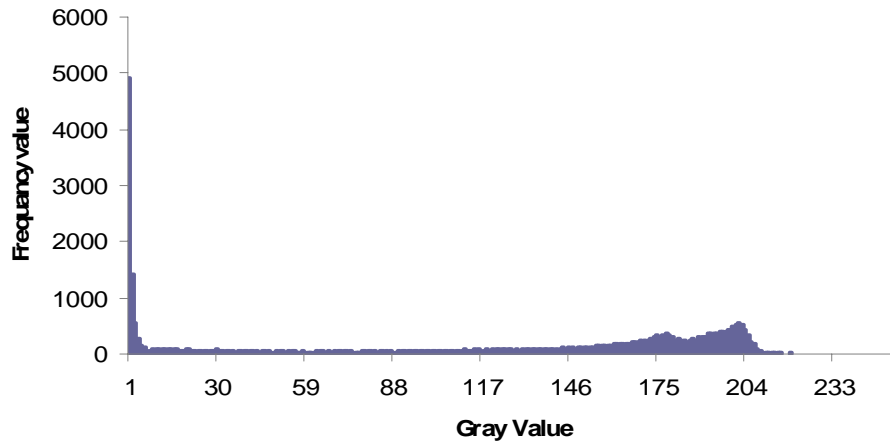


Figure (3.3) The curve of a hand image

Algorithm (3.1) illustrates the implemented steps to convert the hand gray images to binary.

Algorithm (3.1) conversion the color image to the binary image.

Input: img // is the array of the color image

width // is the width of the image

height // is the height of the image

Output: binary // is the array of black and white image

Procedure:

Loop for y=0 to height-1

Loop for x=0 to width-1

Set grayvalue \leftarrow (img(x,y).r +img (x,y).g + img(x,y).b)/3

Set gray (x,y) \leftarrow grayvalue

Set histogram (grayvalue) \leftarrow histogram (grayvalue) +1

End Loop

End Loop

Set mid \leftarrow 255 div 2

Set v1 \leftarrow 0

Set max \leftarrow histogram (v1)

Loop for each grayvalue \leq mid

If histogram (grayvalue) > max then

Set max \leftarrow histogram (grayvalue)

Set v1 \leftarrow grayvalue

```
End if
End Loop
Set v2=255
Set max ← histogram (v2)
Loop for each grayvalue ≤ 255
    If histogram (grayvalue) > max then
        Set max ← histogram (grayvalue)
        Set v2 ← grayvalue
    End if
End Loop
Set threshold ← (v1+v2) / 2
Loop for each pixel in gray image
    If gray (x,y) ≥ threshold then
        Set binary (x,y) ← 255
    Else
        Set binary (x,y) ← 0
    End if
End Loop
```

3.3.2 Smoothing and Erosion

One of the simplest methods could be used to remove noise (white points in black regions) or gaps (black points in white regions) is smoothing. The method used in this research is by checking the surrounding neighbors of each pixels in the image; for example if the pixel is black and most of its neighbors are white then convert the pixel to white, while if the pixel is white and most of the neighbors are black then convert it to black.

Algorithm (3.2) illustrate the steps of smoothing process

```
Input: bin      // is the array of binary image
         width    // is the width of the image
         height   // is the height of the image
Output: binary // is the array without any noise
Procedure:
  Loop for i=1 to 4
    Loop for y=0 to height -1
      Loop for x=0 to width -1
        If bin(x,y) = 0 then
          Set sum ← 0
          Loop for xx=x-1 to x+1
            Loop for yy =y-1 to y+1
              If xx <> x and yy <> y and bin(xx, yy) <> 0 then
                Set sum←sum+1
            End loop
          End loop
          If sum > 4 then
            Set bin (x,y) ← 255
          Else If sum < 4 then
            Set bin (x,y) ← 0
          Else
            Set bin(x,y) ← binary (x,y)
          End if
        End Loop
      End Loop
    End Loop
  For all pixels in bin image
    Set binary (x,y) ← bin (x,y)
  End loop
End Loop
```

3.3.3 Edge Detection

Edge detection was used to find the boundary of the hand image. In this research Laplace filter, shown in Figure (3.4) have been used to obtain the boundary of the binary image. Laplace filter is passed over all pixels in the

binary image, the filter output is determined according to the following equations:

$$P(x,y)=4*\text{Bin}(x,y)-\text{Bin}(x,y-1)-\text{Bin}(x-1,y)-\text{Bin}(x+1,y)-\text{Bin}(x,y+1) \quad (3.2)$$

Where,

$\text{Bin}(x,y)$ is the binary image.

$P(x,y)$ is the output of Laplace filter.

If the value of the output is greater than 0 then the result is considered as an edge pixel, otherwise it is considered as inner or outer pixel.

0	-1	0
-1	4	-1
0	-1	0

Figure (3.4) Laplacian Mask

Algorithm (3.3) illustrates the implemented steps to perform the Laplacian filtering, the input is the binary image and the output is the boundary edge image, it is represented by a two dimensional array contain the binary values of all pixels of the boundary image.

Algorithm (3.3) illustrate Laplace edge detection

Input: bin, width, height

Output: boundaryimage

Procedure:

Loop for y=0 to height-1

Loop for x=0 to width-1

Set $v \leftarrow -\text{bin}(x, y - 1) - (\text{bin}(x - 1, y) - \text{bin}(x, y + 1) - \text{bin}(x + 1, y) + \text{bin}(x, y) * 4$

If $v > 0$ then Set boundaryimage(x,y) $\leftarrow 255$

Else Set boundaryimage(x,y) $\leftarrow 0$

End if

End loop

End loop

3.3.4 Chain Coding for Boundary Tracing

Chain code is usually used to represent the boundary of a closed object in an image as a sequence of line segments of specific length and direction. The boundary pixels which have been produced by applying Laplace edge detector on the binary image need to be represented as a sequence of points in order to be used to extract the hand palm area. The purpose of chain coding is to represent the boundary edge as a one dimensional array of points type records, each point record include the x and y coordinates. The chain coding process is initialized by finding the start point of the hand edge. After registering this start point into an empty boundary array, then check the 8-neighbors of the start point and list the first detected neighbor point into the boundary array. As a next step the 8-neighbors of the second point listed in the boundary array should be checked and its first detected edge point should be added to the array, continue the steps of checking the neighbor of the listed points till reaching the end of the list and no new edge point is added to the list.

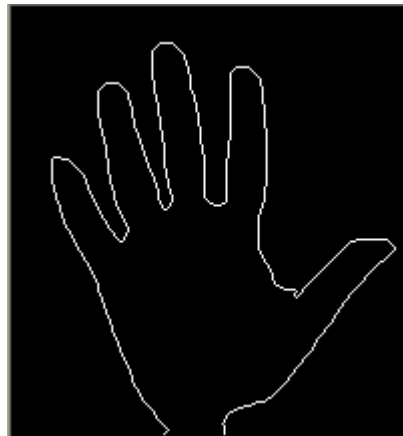


Figure (3.5) an image shows the boundary of a hand

After finding the start edge point and begin searching its neighbors to find if any of them could be considered as a new edge point, if non of them can be considered as a new point in this case the start point should be deleted from boundary array and continue the search process till finding another start point with connected neighbors. The neighbor search direction could be clockwise or counter clockwise. Algorithm (3.4) illustrates the implemented steps to trace the boundary pixel.

Algorithm (3.4) Chain Coding Method

Input: boundaryimage // is the array of the boundary image
width // is the width of the image
height // is the height of the image
Output: bufferx // is the array which contain the sequence of
the x- axis of the boundary of the hand
buffery // is the array which contain the sequence of
the y- axis of the boundary of the hand

Procedure:

dx(0) = 1: dy(0) = 0: dx(1) = 1: dy(1) = 1
dx(2) = 0: dy(2) = 1: dx(3) = -1: dy(3) = 1
dx(4) = -1: dy(4) = 0: dx(5) = -1: dy(5) = -1
dx(6) = 0: dy(6) = -1: dx(7) = 1: dy(7) = -1

h=height

w=width

Loop for y = height to 0

Loop for x = 0 to width

If boundaryimage (x,y) \neq 0 then

Set i \leftarrow 0: Set l \leftarrow 0

Set bufferx (l) \leftarrow x: Set buffery(l) \leftarrow y

Set boundaryimage(x, y) \leftarrow 0

do

flag \leftarrow false

Loop for kk \leftarrow 0 to 7

Select Case kk

Case 0: k \leftarrow i

Case 1: k \leftarrow i + 1: If k = 8 Then Set k \leftarrow 0


```
Case 2: k ← i - 1: If k = -1 Then Set k ← 7
Case 3: k ← i - 2: If k < 0 Then Set k ← k + 8
Case 4: k ← i + 2: If k > 7 Then Set k ← k - 8
Case 5: k ← i + 3: If k > 7 Then Set k ← k - 8
Case 6: k ← i - 3: If k < 0 Then Set k ← k + 8
Case 7: k ← i - 4: If k < 0 Then Set k ← k + 8
End Select
Set xx ← bufferx (l) + dx(k) : Set yy ← buffery(l) +
dy(k)
If xx >= 0 and xx < w and yy >= 0 and yy < h Then
    If boundaryimage (xx, yy) <> 0 Then
        Set l ← l + 1
        Set bufferx (l) ← xx
        Set buffery (l) ← yy
        Set boundary_array (xx, yy) ← 0
        Set i ← k
        Set kk ← 7
        Set flag ← true
    End if
End if
End Loop
End Loop
End if
End Loop
End Loop
```

3.4 Extraction of Palm Area

In order to calculate the area of the hand palm, the regions of the fingers should be excluded from the hand image. The following steps have been implemented to extract the fingers region in the image of boundary edge of the hand from the hand palm area.

3.4.1 Joint Points Detection

The joint points B, C, D and F shown in figure (3.6) which are the lowest points between the fingers edge segments have been computed by

testing all the points of the hand, and allocate the points whose y coordinate are greater than or equal to the y coordinates of the next and previous points. Algorithm (3.5) illustrates the implemented steps to find the joint points of the hand.

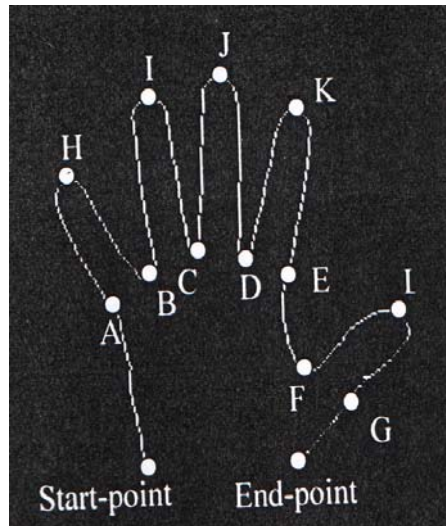


Figure (3.6) Top and Joint Points

Algorithm (3.5) Joint Points B, C, D and F Detection

Input: bufferx // is array which contain the sequence of the x- axis
of the boundary of the hand
buffery // is the array which contain the sequence of the y-
axis of the boundary of the hand

no_of_joint // is the number of points in the boundary array
Output: Joint // is the array of record contain the joint points (B, C,
D and F)

Procedure:

Set counter ← 0

Set joint(counter).x ← bufferx(0)

Set joint(counter).y ← buffery(0)

Set counter ← counter + 1

Loop for i = 20 to no_of_joint - 20

If (buffery(i) > buffery(i - 10)) And (buffery(i) > buffery(i + 10))

Then

```
        Set top_list(f).x ← bufferx(i)
        Set top_list(f).y ← buffery(i)
        Set f ← f + 1
    End If
End Loop
Set i ← 0
Set ss ← 0
Set rr ← 0
While i < f - 1
    Set ii ← 0
    While flag=true
        Set dx ← top_list(i + 1).x - top_list(i).x
        Set dy ← top_list(i + 1).y - top_list(i).y
        Set d ← ((dx *dx) + (dy *dy) / 2
        If (d = 1) or (d = 0.5) Then
            Set choice (ss).xx (ii) ← top_list (i).x
            Set choice (ss).yy (ii) ← top_list (i).y
            Set ii ← ii + 1
        Else
            Set choice (ss).xx (ii) ← top_list(i).x
            Set choice (ss).yy (ii) ← top_list(i).y
            Set ii ← ii + 1
            Set flag ← false
            Set choice (ss).m ← ii
            Set ss ← ss + 1
        End If
        Set i ← i + 1
    Wend
    Set mid ← choice (ss - 1).m / 2
    Set joint (counter).x ← choice (ss - 1).xx (mid)
    Set joint (counter).y ← choice (ss - 1).yy (mid)
    Set counter ← counter + 1
Wend
Set joint(counter).x ← bufferx(1 - 1)
Set joint(counter).y ← buffery(1 - 1)
Set counter ← counter + 1
```

The implemented steps for finding the top points H, K and L are shown in algorithm (3.6). These points should be calculated before start to compute

the points A, E and G. This can be done by testing all the points of the hand and allocate the points whose y coordinate is less than or equal to the y coordinate of the next and previous points.

Algorithm (3.6) Top Points H, K, I, J and L Detection

Input: bufferx, buffery

no_of_point // is the number of points in the boundary array

Output: top2 // is the array of top points (H, K, I, J and L)

Procedure:

Set f ← 0

Loop for i = 20 to no_of_point - 20

If (buffery(i) < buffery(i - 10)) And (buffery(i) < buffery(i + 10))

Then

Set top (f).x ← bufferx(i)

Set top (f).y ← buffery(i)

Set f ← f + 1

End If

End Loop

Set i ← 0

Loop while i < f-1

Set flag ← true

Set ii ← 0

While flag=true

Set dx ← top (i + 1).x - top (i).x

Set dy ← top (i + 1).y - top (i).y

Set d ← ((dx *dy) + (dy*dy)) / 2

If (d = 1) or (d = 0.5) Then

Set choice(ss).xx(ii) ← top (i).x

Set choice(ss).yy(ii) ← top (i).y

Set ii = ii + 1

Else

Set choice(ss).xx(ii) ← top (i).x

Set choice(ss).yy(ii) ← top (i).y

Set ii ← ii + 1: Set flag ← false

Set choice(ss).m ← ii : Set ss ← ss + 1

End If

Set i ← i + 1

Wend

```
Set mid ← choice(ss - 1).m / 2
  Set top2(counter).x ← choice(ss - 1).xx(mid)
  Set top2(counter).y ← choice(ss - 1).yy(mid)
  Set counter ← counter + 1
Wend
```

The joint points A, E and G are considered as the boundary points that have the smallest distance from the points B, D and F respectively. These points A, E and G must be included in the chain algorithm and lays between point H and starting point for the point A, and between points K and F for the point E, and finally between L and end point for the point G.

Algorithm (3.7) demonstrates the implemented steps for determining the joint points A, E and G of the hand palm.

Algorithm (3.7) joint points A, E and G Detection

Input: bufferx, buffery, joint
top2 // is the array of the top points (H, I, J, K and L)
counter // is the number of joint points
Output: joint // is the array of joint points (A, B, C, D, E, F, and G)

Procedure:

```
// 1. Compute the Point A
Set max ← ((bufferx(0) - joint(0).x) ^ 2 + (buffery(0) - joint(0).y) ^ 2) ^ 0.5
Set s ← 0
While bufferx(s) <> top2.x(0) and buffery(s) <> top2.y(0)
  "the point top2(0) is point H
  Set min_dis ← ((bufferx(s)-joint(1).x) ^2-(buffery(s)- joint(1).y) ^2) ^ 0.5
  "joint(1) is the joint point B
  If min_dis <max then
    Set joint(counter).x ← bufferx(s)
    Set joint(counter).y ← buffery(s)
    Set counter ← counter+1
    Set max ← min_dis
End if
```

```
    Set s← s+1
End loop
// 2. Compute the Point E
Set max ← ((bufferx(s) - joint(3).x) ^ 2 + (buffery(s) - joint(3).y) ^ 2) ^
0.5      "joint(3) is the point D
While bufferx(s) <> top2.x(3) and buffery(s)<> top2.y(3)
    "the point top2(3) is the point K
    Set min_dis ← ((bufferx(s)-joint(3).x)^2-(buffery(s)-joint(3).y) ^2)
    ^0.5      "the joint(3) is the joint point D
    If min_dis < max then
        Set joint(counter).x ← bufferx(s)
        Set joint(counter).y ← buffery(s)
        Set counter ← counter+1
        Set max ← min_dis
    End if
    Set s← s+1
End loop

// 3. Compute the Point G
Set max ← ((bufferx(s) - joint(4).x) ^ 2 + (buffery(s) - joint(4).y) ^ 2) ^
0.5
While bufferx(s)<> joint(5).x and buffery(s)<>joint(5).y
    "the joint(5) is the end point
    Set min_dis← ((bufferx(s)-joint(4).x)^2-(buffery(s)-joint(4).y) ^2)
    ^0.5      "joint(4) is the joint point F
    If min_dis < max then
        Set joint(counter).x ← bufferx(s)
        Set joint(counter).y ← buffery(s)
        Set counter ← counter+1
        Set max ← min_dis
    End if
    Set s← s+1
End loop
```

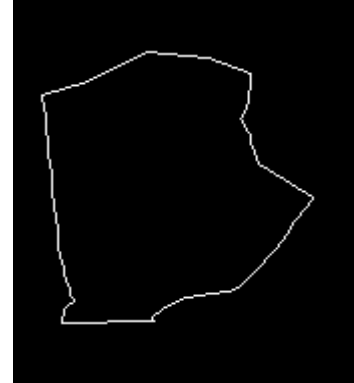
3.4.2 Bresenham Algorithm

After finding the joint points of the hand, then the hand palm region is allocated by isolating the fingers regions from the hand image. This is done by using Bresenham algorithm to connect the joint points by lines, where each two

neighbor joint points are connected by a straight line, as shown in Figure (3.7). Algorithm (3.8) illustrates the implemented steps of Bresenham algorithm.



(a)



(b)

Figure (3.7) hand-palm boundary before and after applying Bresenham algorithm

Algorithm (3.8) Bresenham Algorithm

Input: x_1, x_2, y_1, y_2 // two points
 m_palm // is the array of the boundary hand

Output: draw line between x_1 and x_2

Procedure:

```
Set  $adx \leftarrow Abs(x_2 - x_1)$ 
If  $x_2 \geq x_1$  Then  $stpx \leftarrow 1$  Else  $stpx \leftarrow -1$ 
Set  $ady \leftarrow Abs(y_2 - y_1)$ 
If  $y_2 \geq y_1$  Then  $stpy \leftarrow 1$  Else  $stpy \leftarrow -1$ 
If  $adx \geq ady$  Then
  Set  $r \leftarrow stpy * ady / adx$ 
  Set  $yy \leftarrow y_1$ 
  Loop for  $x = x_1$  to  $x_2$  Step  $stpx$ 
    Set  $m\_palm(x, cint(yy)) \leftarrow 255$ 
    Set  $yy \leftarrow yy + r$ 
  End loop
```

```
Else
  Set  $r \leftarrow stpx * adx / ady$ 
  Set  $xx \leftarrow x1$ 
  Loop for  $y = y1$  to  $y2$  Step  $stpy$ 
    Set  $m\_palm(cint(xx), y) \leftarrow 255$ 
    Set  $xx \leftarrow xx + r$ 
  End loop
End If
```

3.4.3 Seed Filling

After applying the stages of fingers separation from the hand image, and establish a set of straight lines connecting the finger's points, then the hand palm image is extracted by applying seed filling algorithm. The first step in this algorithm is the computation of the center point coordinates of the hand palm and then register this center point into a temporary array (T), and then start checking its 4-neighbors, if any of the four tested points is found black then register this point in the array (T) and convert the value of the detected black point to white (255). Then some steps are applied on the next point listed in array (T) till reaching the last point listed in (T) and no new point is added to (T). Algorithm (3.9) illustrates the implemented steps to fill the palm of the hand by using Seed Filling Algorithm.

Algorithm (3.9) Seed Filling Algorithm

Input: height, width, m_palm

Output: m_palm

Procedure:

```
Loop for each pixel in the m_palm
  If the value of pixel is white then
    Set xc ← xc + x
    Set yc ← yc + y
    Set counter ← counter+1
  End if
End Loop
Set x ← xc/counter : Set y ← yc/counter
Set hm ← height - 1 : Set wm ← width - 1
Set j ← 0: Set i ← 0
Set tempo (j).x ← x
Set tempo (j).y ← y
Set x ← x
Set y ← y
Set m_palm(x, y) ← 255
Loop for each i ≤ j
  Loop for each m ≤ 3
    Set flg ← 0
    Select Case m
      Case 0: If y > 0 Then yy ← y - 1: xx ← x: flg ← 1
      Case 1: If y < hm Then yy ← y + 1: xx ← x: flg ← 1
      Case 2: If x > 0 Then yy ← y: xx ← x - 1: flg ← 1
      Case 3: If x < wm Then yy ← y: xx ← x + 1: flg ← 1
    End Select
    If flg = 1 Then
      If m_palm (xx, yy) = 0 Then
        Set j ← j + 1
        Set tempo (j).x ← xx
        Set tempo (j).y ← yy
        Set m_palm (xx, yy) ← 255
      End If
    End If
  End Loop
  Set i ← i + 1 : Set x ← tempo (i).x
  Set y ← tempo (i).y
End Loop
```

3.4.4 Hand Span Detection

The active area of hand span was confined by the largest circle that can be drawn within the palm area of the hand. The process of determining the radius of this circular area is started by computing the mean value of the x and y-coordinates, of the palm edge points, by using the following equations:

$$X_c = \frac{1}{N} \sum_{i=0}^{N-1} X_i \quad (3.3)$$

$$Y_c = \frac{1}{N} \sum_{i=0}^{N-1} Y_i \quad (3.4)$$

And then compute the variations R_x and R_y along the x and y coordinates, respectively, by applying the following equations:

$$R_x = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (X_i - X_c)^2} \quad (3.5)$$

$$R_y = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} (Y_i - Y_c)^2} \quad (3.6)$$

Then compute the radius as the average value of R_x and R_y :

$$R = \frac{(R_x + R_y)}{2} \quad (3.7)$$

Algorithm (3.10) illustrates the implemented steps for determining the hand span.

Algorithm (3.10) Hand Span Algorithm

Input: buffer, height, width

Output: radius // is the radius of the palm

Procedure:

```
    Loop for each pixel in the array
      If buffer (i, j) = 255 Then
        Set xc ← xc + i
        Set yc ← yc + j
        Set con ← con + 1
      End If
    End Loop
    Set xc ← xc / con
    Set yc ← yc / con
    Set subx ← 0
    Set suby ← 0
    Loop for yi=0 to height-1
      Loop for xi=0 to width-1
        If buffer (xi, yi) = 255 Then
          Set subx ← subx + (xi - xc) * (xi - xc)
          Set suby ← suby + (yi - yc) * (yi - yc)
        End If
      End loop
    End Loop
    Set rx ← (subx / con) ^ 0.5
    Set ry ← (suby / con) ^ 0.5
    Set radius ← 1.1 * (rx + ry) / 2
```

3.5 Haar Wavelet Transform

It's the simplest and oldest method of wavelet transform. After separating the palm area from the hand image then Haar wavelet transform had been applied on the hand palm points exclusively. To determine the Haar wavelet coefficients for the sub bands (LL, LH, HL, HH) the equations of the sub band coefficient describe in equations (2.3 - 2.6).

Algorithm (3.11) illustrates the implemented steps to decompose the image data into four sub bands by applying the Haar wavelet transform.

Algorithm (3.11) Haar Wavelet Transform

Input: m_palm // is the tow dimensional array

Height // is the height of the image

Width // is the width of the image

no // is the number of levels

Output: reci // is the array of record of Haar wavelet transform for 2 passes

Procedure:

hi=height

wi=width

Loop for i=0 to no-1

Loop for y = 0 to (hi / 2) - 1

Loop for x = 0 to (wi / 2) - 1

Set exp = 2 * x + 1: Set conx = 2 * x

Set cyp = 2 * y + 1: Set cony = 2 * y

Set reci(I).ll(x,y) ← 0.5 * (cint(co_im1(conx, cony)) + co_im1(exp, cony) + co_im1(conx, cyp) + co_im1(exp, cyp))

Set reci(I).lh(x, y) ← 0.5 * (cint(co_im1(conx, cony)) + co_im1(exp, cony) - co_im1(conx, cyp) - co_im1(exp, cyp))

Set reci(I).hl(x,y)←0.5*(cint(co_im1(conx, cony)) - co_im1(exp, cony) +co_im1(conx, cyp) - co_im1(exp, cyp))

Set reci(I).hh(x, y)←0.5*(cint(co_im1(conx,cony)) -

co_im1(exp, cony) - co_im1(conx, cyp) + co_im1(exp, cyp))

End Loop

End Loop

Set hi ← (hi + 1) / 2

Set wi ← (wi + 1) / 2

Loop for y = 0 to hi - 1

Loop for x = 0 to wi - 1

Set co_im1(x, y) ← reci(I).ll(x, y)

End Loop

End Loop

End Loop

3.6 Moments

The next step is to compute the invariant moment and central moment of the hand palm for each wavelet sub bands of the hand palm area. The moment of order ($n=p+q$) is determined according to the following equation [Tan98]:

$$M_{pq} = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} x^p y^q f(x, y) dx dy \quad p, q=0,1,2,\dots \quad (3.12)$$

The seven rotation invariant moments have been determined. They are based on the second and third order moments as described in equations (2.12-2.18).

The invariant moment has been determined for each produced sub band of the Haar wavelet. In this research the wavelet transform had been applied on the palm area two times in dynamic architecture and, therefore, seven wavelet sub bands have been produced.

Algorithm (3.12) illustrates the implemented steps for determining the seven invariant moments.

Algorithm (3.12) Invariant Moment

Input: co_im1 // is the two dimensional array of gray image

width // is the width of the image

height // is the height of the image

no // is the number of levels

Output: moin // is the one dimensional array of invariant moment features

Procedure:

Set sumx ← 0

Set sumy ← 0

Set sum ← 0

Loop for count=0 to no-1

Loop for y = 0 to heighth-1

Loop for x = 0 to width-1

Set sumx ← sumx + x * co_im1(x, y)

Set sum ← sum + co_im1(x, y)

```

    Set sumy ← sumy + y * co_im1(x, y)
End Loop
End Loop
Set xxc ← sumx / sum
Set yyc ← sumy / sum
Loop for p = 0 to 3
    Loop for q = 0 to 3
        dif ← 0
        Loop for j = 0 to h-1
            Loop for i = 0 to w-1
                Set xcc ← (i - xxc) / w
                Set ycc ← (j - yyc) / h
                Set dif ← dif + xcc ^ p * ycc ^ q * co_im1(x,y)
            End Loop
        End Loop
    Set mom (p,q) ← dif
End Loop
num ← mom(0,0)
Loop p=0 to 3
    Loop for q = 0 to 3
        mom(p,q) ← mom(p,q) / num
    End Loop
End Loop
Set m0 ← mom(2, 0) - mom(0, 2)
Set m1 ← mom(3, 0) - 3 * mom(1, 2)
Set m2 ← 3 * Mom(2, 1) - Mom(0, 3)
Set m3 ← mom(3, 0) + Mom(1, 2)
Set m4 ← mom(2, 1) + Mom(0, 3)
Set m5 ← m3 ^ 2 - 3 * (m4 ^ 2)
Set m6 ← 3 * (m3 ^ 2) - m4 ^ 2
Set moin (1) ← mom(2, 0) + mom(0, 2)
Set moin (2) ← (m0) ^ 2 + 4 * (mom(1, 1) * mom(1,1))
Set moin (3) ← m1 ^ 2 + m2 ^ 2
Set moin (4) ← m3 ^ 2 + m4 ^ 2
Set moin (5) ← m1 * m3 * m5 + m2 * m4 * m6
Set moin (6) ← m0 * (m3 ^ 2 - m4 ^ 2) + 4 * mom(1, 1) * m3 * m4
Set moin (7) ← m2 * m3 * m5 - m1 * m4 * m6

```

The complex moments up to six orders have been determined to extract the features of the palm area. The number of features which extracted from the palm area is 5 features for each sub bands, which mean 35 features for the seven sub band for each hand person. The previous study indicate that the increase number of features more than 5 features for each sub band shows instability in the behavior of moment features. To compute the complex moments features the following equation has been used:

$$M_p = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} (x + iy)^p f(x, y) \quad (3.20)$$

Algorithm (3.13) illustrates the implemented steps for determining the central complex moments of n order.

Algorithm (3.13) Complex Moment

Input: reci "is the array of record of haar wavelet transform
no "is the number of haar wavelet level
width "is the width of the image
height "is the height of the image
p "is the number of order of moment

Output: cm "is the array of record extracted complex moment features

Procedure:

```

Loop for count=0 to no-1
  Set hm ← height - 1
  Set wm ← width - 1
  Set xcc ← wm / 2
  Set ycc ← hm / 2
  If h > w Then q ← ycc else q ← xcc
  Loop for i = 0 to p
    Set momr(i) ← 0
    Set momi(i) ← 0
  End Loop
  Loop for x = 0 to wm
    Set xx ← (x - xcc) / q
    Loop for y = 0 to hm
      Set yy ← (y - ycc) / q
      Set zr(1) ← xx
      Set zi(1) ← yy
      Set c ← reci(count).ll(x, y)
      Set momr(1) ← momr(1) + zr(1) * c
      Set momi(1) ← momi(1) + zi(1) * c
      Loop for i = 2 to p
        Set j ← i - 1
        Set zr(i) ← zr(j) * xx - zi(j) * yy
        Set zi(i) ← zr(j) * yy + zi(j) * xx
        Set momr(i) ← momr(i) + zr(i) * c
        Set momi(i) ← momi(i) + zi(i) * c
      End Loop
    End Loop
  End Loop
  Loop for i = 1 to p
    cm (i) ← Sqr(momr(i) * momr(i) + momi(i) * momi(i))
  End Loop
End Loop

```


3.7 Feature Analysis

The matching process had been done by using some selected sets of extracted features (whether they are complex or invariant moments). Before starting the matching process, the template feature vector for the hand palm of each person should be determined and registered in a database file. To compute the template vectors, the mean and standard deviation vectors for the feature vectors extracted from different image samples of the hand palm taken for each person are determined. From the values of the standard deviation vector elements the variability of the features could be assessed. The mean and standard deviation vectors are determined by using the following equations [The03]:

$$\overline{F}_{ip} = \frac{1}{n} \sum_{j=1}^n f_{ijp} \quad (4.1)$$

$$\sigma_{ip} = \sqrt{\frac{1}{n} \sum_{j=1}^n (f_{ijp} - \overline{F}_{ip})^2} \quad (4.2)$$

Where

f_{ijp} is the value of the i^{th} moment extracted from j^{th} sample image of the hand palm of p person.

n is the number of sample images taken for p person.

\overline{F}_{ip} is the mean value of the i^{th} moment for p person.

σ_{ip} is the standard deviation of the i^{th} moment for p person.

It was noticed that the dispersion (i.e. standard deviation) values of some features is high, so in order to handle this problem the values of the features, which are far from the mean value have been excluded by replacing the value of the feature by the 0 value in order to enhance the computation of the template vector. After the mean and standard deviation have been determined,

then for each extracted value of each feature whose absolute difference from corresponding mean value is greater than the standard deviation, then this value will be excluded and not used to predetermine the final values of the mean and standard deviation. Algorithm (3.14) illustrates the implemented steps for determining the mean and standard deviation for both complex and invariant moments and how to exclude the inappropriate samples values of these features.

Algorithm (3.14) the computation of the mean and standard deviation vectors for both complex and invariant moment

Input: feat // is the array of extracted moments (whether they are complex moments or moments invariant)
noperson // is the number of persons
nosubband // is the number of wavelet sub bands
nosample // is the number of training samples per person
Output: mean // is the template (mean) feature vectors
std // is the standard deviation of the feature vectors.

Procedure:

```
Loop for pid = 1 to noperson
  Loop for sband = 1 to nosubband
    Loop for feature = 1 to nosample
      Set mean (pid, sband, feature) ← 0
      Loop for sa = 1 to nosample
        Set mean (pid, sband, feature) ← mean (pid, sband, feature) +
          feat (pid, sa, sband, feature)
      End Loop
      Set mean (pid, sband, feature) ← mean (pid, sband, feature) /
        nosample
      Set std (pid, sband, feature) ← 0
      Loop for sample = 1 to nosample
        Set std (pid, sband, feature) ← std (pid, sband, feature) +
          (feat (pid, sample, sband, feature) – mean (pid, sband,
            feature)) ^ 2
      End Loop
```

```
Set std (pid, sband, feature) ← sqr (std (pid, sband, feature) /
nosample)
Set counter ← nosample
Loop for sample = 1 to nosample
  Set val ← abs (feat (pid, sample, sband, feature) - mean (pid,
sband, feature))
  If val > (1.5 * std (pid, sband, feature)) Then
    Set counter = counter - 1
    Set feat2(pid, sample, sband, feature) ← 0
  Else
    Set feat2 (pid, sample, sband, feature) ← temp (pid, sample,
sband, feature)
  End If
End Loop
Set mean (pid, sband, feature) = 0
Loop for sample = 1 to nosample
  Set mean (pid, sband, feature) ← mean (pid, sband,
feature) + feat (pid, sample, sband, feature)
End Loop
Set mean (pid, sband, feature) ← mean (pid, sband, feature) /
counter
Set std (pid, sband, feature) ← 0
Loop for sample = 1 to nosample
  If feat2 (pid, sample, sband, feature) <> 0 Then
    Set std (pid, sband, feature) ← std (pid, sband, feature)
+ (feat2 (pid, sample, sband, feature) - mean (pid,
sband, feature)) ^ 2
  End If
End Loop
Set std (pid, sband, feature) ← sqr (std (pid, sband, feature) /
counter)
End Loop
End Loop
End Loop
```

The adopted mechanism to handle the feature analysis task was aimed to find out the lowest possible combinations of features that can lead to good hand palm recognition results. The size of features combination was gradually

increases and at each increment step the best sets of features which can give the highest recognition rate are found by using heuristic search. In this work, the sets of features combinations was started with single features and gradually expanded to double features, and then triple and so on until reaching combinations of six features, where acceptable levels of recognition rates have been reached. The algorithm (3.15) illustrates the implemented steps to perform the exhaustive search for different possible combinations of two moment features, this search could be applied for both complex and invariant moments. The algorithm (3.16) illustrates the steps of the applied exhaustive search for different possible combinations of four moment features (whether they are complex and invariant moments), while the algorithm (3.17) illustrates the exhaustive search for different possible combination of six moment features.

Algorithm (3.15) Testing the identification rate for the combinations of 2 moment features.

Input: noperson // is the number of persons
nofeature // is the number of extracted feature for each sample image of person
nosample // is the number of training samples per person
mean // is the template (mean) feature vectors
std // is the standard deviation of the feature vectors.
feat2 // is the array of extracted moments after exclusion the inappropriate features (whether they are complex moments or moments invariant)

Output: success // is the array of success recognition rates

Procedure:

counter = 1

Loop for $k_1 = 1$ to nofeature

Set $sband_1 \leftarrow ((k_1 - 1) \setminus nosample) + 1$

Set $feature_1 \leftarrow ((k_1 - 1) \bmod nosample) + 1$

Loop for $k_2 = k_1 + 1$ to nofeature

Set $sband_2 \leftarrow ((k_2 - 1) \setminus nosample) + 1$

```

Set feature2 ← ((k2 - 1) mod nosample) + 1
Loop for pid = 1 to noperson
  Loop for sa = 1 to nosample
    Set i ← 1
    Loop for p_id = 1 to noperson
      If feat2 (pid, sa, sband1, feature1) <> 0 Then

```

$$dif(i) = \sum_{k \in \{k_1, k_2\}} \frac{feat2(pid, sa, sband_k, feature_k) - mean(p_id, sband_k, feature_k)}{std(p_id, sband_k, feature_k)}$$

```

    Set pos(i) ← p_id
    Set i ← i + 1
  End If
End Loop
Set di ← dif(1)
Set po ← pos(1)
Loop for i = 2 To noperson
  If dif(i) < di Then
    Set di ← dif(i)
    Set po ← pos(i)
  End If
End Loop
If pid = po Then Set sss ← sss + 1
End Loop
End Loop
success (counter) ← sss
counter ← counter + 1
End Loop
End Loop

```

Algorithm (3.16) Testing the identification rate for the combinations of 4 moment features.

Input: noperson // is the number of persons
 nofeature // is the number of extracted feature for each sample image of person
 nosample // is the number of training samples per person
 mean // is the template (mean) feature vectors
 std // is the standard deviation of the feature vectors.
 feat2 // is the array of extracted moments after exclusion the inappropriate features (whether they are complex moments or moments invariant)

Output: success // is the array of success recognition rates

Procedure:

Set counter $\leftarrow 1$
 Loop for $k_1 = 1$ to nofeature
 Set sband1 $\leftarrow ((k_1 - 1) \setminus \text{nosample}) + 1$
 Set feature1 $\leftarrow ((k_1 - 1) \bmod \text{nosample}) + 1$
 Loop for $k_2 = k_1 + 1$ To nofeature
 Set sband2 $\leftarrow ((k_2 - 1) \setminus \text{nosample}) + 1$
 Set feature2 $\leftarrow ((k_2 - 1) \bmod \text{nosample}) + 1$
 Loop for $k_3 = k_2 + 1$ to nofeature
 Set sband3 $\leftarrow ((k_3 - 1) \setminus \text{nosample}) + 1$
 Set feature3 $\leftarrow ((k_3 - 1) \bmod \text{nosample}) + 1$
 Loop for $k_4 = k_3 + 1$ to nofeature
 Set sband4 $\leftarrow ((k_4 - 1) \setminus \text{nosample}) + 1$
 Set feature4 $\leftarrow ((k_4 - 1) \bmod \text{nosample}) + 1$
 Set sss $\leftarrow 0$
 Loop for pid = 1 to noperson
 Loop for sa = 1 to nosample
 Set i $\leftarrow 1$
 Loop for p_id = 1 to noperson
 If feat2(pid, sample, sband1, feature1) $\neq 0$ Then

$$dif(i) = \sum_{k \in \{k_1, k_2, k_3, k_4\}} \frac{feat2(pid, sa, sband_k, feature_k) - mean(p_id, sband_k, feature_k)}{std(p_id, sband_k, feature_k)}$$

```
        Set pos(i) ← p_id
    Set i ← i + 1
End If
End Loop
Set di ← dif(1)
Set po ← pos(1)
Loop for i = 2 To noperson
    If dif(i) < di Then
        Set di ← dif(i)
        Set po ← pos(i)
    End If
End Loop
If pid = po Then Set sss ← sss + 1
End Loop
End Loop
If sss > 20 Then
    success (counter) ← sss
    counter ← counter + 1
End If
End Loop
End Loop
End Loop
```

Algorithm (3.17) Testing the identification rate for the combinations of 6 moment features.

Input: noperson // is the number of persons
nofeature // is the number of extracted feature for each sample image of person
nosample // is the number of training samples per person
mean // is the template (mean) feature vectors
std // is the standard deviation of the feature vectors.
feat2 // is the array of extracted moments after exclusion the inappropriate features (whether they are complex moments or moments invariant)

Output: success // is the array of success recognition rates

Procedure:

```

Set counter ← 1
Loop for k1 = 1 to nofeature
  Set sband1 ← ((k1 - 1) \ nosample) + 1
  Set feature1 ← ((k1 - 1) mod nosample) + 1
  Loop for k2 = k1 + 1 to nofeature
    Set sband2 ← ((k2 - 1) \ nosample) + 1
    Set feature2 ← ((k2 - 1) mod nosample) + 1
    Loop for k3 = k2 + 1 to feature_no
      Set sband3 ← ((k3 - 1) \ nosample) + 1
      Set feature3 ← ((k3 - 1) mod nosample) + 1
      Loop for k4 = k3 + 1 to nofeature
        Set sband4 ← ((k4 - 1) \ nosample) + 1
        Set feature4 ← ((k4 - 1) mod nosample) + 1
        Loop for k5 = k4 + 1 to nofeature
          Set sband5 ← ((k5 - 1) \ nosample) + 1
          Set feature5 ← ((k5 - 1) mod nosample) + 1
          Loop for k6 = k5 + 1 to nofeature
            Set sband6 ← ((k6 - 1) \ nosample) + 1
            Set feature6 ← ((k6 - 1) mod nosample) + 1
            Set sss ← 0
        Loop for pid = 1 to noperson
          Loop for sa = 1 to nosample
            Set i ← 1
            Loop for p_id = 1 To noperson
              If feat2(pid, sa, sband1, feature1) <> 0 Then

```

$$dif(i) = \frac{\sum_{k \in \{k_1, k_2, k_3, k_4, k_5, k_6\}} feat2(pid, sa, sband_k, feature_k) - mean(p_id, sband_k, feature_k)}{std(p_id, sband_k, feature_k)}$$

```

          Set pos(i) ← p_id
          Set i ← i + 1
        End If
      End Loop
    End Loop
  Set di ← dif(1)
  Set po ← pos(1)
  Loop for i = 2 To noprson

```




Chapter Four

Chapter Four

Tests and Results

4.1 Introduction

This chapter is dedicated to demonstrate the results of the tests which have been conducted to assess the performance of the established system. In the next section the step of the test procedures are illustrated. Then, the specifications of the tested hand images are given, and some samples of the preprocessing stages are shown. Finally, some samples of the matching stages are given.

4.2 Test Material

In this research, 5 samples of images of the right hand of 10 persons have been taken. Each image sample as a bitmap data with resolution 24 bit/pixel and the size of each image is 200 x 200. The type of scanner that used to collect the samples of images is canon scan and the resolution of the image is 100 dpi. Figure (4.1) shows samples of the hand images which have been taken for 10 persons.



(Ahmed)



(Aunss)



(Basma)



(Eiman)



(Einas)



(Farah)



(Sinan)



(Usama)



(Zahraa)



(Zainab)

Figure (4.1) Samples of the tested hand images

4.3 Experimental Results

In this section some samples of the results of the preprocessing, feature extraction and matching stages are listed in the following sub-sections.

4.3.1 The Results of the Preprocessing Stage

The input image to the preprocessing stage was color image, the first step in the preprocessing stage is the conversion of the color image to binary image (black and white) which had been done by using the binarization (i.e., thresholding) method. Figure (4.2) shows the result of conversion a color image to binary image.

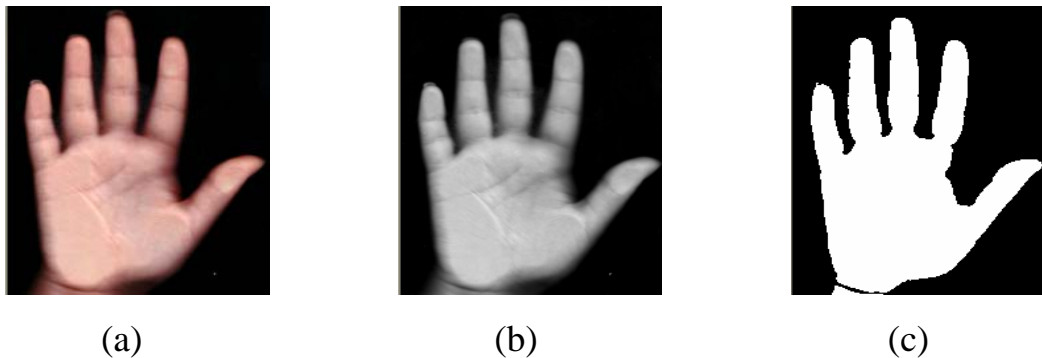


Figure (4.2) The conversion of the color image to binary image,
(a) color image, (b) gray image, (c) binary image

The next step in this stage is the edge detection method, where Laplace filter had been used to extract the boundary of the hand. Figure (4.3) shows the boundary image after applying Laplace filter. The last step in this stage is using chain coding algorithm to extract the sequence of the boundary of the hand.

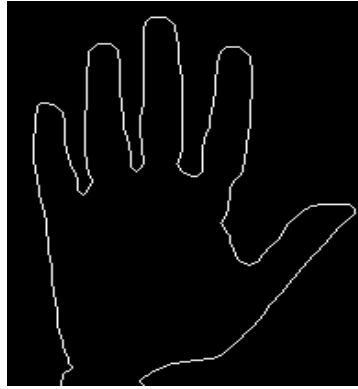


Figure (4.3) The boundary image of a hand

4.3.2 The Results of Hand Palm Area Extraction

In this stage, the first step is extracting the joint points of the hand, then using Bresenham algorithm to establish a line between these joint points. Figure (4.4) shows the hand palm boundary image before and after using Bresenham algorithm.

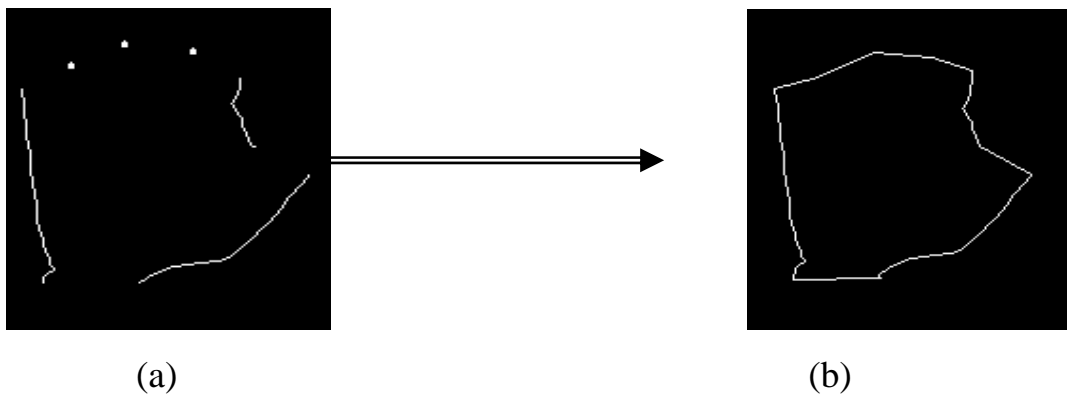


Figure (4.4) hand-palm boundary (a. Before applying Bresenham algorithm; b. After applying Bresenham algorithm)

After the hand palm image was extracted, the palm area was marked by white colored pixels; this was done by using the seed filling algorithm. Figure (4.5) shows the hand palm image before and after using the seed filling algorithm.

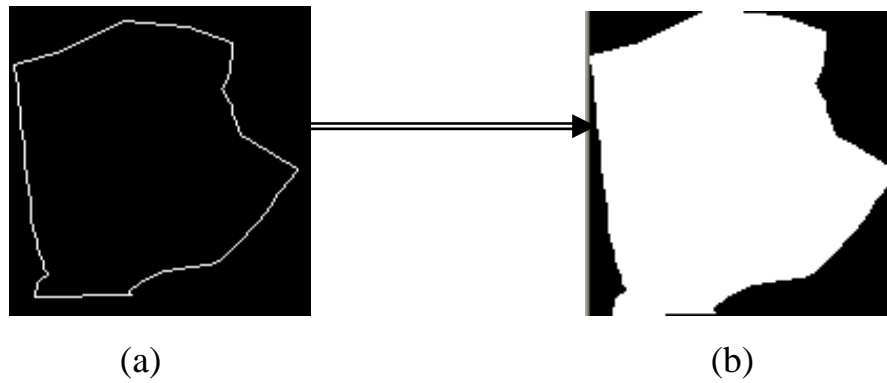


Figure (4.5) The extracted palm image, (a. hand palm boundary; b. hand palm area)

After the application of seed filling algorithm, the color palm image was constructed by replacing only the white color of the marked pixels by the actual gray values of these white pixels. Then the Laplace filter was applied on the gray hand palm image to find all the lines and ridges within the central circular area in the hand palm region. As a next step Haar wavelet transform was applied to decompose the image into sub-bands (LL, LH, HL, and HH). Figure (4.6) shows the sub-bands (LL, LH, HL and HH) after using Haar wavelet transform (for 2 passes).

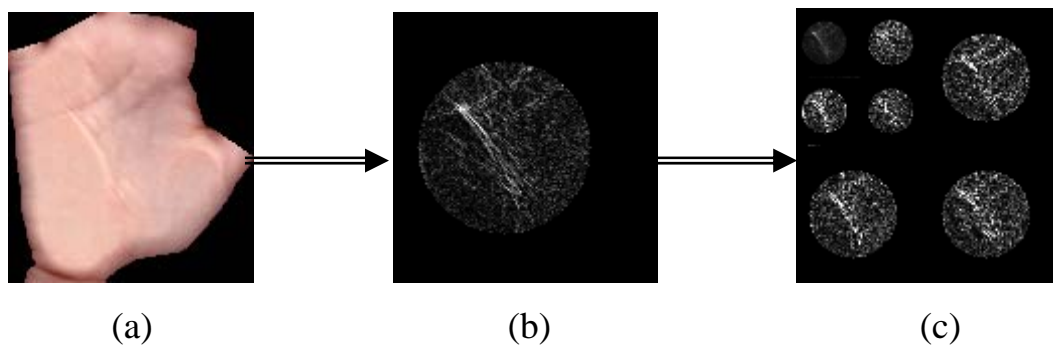


Figure (4.6) The result of decomposition the image of hand palm area by using Haar wavelet (a. gray image; b. circle palm area; c. result of Haar wavelet transform)

In the last step the invariant moments and complex moments have been determined as discriminating features for the palm area, then the moments value have been stored into a database file.

4.3.3 The Templates Features Vectors

In this stage, the mean and standard deviation feature vectors had been calculated from the features vectors which have been extracted from the samples of the hand palm images for each person. These mean and standard deviation vectors are determined for both invariant and complex moments. Table (4.1) lists the mean and standard deviation for complex moment features and table (4.2) shows the mean and standard deviation for invariant moment features.

Table (4.1) The mean and standard deviation for complex moment features

Person Id	Sub-bands No. & Name	Mean values (M1, M2, M3, M4, M5)	Standard Deviation values (std1, std2, std3, std4, std5)
1	HL1	673.31 , 209.37 , 103.68 , 50.31 , 29.63	75.09 , 180.61 , 46.22 , 8.74 , 17.78
	HH1	344.18 , 182.55 , 142.67 , 57.94 , 46.57	22.59 , 74.92 , 46.58 , 19.39 , 10.60
	LH1	929.66 , 311.30 , 160.21 , 64.20 , 49.89	78.44 , 99.21 , 24.66 , 19.87 , 34.03
	HL2	931.96 , 135.09 , 50.75 , 72.02 , 33.73	87.69 , 26.09 , 26.83 , 23.01 , 6.97
	HH2	293.78 , 65.03 , 78.95 , 70.22 , 35.56	68.18 , 28.36 , 30.08 , 11.89 , 18.73
	LH2	917.65 , 165.99 , 76.78 , 36.19 , 28.57	83.69 , 83.09 , 21.50 , 18.78 , 11.03
	LL	5186.85 , 556.58 , 386.14 , 159.06 , 69.61	505.85 , 109.29 , 94.08 , 64.62 , 9.69
2	HL1	446.89 , 158.18 , 106.66 , 55.69 , 41.68	160.63 , 53.94 , 15.63 , 7.29 , 9.70
	HH1	127.68 , 195.58 , 98.99 , 55.87 , 21.59	39.17 , 84.33 , 71.64 , 0.15 , 9.10
	LH1	535.64 , 169.66 , 153.86 , 74.97 , 30.59	298.93 , 32.21 , 43.02 , 14.98 , 8.93
	HL2	768.06 , 274.66 , 123.20 , 42.18 , 53.32	77.69 , 40.87 , 0.02 , 17.64 , 2.28
	HH2	132.88 , 148.09 , 66.86 , 35.19 , 24.14	32.18 , 25.34 , 35.30 , 19.19 , 4.39
	LH2	785.96 , 149.29 , 57.25 , 52.00 , 23.37	225.57 , 38.94 , 25.48 , 12.30 , 1.60
	LL	5442.86 , 633.48 , 264.64 , 62.87 , 67.98	1266.97 , 465.75 , 114.87 , 49.76 , 39.16
3	HL1	457.28 , 174.38 , 50.03 , 53.14 , 20.16	66.07 , 34.06 , 26.54 , 13.69 , 8.29
	HH1	315.04 , 98.77 , 68.54 , 27.53 , 24.91	142.92 , 21.33 , 29.93 , 7.13 , 5.85
	LH1	1069.09 , 306.28 , 74.18 , 84.91 , 30.84	357.84 , 33.62 , 37.62 , 31.72 , 7.07
	HL2	621.91 , 145.07 , 46.15 , 23.73 , 27.67	86.32 , 18.50 , 31.96 , 9.17 , 11.69
	HH2	170.63 , 92.06 , 49.56 , 29.47 , 13.79	54.17 , 20.68 , 15.81 , 6.55 , 5.79
	LH2	701.53 , 177.07 , 56.98 , 19.17 , 14.01	283.20 , 61.48 , 23.10 , 4.43 , 5.89
	LL	4134.13 , 194.08 , 416.48 , 178.30 , 30.10	574.41 , 64.18 , 97.35 , 23.69 , 19.96
4	HL1	473.75 , 267.73 , 81.55 , 59.59 , 49.63	72.38 , 110.42 , 11.43 , 3.69 , 21.81
	HH1	336.73 , 197.43 , 166.18 , 70.80 , 41.08	172.47 , 80.68 , 11.96 , 8.26 , 4.41

Chapter Four: Tests and Results

	LH1	582.42 , 137.69 , 80.92 , 96.67 , 51.44	88.99 , 69.06 , 25.74 , 11.54 , 1.27
	HL2	581.19 , 145.92 , 71.44 , 41.05 , 25.30	28.44 , 16.22 , 26.47 , 15.57 , 1.55
	HH2	175.47 , 106.49 , 41.29 , 30.30 , 16.59	81.63 , 15.86 , 5.26 , 20.03 , 7.30
	LH2	589.78 , 174.17 , 75.34 , 30.41 , 26.73	37.14 , 117.59 , 18.86 , 3.28 , 14.72
	LL	2683.18 , 699.30 , 655.37 , 240.52 , 134.26	1018.39 , 49.27 , 132.15 , 44.92 , 17.34
5	HL1	754.80 , 279.09 , 102.88 , 80.92 , 30.03	310.20 , 159.28 , 52.08 , 20.83 , 12.88
	HH1	443.82 , 301.35 , 191.61 , 53.20 , 41.84	153.86 , 20.89 , 64.71 , 8.98 , 9.89
	LH1	1142.13 , 291.27 , 135.49 , 99.08 , 45.95	50.07 , 143.76 , 68.46 , 43.42 , 22.49
	HL2	770.20 , 110.71 , 121.01 , 26.88 , 26.31	213.12 , 69.34 , 30.55 , 10.66 , 3.48
	HH2	225.99 , 110.57 , 63.57 , 41.12 , 21.92	23.78 , 16.61 , 19.56 , 18.74 , 1.45
	LH2	852.54 , 115.55 , 60.97 , 26.54 , 28.48	175.86 , 27.23 , 40.81 , 3.63 , 10.64
	LL	2674.99 , 819.03 , 237.38 , 226.54 , 33.57	359.84 , 213.25 , 110.57 , 19.50 , 16.99
6	HL1	806.62 , 342.08 , 146.46 , 82.29 , 20.22	110.97 , 106.91 , 33.37 , 10.72 , 5.97
	HH1	309.49 , 275.07 , 75.21 , 91.34 , 53.51	142.35 , 54.86 , 23.70 , 22.48 , 6.97
	LH1	701.03 , 128.15 , 76.47 , 71.64 , 26.44	279.02 , 34.23 , 47.06 , 39.18 , 18.75
	HL2	638.87 , 117.42 , 61.81 , 32.96 , 17.16	62.45 , 41.63 , 8.35 , 7.70 , 7.21
	HH2	81.14 , 118.90 , 40.07 , 33.48 , 13.77	45.79 , 10.55 , 27.96 , 5.29 , 3.49
	LH2	744.61 , 118.91 , 32.37 , 22.83 , 20.04	127.91 , 54.79 , 16.16 , 10.65 , 14.91
	LL	2837.43 , 806.92 , 458.95 , 109.07 , 57.13	141.91 , 127.23 , 13.48 , 23.20 , 7.68
7	HL1	796.42 , 223.15 , 57.36 , 52.33 , 14.47	375.01 , 71.26 , 21.69 , 9.64 , 4.25
	HH1	447.24 , 231.62 , 176.67 , 93.27 , 38.36	103.97 , 09.46 , 14.62 , 28.81 , 3.03
	LH1	762.08 , 248.58 , 146.72 , 93.35 , 46.92	189.21 , 138.51 , 43.92 , 32.56 , 16.94
	HL2	799.88 , 366.44 , 146.20 , 69.03 , 27.93	65.82 , 21.14 , 18.51 , 19.62 , 8.99
	HH2	171.51 , 110.34 , 58.46 , 29.76 , 22.85	0.73 , 60.35 , 4.68 , 6.71 , 8.60
	LH2	825.52 , 199.78 , 47.16 , 44.69 , 8.13	13.15 , 28.09 , 13.17 , 1.00 , 1.34
	LL	4890.81 , 870.27 , 614.02 , 176.84 , 115.11	314.58 , 114.47 , 150.25 , 61.70 , 15.36
8	HL1	670.35 , 264.40 , 115.73 , 45.78 , 21.82	97.37 , 18.15 , 10.00 , 30.36 , 13.33
	HH1	380.78 , 106.88 , 51.12 , 58.29 , 20.23	48.73 , 37.37 , 20.68 , 7.42 , 5.83
	LH1	624.53 , 135.33 , 131.97 , 58.79 , 51.50	128.16 , 30.58 , 16.53 , 6.73 , 14.08
	HL2	827.46 , 227.46 , 83.00 , 39.45 , 29.95	58.26 , 7.32 , 16.26 , 11.18 , 6.85
	HH2	145.83 , 91.81 , 14.98 , 22.39 , 6.92	20.21 , 20.02 , 13.90 , 14.68 , 1.59
	LH2	727.22 , 301.79 , 100.33 , 41.87 , 40.82	34.85 , 67.46 , 12.55 , 8.13 , 5.52
	LL	4219.88 , 661.98 , 263.71 , 100.72 , 54.88	790.67 , 53.40 , 85.65 , 21.49 , 25.79
9	HL1	861.67 , 241.19 , 98.63 , 67.79 , 15.93	35.00 , 47.03 , 18.90 , 34.03 , 9.38
	HH1	404.36 , 132.07 , 139.00 , 34.31 , 32.50	91.10 , 30.26 , 23.62 , 17.89 , 9.91
	LH1	442.99 , 217.09 , 59.53 , 46.55 , 18.09	150.37 , 122.78 , 23.76 , 6.92 , 1.40
	HL2	601.68 , 95.94 , 55.58 , 32.93 , 15.83	48.02 , 39.25 , 38.14 , 12.12 , 6.53
	HH2	152.23 , 61.55 , 59.35 , 13.74 , 14.30	43.04 , 23.24 , 24.36 , 4.32 , 2.26
	LH2	781.21 , 64.25 , 54.07 , 34.74 , 20.79	65.34 , 34.24 , 17.74 , 13.30 , 7.10
	LL	3578.74 , 217.49 , 278.04 , 41.55 , 24.58	659.81 , 70.88 , 44.21 , 19.63 , 10.15
10	HL1	984.69 , 158.94 , 120.55 , 66.65 , 15.09	300.50 , 83.53 , 23.61 , 17.62 , 2.55
	HH1	267.04 , 160.01 , 113.40 , 43.97 , 48.59	22.61 , 32.49 , 4.94 , 34.22 , 4.67
	LH1	565.04 , 312.33 , 107.36 , 61.45 , 28.77	42.34 , 52.30 , 15.24 , 11.77 , 6.57
	HL2	538.11 , 134.27 , 89.23 , 47.22 , 27.11	42.50 , 31.74 , 28.43 , 10.08 , 5.06
	HH2	174.62 , 56.50 , 47.00 , 26.15 , 21.40	22.39 , 27.16 , 11.56 , 9.40 , 1.83
	LH2	568.83 , 214.55 , 105.98 , 38.15 , 23.74	114.01 , 132.89 , 29.18 , 12.89 , 9.08
	LL	4210.19 , 647.84 , 312.60 , 63.06 , 39.60	459.44 , 193.46 , 51.76 , 15.71 , 17.30

Table (4.2) The mean and standard deviation for invariant moment features

Person ID	sub-band No. & Name	Mean Values (M1, M2, M3, M4, M5, M6, M7)	Standard Deviation Values (std1, std2, std3, std4, std5, std6, std7)
1	HL1	1.48E-02, 2.07E-02, 9.96E-03, 3.63E-03, -3.23E06, -4.77E-04, 1.99E-05	2.14E-02, 1.69E-02, 3.03E-03, 1.63E-03, 1.21E05, 3.84E-04, 1.47E-05
	HH1	2.54E-01, 6.41E-02, 2.10E-02, 2.88E-02, 1.59E03, 1.00E-02, -2.26E-04	1.28E-01, 6.18E-02, 2.25E-02, 3.33E-02, 2.61E03, 1.47E-02, 3.36E-04
	LH1	-2.53E-02, 1.99E-01, 1.01E-01, 7.56E-02, 1.35E02, 3.41E-02, -2.42E-03	3.26E-01, 1.62E-01, 1.06E-01, 7.90E-02, 2.17E-02, 6.53E-02, 3.48E-03
	HL2	-4.35E+00, 9.53E+01, 5.48E+02, 2.96E+02, 1.98E+05, -3.65E+03, 4.66E+04	4.55E+00, 7.94E+01, 4.60E+02, 2.69E+02, 2.02E+05, 3.49E+03, 5.71E+04
	HH2	1.62E+00, 5.47E+00, 1.98E+01, 1.91E+01, 4.41E+02, 3.47E+01, -7.93E-01	1.66E+00, 3.27E+00, 7.31E+00, 8.64E+00, 2.97E+02, 4.38E+01, 4.29E+00
	LH2	2.04E+00, 2.26E+01, 1.10E+02, 8.35E+01, 1.19E+04, 2.49E+02, -5.89E+02	3.56E+00, 1.39E+01, 6.39E+01, 6.45E+01, 1.33E+04, 6.02E+02, 6.92E+02
	LL	2.47E+00, 5.85E+00, 1.39E+01, 1.51E+01, 2.20E+02, 3.64E+01, -1.56E-01	3.46E-02, 1.61E-01, 4.36E-01, 4.89E-01, 1.42E+01, 1.56E+00, 4.50E-02
2	HL1	1.65E-01, 8.50E-02, 4.98E-02, 1.91E-02, 5.69E04, 6.76E-03, 3.09E-04	1.27E-01, 7.55E-02, 4.51E-02, 2.12E-02, 9.42E04, 8.09E-03, 3.58E-04
	HH1	1.53E-01, 1.80E-02, 5.21E-03, 3.69E-03, 7.89E06, 3.91E-04, 2.21E-06	1.89E-02, 5.60E-03, 1.61E-03, 2.48E-03, 1.12E05, 2.77E-04, 5.40E-06
	LH1	-8.74E-02, 6.91E-01, 5.51E-01, 1.55E-01, 1.22E03, -5.46E-02, -2.01E-02	6.02E-02, 6.53E-02, 2.08E-01, 3.90E-02, 8.16E04, 3.72E-02, 1.64E-02
	HL2	2.20E+00, 2.04E+01, 1.17E+02, 2.81E+01, 1.98E+03, 8.65E+01, 5.35E+02	2.23E+00, 1.21E+01, 3.33E+01, 3.05E+01, 3.47E+03, 2.04E+02, 4.20E+02
	HH2	1.55E+00, 4.51E+00, 1.28E+01, 7.17E+00, 6.20E+01, 1.19E+01, 2.22E+00	7.10E-01, 3.23E-01, 3.22E+00, 2.42E+00, 3.53E+01, 1.01E+01, 1.22E+01
	LH2	3.67E+00, 1.34E+01, 4.95E+01, 3.75E+01, 2.98E+03, 1.81E+02, -2.41E+02	2.05E+00, 1.09E+01, 4.39E+01, 3.49E+01, 2.97E+03, 1.77E+02, 2.44E+02
	LL	2.29E+00, 5.02E+00, 1.13E+01, 1.23E+01, 1.45E+02, 2.75E+01, -2.18E-04	3.37E-02, 1.38E-01, 4.31E-01, 5.18E-01, 1.19E+01, 1.54E+00, 7.97E-03
3	HL1	3.53E-01, 1.85E-01, 6.37E-02, 1.42E-01, 1.93E02, -5.85E-02, 9.07E-03	2.51E-01, 2.60E-01, 9.99E-02, 1.53E-01, 3.28E02, 1.09E-01, 1.56E-02
	HH1	2.06E-01, 3.63E-02, 1.33E-02, 9.36E-03, 8.56E05, 4.91E-04, -2.10E-05	4.93E-02, 2.17E-02, 8.26E-03, 3.02E-03, 3.90E05, 1.89E-03, 3.36E-05
	LH1	-8.92E-02, 2.63E-01, 8.71E-02, 3.02E-02, 1.08E03, -1.56E-02, -3.13E-03	1.06E-01, 2.38E-01, 6.71E-02, 2.64E-02, 1.65E03, 2.01E-02, 3.38E-03
	HL2	5.35E-01, 1.14E+01, 5.31E+01, 1.46E+01, 3.75E+02, -1.23E+01, 5.46E+02	1.81E+00, 8.74E+00, 5.09E+01, 1.15E+01, 3.68E+02, 7.44E+01, 6.68E+02
	HH2	1.57E+00, 8.78E+00, 3.95E+01, 1.41E+01, 1.75E+02, -4.50E+00, -9.96E+01	1.11E+00, 6.93E+00, 4.93E+01, 7.00E+00, 1.38E+02, 5.63E+01, 1.74E+02
	LH2	-2.50E-01, 8.84E+00, 6.87E+01, 3.09E+01, 2.06E+03, -9.82E+01, -5.08E+02	2.01E+00, 5.51E+00, 5.13E+01, 2.54E+01, 2.05E+03, 1.09E+02, 4.36E+02
	LL	2.28E+00, 4.98E+00, 1.11E+01, 1.21E+01, 1.40E+02, 2.69E+01, 1.73E-02	2.41E-02, 1.09E-01, 3.66E-01, 3.74E-01, 8.64E+00, 1.12E+00, 1.36E-02
	HL1	7.16E-02, 2.27E-01, 1.09E-01, 9.16E-02	3.76E-01, 2.11E-01, 1.09E-01, 9.25E-02

Chapter Four: Tests and Results

4		1.72E02, -3.65E-02, 6.37E-03	2.59E02, 8.86E-02, 1.06E-02
	HH1	3.94E-01, 1.46E-01, 5.19E-02, 6.10E-02, 2.03E03, 2.37E-02, -2.39E-04	5.39E-02, 2.05E-02, 1.33E-02, 1.96E- 02, 1.56E03, 8.38E-03, 3.60E-04
	LH1	2.05E-01, 2.72E+00, 7.32E-01, 2.07E-01, - 1.15E01, 1.44E-01, 8.89E-02	9.19E-02, 2.14E+00, 6.73E-01, 1.85E-01, 1.15E01, 1.40E-01, 8.87E-02
	HL2	1.61E+00, 8.57E+00, 3.59E+01, 1.60E+01, 9.74E+02, 6.51E+01, 2.09E+02	1.29E+00, 7.31E+00, 3.03E+01, 2.34E+01, 1.71E+03, 1.12E+02, 2.72E+02
	HH2	3.47E+00, 3.26E+01, 1.74E+02, 4.67E+01, 3.87E+03, 3.00E+02, -5.16E+03	4.32E-01, 3.06E+01, 1.86E+02, 3.23E+01, 3.99E+03, 2.89E+02, 6.30E+03
	LH2	1.49E+00, 5.35E+00, 1.94E+01, 7.52E+00, 1.57E+02, 1.91E+01, -5.77E+00	7.63E-01, 2.50E+00, 7.27E+00, 9.35E+00, 2.82E+02, 3.14E+01, 2.84E+01
	LL	2.42E+00, 5.65E+00, 1.34E+01, 1.45E+01, 2.04E+02, 3.46E+01, -1.61E-01	2.52E-02, 1.22E-01, 4.32E-01, 4.50E-01, 1.28E+01, 1.45E+00, 6.99E-02
5	HL1	4.23E-01, 2.36E-01, 1.34E-01, 7.30E-02, 8.04E03, -9.67E-03, 7.12E-03	2.29E-01, 1.85E-01, 9.84E-02, 4.93E-02, 7.91E03, 5.56E-02, 7.91E-03
	HH1	3.96E-01, 7.64E-01, 4.31E-01, 1.81E-01, 1.78E01, 3.06E-01, -7.60E-02	3.75E-01, 1.29E+00, 7.40E-01, 3.02E-01, 3.09E01, 5.29E-01, 1.32E-01
	LH1	-1.20E+00, 1.59E+01, 6.70E+00, 3.28E+00, 4.23E+01, -2.53E+01, -4.39E+01	1.95E+00, 2.74E+01, 1.15E+01, 5.65E+00, 7.32E+01, 4.38E+01, 7.60E+01
	HL2	3.24E+00, 1.37E+01, 6.19E+01, 4.97E+01, 3.73E+03, -5.92E+00, 4.30E+02	7.03E-01, 5.56E+00, 4.69E+01, 2.86E+01, 3.89E+03, 2.43E+02, 7.15E+02
	HH2	2.62E+00, 3.94E+01, 2.14E+02, 5.29E+01, - 1.16E+04, 1.09E+02, 3.77E+03	6.31E-01, 5.40E+01, 3.30E+02, 4.73E+01, 2.14E+04, 6.41E+01, 6.54E+03
	LH2	1.64E+00, 1.15E+01, 4.92E+01, 3.51E+01, 3.64E+03, 1.62E+02, -8.31E+01	2.33E+00, 1.15E+01, 3.98E+01, 5.10E+01, 6.29E+03, 3.01E+02, 6.15E+01
	LL	2.40E+00, 5.51E+00, 1.29E+01, 1.41E+01, 1.91E+02, 3.32E+01, -1.26E-01	2.60E-02, 1.26E-01, 4.48E-01, 4.58E-01, 1.25E+01, 1.45E+00, 1.81E-02
6	HL1	3.32E-01, 1.10E-01, 3.69E-02, 4.49E-02, 2.09E03, 1.56E-02, 2.53E-04	3.68E-02, 4.50E-02, 2.29E-02, 1.55E-02, 1.64E-03, 8.52E-03, 2.91E-04
	HH1	-1.32E-01, 4.23E-01, 1.96E-01, 7.71E-02, 1.44E-02, -9.39E-02, 4.05E-03	4.46E-01, 3.52E-01, 1.94E-01, 9.59E-02, 2.57E-02, 1.25E-01, 5.28E-03
	LH1	3.89E-01, 2.68E-01, 1.05E-01, 8.05E-02, 9.07E-03, 4.66E-02, -2.16E-03	1.35E-01, 1.29E-01, 5.38E-02, 5.00E-02, 8.65E-03, 3.45E-02, 3.51E-03
	HL2	1.55E+00, 1.50E+01, 6.95E+01, 4.10E+01, 2.95E+03, -3.10E+01, 9.17E+02	2.48E+00, 9.70E+00, 6.75E+01, 2.53E+01, 3.25E+03, 2.32E+02, 1.58E+03
	HH2	2.73E+00, 8.35E+00, 2.67E+01, 2.15E+01, 5.38E+02, 6.40E+01, -3.49E+01	4.85E-01, 2.32E+00, 1.13E+01, 6.20E+00, 2.92E+02, 2.80E+01, 6.60E+01
	LH2	-8.45E-02, 6.71E+00, 3.15E+01, 1.12E+01, 2.40E+02, 2.06E+00, -1.47E+02	1.13E+00, 4.18E+00, 2.78E+01, 9.61E+00, 2.59E+02, 4.70E+01, 2.34E+02
	LL	2.43E+00, 5.65E+00, 1.35E+01, 1.45E+01, 2.03E+02, 3.45E+01, 6.65E-03	2.67E-02, 9.38E-02, 3.37E-01, 3.55E-01, 1.01E+01, 1.13E+00, 1.50E-02
7	HL1	2.22E-01, 1.03E+00, 3.69E-01, 3.38E-01, 1.53E-01, 1.69E-01, 4.08E-02	7.45E-01, 5.60E-01, 1.71E-01, 2.20E-01, 1.30E-01, 4.60E-01, 2.76E-02
	HH1	4.22E-02, 7.02E-01, 4.01E-01, 3.18E-01, 1.16E-01, -1.08E-02, 4.86E-03	6.69E-01, 3.51E-01, 2.19E-01, 4.22E-02, 1.02E04, 2.48E-01, 1.07E-02
	LH1	8.17E-01, 9.38E-01, 2.51E-01, 4.47E-01, 2.49E-01, 5.63E-01, 1.76E-02	6.64E-01, 7.72E-01, 2.07E-01, 4.02E-01, 2.41E01, 5.20E-01, 1.75E-02
	HL2	-7.58E+00, 3.62E+02, 2.12E+03, 1.27E+03, 4.96E+06, -3.67E+04, 8.71E+05	1.16E+01, 4.26E+02, 2.52E+03, 1.51E+03, 6.07E+06, 4.52E+04, 1.07E+06

Chapter Four: Tests and Results

	HH2	2.84E+00, 7.15E+00, 1.98E+01, 2.06E+01, 5.00E+02, 5.82E+01, 1.51E+00	2.97E-01, 2.24E+00, 8.81E+00, 9.35E+00, 3.78E+02, 3.31E+01, 4.30E-01
	LH2	-4.50E-01, 3.21E+00, 2.49E+01, 2.57E+01, 5.92E+02, -3.21E+01, -1.75E+01	1.23E+00, 3.33E-01, 1.25E+01, 2.17E+01, 5.98E+02, 3.03E+01, 1.50E+01
	LL	2.44E+00, 5.77E+00, 1.39E+01, 1.50E+01, 2.19E+02, 3.61E+01, 6.05E-02	1.03E-01, 4.93E-01, 1.78E+00, 1.86E+00, 5.40E+01, 6.00E+00, 3.17E-02
8	HL1	8.20E-01, 1.28E+00, 5.26E-01, 4.74E-01, 4.58E-01, 7.38E-01, 9.22E-02	5.49E-01, 1.24E+00, 5.22E-01, 4.67E-01, 5.28E-01, 8.05E-01, 1.05E-01
	HH1	9.53E-03, 1.18E-01, 4.63E-02, 9.12E-02, 6.44E-03, -8.08E-03, 1.04E-03	4.20E-01, 1.64E-02, 1.73E-02, 3.24E-02, 3.44E03, 3.14E-02, 1.83E-03
	LH1	-9.79E-03, 9.41E-02, 2.74E-02, 3.29E-02, 1.45E-03, -8.03E-03, 3.18E-06	2.54E-01, 4.71E-02, 1.39E-02, 2.57E-02, 1.58E-03, 1.35E-02, 1.23E-04
	HL2	6.04E+00, 3.67E+01, 1.52E+02, 1.35E+02, 3.22E+04, 1.03E+03, 1.02E+03	1.92E+00, 2.86E+01, 1.23E+02, 1.10E+02, 3.29E+04, 9.53E+02, 9.34E+02
	HH2	1.75E+00, 2.99E+00, 5.22E+00, 5.40E+00, 2.94E+01, 9.40E+00, 8.29E-01	1.35E-01, 2.96E-01, 8.61E-01, 1.29E+00, 1.20E+01, 2.71E+00, 7.59E-01
	LH2	4.34E+00, 1.94E+01, 7.57E+01, 6.02E+01, 6.54E+03, 3.25E+02, -4.27E+02	1.13E+00, 1.48E+01, 6.21E+01, 4.64E+01, 6.94E+03, 3.07E+02, 5.00E+02
	LL	2.24E+00, 4.78E+00, 1.05E+01, 1.15E+01, 1.26E+02, 2.51E+01, 9.22E-02	5.70E-02, 2.56E-01, 8.54E-01, 8.76E-01, 1.97E+01, 2.59E+00, 7.85E-03
9	HL1	7.66E-01, 7.28E-01, 3.54E-01, 2.01E-01, 5.56E-02, 1.55E-01, 4.34E-02	2.34E-01, 3.25E-01, 1.93E-01, 1.15E-01, 6.29E02, 1.75E-01, 4.23E-02
	HH1	2.66E-01, 1.78E-01, 8.70E-02, 7.30E-02, 5.71E-03, 3.20E-02, -2.15E-03	2.34E-01, 2.00E-01, 1.08E-01, 6.90E-02, 8.86E03, 4.63E-02, 3.83E-03
	LH1	-9.80E-02, 1.49E-01, 7.26E-02, 6.34E-02, 8.99E-03, -3.04E-02, -2.43E-03	2.79E-01, 1.52E-01, 7.36E-02, 7.60E-02, 1.54E02, 5.30E-02, 3.60E-03
	HL2	6.58E+00, 9.20E+01, 4.45E+02, 2.98E+02, 2.54E+05, 4.12E+03, 3.22E+04	4.08E+00, 9.88E+01, 5.12E+02, 3.49E+02, 4.03E+05, 5.83E+03, 4.84E+04
	HH2	1.60E+00, 1.07E+01, 3.97E+01, 1.62E+01, - 7.64E+02, -2.05E+01, 1.46E+02	7.02E-01, 1.07E+01, 4.90E+01, 1.29E+01, 1.48E+03, 6.83E+01, 2.40E+02
	LH2	5.85E+00, 1.58E+02, 7.21E+02, 5.35E+02, 1.22E+06, 1.25E+04, -1.03E+05	8.56E+00, 2.55E+02, 1.16E+03, 8.85E+02, 2.12E+06, 2.17E+04, 1.77E+05
	LL	2.32E+00, 5.17E+00, 1.18E+01, 1.28E+01, 1.57E+02, 2.91E+01, 1.08E-02	4.15E-02, 1.96E-01, 6.87E-01, 6.97E-01, 1.77E+01, 2.15E+00, 1.07E-02
10	HL1	7.30E-01, 4.00E-01, 1.32E-01, 1.48E-01, 3.32E-02, 1.16E-01, 6.85E-03	2.27E-01, 3.16E-01, 1.04E-01, 1.24E-01, 3.59E02, 1.06E-01, 7.87E-03
	HH1	9.30E-02, 6.65E-02, 3.65E-02, 9.96E-03, 2.43E-04, 1.57E-03, 1.45E-04	1.07E-01, 4.84E-02, 2.18E-02, 1.02E-02, 4.01E04, 4.72E-03, 1.46E-04
	LH1	1.10E-01, 9.63E-02, 3.77E-02, 3.17E-02, 2.36E-03, 1.23E-02, -1.03E-03	2.18E-01, 1.20E-01, 3.80E-02, 3.93E-02, ,3.86E03, 2.44E-02, 1.62E-03
	HL2	-6.09E-01, 3.93E+01, 1.98E+02, 9.08E+01, 1.15E+04, -1.53E+02, 4.32E+03	3.59E+00, 1.29E+01, 7.00E+01, 2.80E+01, 5.57E+03, 5.98E+02, 3.33E+03
	HH2	2.25E+00, 5.29E+00, 1.28E+01, 1.22E+01, 1.60E+02, 2.85E+01, 5.43E+00	2.54E-01, 8.98E-01, 3.52E+00, 3.39E+00, 7.23E+01, 9.58E+00, 9.96E+00
	LH2	2.23E+00, 7.90E+00, 2.64E+01, 1.93E+01, 7.74E+02, 6.91E+01, -4.39E+01	1.17E+00, 5.66E+00, 2.05E+01, 1.82E+01, 7.75E+02, 6.82E+01, 6.64E+01
	LL	2.25E+00, 4.87E+00, 1.08E+01, 1.17E+01, 1.32E+02, 2.58E+01, 4.22E-02	4.69E-02, 1.95E-01, 6.33E-01, 7.20E-01, 1.59E+01, 2.10E+00, 3.93E-02

After the mean and standard deviation vectors are extracted for each person, each mean vector will be considered as a template vector for a specific person. As a last stage in the establishment of the hand palm identification system is the features analysis and evaluation stage. Various features combinations, up to six features, have been used to measure the similarity (matching) between the person's template features and the extracted features vectors.

The relative Euclidean distance measure had been used to determine the degree of similarity between the template vector and the features vector, the relative Euclidean measure could be expressed mathematically as follows [The03]:

$$d(t, f) = \sum_{i=1}^n \frac{t_i - f_i}{\sigma_i} \quad (4.1)$$

Where

d is the distance measure value.

t is the template feature vector.

f is the extracted feature vector for the unknown person.

n is the number of used features in the matching process.

The results of the conducted tests have indicated that the use of complex moments features lead to the identification rates much better than those obtained when using the invariant moment features. The ratio of success have reached 96% when using complex moment features, but when using the invariant moment features the rate of success haven't exceed 76%. Also it was found that a combination of six features leads to an acceptable recognition level. The increase to more than 6 features had not improved the matching results. Tables (4.3 - 4.7) shows the best results of identification rates for combinations of two, three up to six moments.

Table (4.3) Samples of the identification test results for combinations of two complex features or invariant moment features

Complex Moments		Invariant Moments	
Combination features (First, Second)	Success Rate	Combination features (First , Second)	Success Rate
17 , 34	60%	1 , 7	48%
9 , 35	56%	9 , 35	48%
34 , 35	56%	1 , 9	46%
9 , 11	50%	9 , 33	46%
9 , 31	50%	1 , 10	44%
17 , 25	50%	1 , 21	44%
3 , 35	48%	9 , 32	44%
4 , 34	48%	10 , 32	44%
9 , 24	48%	1 , 8	42%
17 , 20	48%	3 , 9	42%
18 , 34	48%	3 , 19	42%
11 , 18	46%	10 , 31	42%
17 , 18	46%	10 , 33	42%
17 , 23	46%	4 , 25	40%
17 , 33	46%	7 , 24	40%
18 , 35	46%	9 , 34	40%
20 , 34	46%	10 , 34	40%
20 , 35	46%	10 , 35	40%
31 , 34	46%	20 , 25	40%
9 , 26	44%	21 , 35	40%
10 , 17	44%	2 , 17	38%
11 , 17	44%	4 , 8	38%
16 , 34	44%	4 , 16	38%
17 , 19	44%	4 , 17	38%
17 , 21	44%	5 , 8	38%
17 , 27	44%	5 , 10	38%
17 , 35	44%	7 , 32	38%
18 , 21	44%	7 , 33	38%
18 , 31	44%	7 , 34	38%
20 , 24	44%	7 , 35	38%
20 , 32	44%	8 , 33	38%
21 , 35	44%	9 , 19	38%
25 , 32	44%	9 , 31	38%
4 , 17	42%	1 , 24	36%
7 , 18	42%	2 , 19	36%
10 , 30	42%	3 , 8	36%
11 , 27	42%	3 , 10	36%
16 , 20	42%	3 , 16	36%
17 , 22	42%	3 , 21	36%
17 , 28	42%	4 , 20	36%

Table (4.4) Samples of the identification test results for combinations of three complex features or invariant moment features

Complex Moments		Invariant Moments	
Combination features (First, Second ,Third)	Success Rate	Combination features (First, Second ,Third)	Success Rate
17 , 20 , 34	76%	1 , 9 , 16	68%
17 , 24 , 34	76%	3 , 12 , 16	64%
9 , 11 , 19	72%	1 , 10 , 16	60%
9 , 28 , 35	72%	2 , 10 , 17	60%
17 , 19 , 34	72%	2 , 10 , 20	60%
8 , 17 , 34	70%	10 , 18 , 32	60%
9 , 10 , 35	70%	10 , 18 , 34	60%
9 , 20 , 35	70%	10 , 18 , 35	60%
17 , 32 , 34	70%	10 , 26 , 35	60%
8 , 9 , 31	68%	1 , 9 , 18	58%
9 , 11 , 18	68%	1 , 9 , 22	58%
9 , 11 , 24	68%	1 , 10 , 21	58%
9 , 11 , 33	68%	1 , 10 , 31	58%
9 , 11 , 35	68%	1 , 12 , 21	58%
11 , 18 , 32	68%	2 , 10 , 18	58%
17 , 25 , 32	68%	2 , 10 , 19	58%
17 , 26 , 34	68%	3 , 10 , 17	58%
17 , 27 , 34	68%	9 , 17 , 33	58%
17 , 28 , 34	68%	9 , 17 , 35	58%
17 , 31 , 34	68%	9 , 19 , 35	58%
18 , 34 , 35	68%	9 , 20 , 33	58%
20 , 31 , 33	68%	10 , 18 , 31	58%
20 , 34 , 35	68%	10 , 19 , 31	58%
1 , 11 , 35	66%	10 , 19 , 33	58%
1 , 21 , 35	66%	1 , 9 , 17	56%
3 , 18 , 34	66%	1 , 10 , 17	56%
6 , 10 , 34	66%	1 , 10 , 18	56%
6 , 32 , 34	66%	3 , 7 , 16	56%
9 , 10 , 31	66%	3 , 10 , 19	56%
9 , 13 , 35	66%	4 , 9 , 17	56%
9 , 20 , 28	66%	4 , 9 , 19	56%
9 , 31 , 33	66%	4 , 10 , 17	56%
9 , 32 , 35	66%	5 , 10 , 17	56%
10 , 17 , 34	66%	9 , 17 , 31	56%
10 , 18 , 34	66%	9 , 17 , 32	56%
10 , 20 , 34	66%	9 , 17 , 34	56%
11 , 17 , 34	66%	9 , 19 , 31	56%
17 , 18 , 34	66%	9 , 19 , 33	56%
17 , 19 , 25	66%	9 , 20 , 35	56%

Chapter Four: Tests and Results

Table (4.5) Samples of the identification test results for combinations of four complex features or invariant moment features

Complex Moments		Invariant Moments	
Combination features (First, Second, Third, Fourth)	Success Rate	Combination features (First, Second, Third, Fourth)	Success Rate
9 , 13 , 24 , 29	82%	1 , 9 , 12 , 16	72%
17 , 19 , 27 , 34	82%	1 , 9 , 13 , 16	72%
3 , 11 , 18 , 32	80%	1 , 9 , 16 , 22	72%
9 , 11 , 13 , 24	80%	1 , 9 , 16 , 25	70%
9 , 13 , 28 , 35	80%	1 , 9 , 16 , 28	70%
9 , 16 , 34 , 35	80%	1 , 3 , 9 , 16	68%
9 , 20 , 31 , 33	80%	1 , 4 , 9 , 16	68%
9 , 28 , 32 , 35	80%	1 , 7 , 9 , 16	68%
17 , 18 , 26 , 34	80%	1 , 9 , 12 , 17	68%
17 , 20 , 32 , 34	80%	1 , 9 , 14 , 16	68%
17 , 22 , 26 , 34	80%	1 , 9 , 16 , 18	68%
17 , 25 , 31 , 34	80%	1 , 9 , 16 , 20	68%
18 , 31 , 32 , 33	80%	1 , 9 , 16 , 27	68%
1 , 10 , 20 , 21	78%	1 , 10 , 16 , 28	68%
1 , 11 , 30 , 35	78%	2 , 3 , 10 , 18	68%
2 , 4 , 34 , 35	78%	4 , 9 , 12 , 16	68%
2 , 7 , 20 , 35	78%	4 , 9 , 16 , 25	68%
2 , 11 , 18 , 32	78%	1 , 6 , 10 , 16	66%
2 , 17 , 20 , 34	78%	1 , 9 , 10 , 16	66%
2 , 20 , 34 , 35	78%	1 , 9 , 16 , 17	66%
3 , 4 , 32 , 34	78%	1 , 9 , 16 , 29	66%
3 , 20 , 34 , 35	78%	1 , 9 , 16 , 30	66%
5 , 20 , 34 , 35	78%	1 , 9 , 16 , 31	66%
6 , 8 , 27 , 34	78%	1 , 9 , 16 , 35	66%
7 , 11 , 18 , 32	78%	1 , 10 , 16 , 26	66%
9 , 13 , 19 , 35	78%	2 , 9 , 12 , 16	66%
9 , 13 , 24 , 34	78%	3 , 4 , 10 , 18	66%
9 , 21 , 24 , 32	78%	3 , 10 , 16 , 26	66%
9 , 25 , 28 , 32	78%	4 , 9 , 17 , 32	66%
10 , 20 , 31 , 33	78%	4 , 9 , 18 , 33	66%
17 , 18 , 19 , 34	78%	4 , 9 , 19 , 33	66%
17 , 19 , 25 , 34	78%	5 , 9 , 12 , 16	66%
17 , 20 , 24 , 34	78%	1 , 2 , 9 , 16	64%
17 , 20 , 28 , 34	78%	1 , 3 , 9 , 17	64%
17 , 20 , 31 , 34	78%	1 , 3 , 10 , 17	64%
17 , 25 , 26 , 34	78%	1 , 5 , 9 , 16	64%
17 , 26 , 27 , 34	78%	1 , 7 , 10 , 16	64%
17 , 28 , 31 , 34	78%	1 , 8 , 9 , 16	64%
18 , 24 , 32 , 34	78%	1 , 9 , 11 , 16	64%

Table (4.6) Samples of the identification test results for combinations of five complex features or invariant moment features

Complex Moments		Invariant Moment	
Combination features (First, Second, Third, Fourth, Fifth)	Success Rate	Combination features (First, Second, Third, Fourth, Fifth)	Success Rate
9, 13, 24, 29, 33	90%	1, 3, 10, 16, 28	74%
3, 6, 19, 27, 34	88%	1, 4, 9, 12, 16	74%
9, 10, 13, 24, 27	88%	2, 10, 16, 26, 35	74%
2, 20, 32, 34, 35	86%	1, 2, 9, 12, 17	72%
3, 6, 17, 31, 34	86%	1, 2, 9, 14, 16	72%
3, 6, 27, 32, 34	86%	1, 2, 9, 16, 28	72%
7, 11, 18, 32, 34	86%	1, 3, 9, 16, 27	72%
7, 16, 18, 31, 34	86%	1, 3, 9, 16, 28	72%
8, 20, 26, 28, 34	86%	1, 4, 9, 13, 16	72%
9, 10, 13, 18, 27	86%	1, 4, 9, 14, 16	72%
9, 10, 13, 20, 24	86%	1, 5, 9, 12, 16	72%
9, 10, 13, 24, 29	86%	1, 5, 9, 13, 16	72%
9, 10, 13, 31, 33	86%	1, 5, 9, 14, 16	72%
9, 10, 24, 31, 32	86%	1, 9, 14, 16, 35	72%
9, 11, 13, 24, 32	86%	1, 9, 16, 22, 27	72%
9, 11, 19, 27, 35	86%	2, 10, 16, 26, 31	72%
9, 12, 13, 29, 35	86%	2, 10, 16, 26, 33	72%
9, 13, 19, 24, 35	86%	4, 9, 12, 16, 17	72%
9, 13, 21, 24, 35	86%	4, 9, 12, 16, 19	72%
9, 13, 24, 29, 32	86%	4, 9, 12, 16, 20	72%
9, 13, 24, 29, 34	86%	4, 9, 12, 16, 26	72%
9, 13, 28, 29, 33	86%	4, 10, 16, 26, 32	72%
9, 20, 24, 31, 32	86%	5, 10, 16, 26, 32	72%
9, 24, 28, 29, 31	86%	5, 10, 16, 26, 33	72%
17, 22, 24, 32, 34	86%	5, 10, 16, 26, 35	72%
1, 5, 27, 34, 35	84%	1, 2, 9, 12, 16	70%
2, 4, 20, 32, 34	84%	1, 2, 9, 13, 16	70%
2, 5, 20, 34, 35	84%	1, 2, 9, 16, 18	70%
2, 17, 20, 32, 34	84%	1, 2, 9, 16, 22	70%
2, 20, 28, 34, 35	84%	1, 2, 9, 16, 25	70%
2, 20, 31, 32, 33	84%	1, 2, 10, 16, 26	70%
3, 4, 20, 32, 34	84%	1, 3, 9, 13, 16	70%
3, 6, 18, 34, 35	84%	1, 3, 9, 15, 16	70%
3, 6, 31, 32, 33	84%	1, 3, 9, 16, 18	70%
3, 20, 24, 31, 32	84%	1, 3, 9, 16, 22	70%
3, 20, 31, 32, 34	84%	1, 3, 9, 16, 25	70%
3, 20, 32, 34, 35	84%	1, 4, 7, 10, 16	70%
7, 11, 18, 32, 33	84%	1, 4, 9, 16, 18	70%
7, 17, 20, 32, 34	84%	1, 4, 10, 16, 26	70%
9, 10, 11, 15, 31	84%	1, 5, 8, 10, 16	70%

Table (4.7) Samples of the identification test results for combinations of six complex features or invariant moment features

Complex Moments		Invariant Moment	
Combination features (First, Second, Third, Fourth, Fifth, sixth)	Success Rate	Combination features (First, Second, Third, Fourth, Fifth, sixth)	Success Rate
9, 10, 20, 24, 31, 32	96%	1, 2, 7, 10, 16, 26	76%
3, 6, 10, 15, 27, 31	92%	1, 2, 9, 14, 16, 26	76%
3, 6, 10, 27, 31, 34	92%	4, 7, 10, 12, 16, 35	76%
3, 6, 15, 17, 31, 34	92%	4, 9, 12, 16, 25, 35	76%
3, 6, 19, 27, 31, 34	92%	4, 10, 12, 16, 25, 34	76%
3, 10, 15, 18, 27, 31	92%	4, 10, 12, 16, 25, 35	76%
3, 20, 24, 31, 32, 34	92%	1, 2, 6, 9, 12, 16	74%
9, 10, 19, 24, 27, 32	92%	1, 2, 9, 12, 16, 26	74%
9, 13, 19, 24, 29, 33	92%	1, 2, 9, 14, 16, 28	74%
9, 13, 20, 24, 31, 32	92%	1, 2, 9, 14, 16, 29	74%
9, 13, 24, 25, 34, 35	92%	1, 2, 9, 15, 16, 28	74%
9, 13, 24, 28, 29, 31	92%	1, 3, 4, 10, 16, 26	74%
9, 13, 24, 28, 29, 33	92%	1, 3, 7, 9, 16, 26	74%
9, 13, 24, 29, 31, 33	92%	1, 3, 7, 10, 16, 26	74%
9, 20, 24, 28, 29, 31	92%	1, 3, 7, 10, 16, 28	74%
1, 6, 18, 28, 34, 35	90%	1, 3, 8, 9, 16, 28	74%
3, 6, 10, 19, 27, 34	90%	1, 3, 9, 13, 16, 20	74%
3, 6, 14, 17, 31, 34	90%	1, 3, 9, 13, 16, 28	74%
3, 6, 17, 27, 31, 34	90%	1, 3, 9, 14, 16, 26	74%
3, 6, 28, 31, 34, 35	90%	1, 3, 9, 14, 16, 28	74%
3, 10, 18, 27, 31, 33	90%	1, 3, 9, 15, 16, 28	74%
3, 20, 28, 31, 32, 34	90%	1, 3, 10, 15, 16, 28	74%
9, 10, 13, 19, 24, 27	90%	1, 4, 7, 9, 12, 16	74%
9, 10, 13, 20, 24, 30	90%	1, 4, 9, 12, 16, 28	74%
9, 10, 13, 24, 27, 34	90%	1, 4, 9, 12, 16, 30	74%
9, 10, 13, 24, 29, 30	90%	1, 5, 9, 14, 16, 26	74%
9, 10, 19, 24, 32, 34	90%	1, 5, 9, 15, 16, 25	74%
9, 10, 20, 24, 32, 34	90%	1, 9, 12, 16, 27, 35	74%
9, 12, 14, 21, 24, 34	90%	1, 10, 12, 16, 27, 35	74%
9, 12, 24, 28, 29, 31	90%	2, 7, 10, 16, 26, 31	74%
9, 13, 19, 24, 27, 34	90%	2, 7, 10, 16, 26, 32	74%
9, 13, 19, 27, 34, 35	90%	2, 10, 16, 17, 26, 34	74%
9, 13, 20, 24, 28, 31	90%	2, 10, 16, 20, 26, 31	74%
9, 13, 20, 24, 32, 34	90%	2, 10, 16, 20, 26, 34	74%
9, 13, 21, 24, 29, 33	90%	2, 10, 16, 25, 26, 31	74%
9, 13, 23, 24, 29, 33	90%	2, 10, 16, 26, 29, 35	74%
9, 13, 24, 31, 32, 34	90%	2, 10, 16, 26, 30, 35	74%
9, 14, 19, 24, 32, 34	90%	4, 5, 9, 12, 16, 20	74%
9, 14, 24, 28, 31, 34	90%	4, 6, 9, 12, 16, 17	74%
9, 14, 24, 30, 31, 34	90%	4, 6, 9, 12, 16, 18	74%

4.3.4 Identification Rates for one level Haar Wavelet Transform

Before implementing two levels Haar wavelet transform to decompose the image into sub-bands, whose contents used as a media to get the features, the one level Haar wavelet transform have been used. The identification rates for the case of one level Haar wavelet have not reached to a satisfactory rate as shown in table (4.8). For this reason the two levels Haar wavelet transform have been adopted. The results of identification rates when using two levels of Haar wavelet transform are much better than when using one level, as shown in the previous sections. Table (4.8) shows the results of identifications rates when one level of Haar wavelet have been used, where the combinations of 2, 3, 4, 5, and 6 features for (both complex and invariant moments features) were utilized.

Table (4.8) The results of identification rates for both complex and invariant moments features

Number of combinations	Complex moments		Invariant moments	
	Combinations features	Identification rates	Combinations features	Identification rates
2 Features	9 , 13	52 %	12 , 18	58 %
3 Features	1, 6, 20	72 %	3, 12, 20	66 %
4 Features	1, 6, 7, 20	82 %	3, 11, 13, 19	78 %
5 Features	1, 5, 6, 7, 20	84 %	1, 3, 11, 13, 19	80 %
6 Features	6, 9, 13, 17, 18, 19	88 %	1, 3, 4, 11, 13, 19	78 %

Chapter Five

Chapter Five

Conclusions and Future Work

5.1 Conclusions

According to the tests results presented in chapter four, the following conclusions have been derived:

1. The obtained identification rates in the test indicate that the use of complex moment as discriminating feature is much better than the use of the invariant moment.
2. When using complex moment features, the features which gave the best identification rate (are: feature 34, feature 31, feature 33 and feature 35). The sub band which contain these features and gave the best identification rate is LL sub band, and the best order of the moments for these features are (fourth, first, third and fifth) respectively.
3. When using invariant moment features, the features which gave the best identification rate (are: feature 16, feature 1, feature 4, and feature 2). The sub band which contain these features and gave the best identification rate is HL sub band, and the best order of the moments for these features are (first, first, fourth and second) respectively.
4. In matching stage, the identification rates when using combinations of six features are much better than identification rates when using combinations of five, four, three or two features.
5. When using two levels of Haar wavelet transform the identification rates which are determined in the matching process is much better than the identification rate of using one level of Haar wavelet transform.

5.2 Future Work

1. One can use another kind of wavelet transform such as Daubechies wavelet transform, tab 9/7 wavelet transform, packet wavelet.
2. Using another kind of moment features such as Zernike moment features.
3. Using two kinds of biometric features (such as hand palm with fingerprint) to increase the accuracy of the identification system.
4. Using more than 2 levels of Haar wavelet and study how this will affect the identification accuracy of the established system.

References

References

- [Ale02] Alexandra L. N. W., Pengcheng, S., “Peg-Free Hand Geometry Recognition Using Hierarchical Geometry and Shape Matching”, University of Science and Technology, pp. 281-284, 2002.
www.ee.ust.hk/~eeship/Papers/MVA02.pdf
- [Bur98] Burrus, C. S., Gopinath, R. A., Guo, H., ”Introduction to Wavelets and Wavelet Transforms”, Simone and Schuster/ A Viacom Company, United State in America, ISBN 0-13-489600-9,1998.
- [Cun90] Cunningham, S.,”The Ubiquitous Bresenham Algorithm as a Basis for Graphics Interpolation Processes”, 1990.
www.cs.csustan.edu/~rsc/SDSU/Interpolation.pdf
- [Due04] Due, M. A., Mikkelsen, J. R., Molbo, J., Nielsen, A. H. , Rohde, M. L., Stobbe, K., "Sorting of Fasteners", pp. 1-133, Report in 2004.
- [Eki03] Ekinci, M., Nabiyev, V. V., Öztürk, Y., “A Biometric personal Verification Using Palm Print Structural Features and Classifications”, IJCI Proceedings of Intl. XII. Turkish Symposium on Artificial Intelligence and Neural Networks, vol.1, no .1, 2003.
- [Gat04] Gatos, B., Pratikakis, I., Perantonis S. J., "An Adaptive linearization Technique for Low Quality Historical Documents", Computational Intelligence Laboratory, pp.102–113,2004.
<http://www.iit.demokritos.gr/cil>
- [Geo05] George, L. A., Al-Abudi, B. K., "Color Image Compression using Wavelet Transform", GVIP 05 Conference, pp. 1-7, 2005.
- [Gon87] Gonzalez, R. C., “Digital Image Processing”, 2, Wesley Publishing Company, Addison, 1987.

- [Gon74] Gonzalez, R. C., Tou, T. J., “Pattern Recognition Principles”, Reading Mass., Wesley Publishing Company, Addison, 1974.
- [Hug04] Hughes, N. P., Roberts, S. J., Tarassenko, L., “Semi-Supervised Learning of Probabilistic Models for ECG Segmentation”, Conf Proc IEEE Eng Med Biol. Soc., vol. 1, 2004.
- [Jai99] Jain, A. K., Ross, A., Pankanti, S., "A Prototype Hand Geometry-based Verification System", 2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication (AVBPA), pp.166-171, 1999.
- [Kar00] Karuppuswamy, J.,”Detection and Avoidance of Simulated Potholes in Autonomous Vehicle Navigation in an Unstructured Environment”, M.Sc. thesis, University of Cincinnati, 2000.
- [Kon03] Kong, W. K., Zhang, D. , Li, W.,” Palm print feature extraction using 2-D Gabor filters”, Pattern Recognition , vol. 36, pp 2339 – 2347,2003.
- [Mic04] Michael S., Arigovindan, M., Hunziker, P., Unser, M.,”Multiresolution Moment Filters: Theory and Applications”, IEEE Transaction on Image Processing, vol. 13, no. 4, 2004.
- [Moh05] Mohammed, A. M.,”Hand Identification Using Fuzzy-Neural”, Computer Science, M.Sc. Thesis, Al-Nahrain University, 2005.
- [Phi94] Phillips, D., “Image Processing in C”, 2, CMP Media, Kansas, 1994.
- [Pra78] Pratt, W.K, “Digital Image Processing”, 3, John Wiley and Sons. , New York, 1978.
- [Rho04] Rhodes, K. A., "Aviation Security Challenges in Using Biometric Technologies", United States General Accounting Office, pp. 4-25, 2004.

www.gao.gov/new.items/d04785t.pdf

- [Rib02] Ribari, S., Ribari, D., Paveši, N., "A multimodal biometric-based user-identification system for internet", Proceedings of the 10th International Conference on Software, Telecommunication and Computer Networks SoftCom, Published by FESB, pp. 143-146, 2002.
- [Rih01] Riha, Z., "Biometric Authentication Systems", Report, Masaryk University, pp. 22 - 24. 10, 2001.
<http://www.ecom-monitor.com/papers/biometricsTR2000.pdf>
- [Roj06] Rojals, J. S., "Optimization and Generalization of Lifting Schemes: Application to Lossless Image Compression", PhD thesis, University Polit`ecnica de Catalunya, April 2006.
- [Roy98] Roy, V., Jawahar, C. V. "Feature Selection for Hand-Geometry based Person Authentication", html, 1998.
www.citeseer.ist.psu.edu/761998.html
- [Sac96] Sachs J., "Digital Image Basics", Technical Report, Digital light and color , Cambridge Press, pp. 1-14, 1996.
www.dl-c.com/basics.pdf
- [Tan95] Tanguay, D. O., "Hidden Markov Models for Gesture Recognition", M.Sc. thesis, Massachusetts Institute of Technology, 1995.
- [Tan98] Tangi, X., Stewart, W. K., Vincent, L., Huang, H., Marra, M., Gallager S. M., Davis, C. S., "Automatic Plankton Image Recognition", Kluwer Academic Publishers. Netherlands, pp 177–199, 1998.
- [The03] Theodoridis, S., Koutroumbas, K., "Pattern Recognition", 2, Elsevier, USA, 2003.
- [Tza01] Tzanetakis, G., Essi, G., Cook, P., "Audio Analysis using the Discrete Wavelet Transform", WSES Int. Conf. Acoustics and Music: Theory and Applications (AMTA 2001) Skiathos, Greece, 2001.

- [Val99] Valad, F. M., “A Drawing Interpretation Tool for the Sight-impaired”, M.Sc. thesis, Virginia Polytechnic Institute, 1999.
 - [Wan05] Wang, K., B.Eng., M. Eng., “Region-Based Three-Dimensional Wavelet Transform Coding”, M.Sc. thesis, Carleton University, 2005.
 - [Woo03] Woodward, J. D., Riley, J., Horn, C., Gatune, J., Thomas, A.,” Biometrics A Look at Facial Recognition”, Rand Public Safety and Justice for the Virginia State Crime Commission, Santa Monica, 2003.
 - [Xia01] Xiao P.,” Image Compression by Wavelet Transform”, M.Sc. Thesis, Tennessee State University, 2001.
 - [You02] You, J., Li, W., Zhang, D., “Hierarchical Palm Print Identification via Multiple Feature Extraction”, Pattern Recognition, vol. 35, pp. 847-859, 2002.
 - [Zha98] Zhang, D., Shu, W.,”Two novel characteristics in palm print verification: datum point invariance and line feature matching”, Pattern Recognition, vol. 32, pp 691–702, 1998.
 - [Zha03] Zhang, D., Kong, W.K., You, J. and Wong, M, " Online Palm print Identification, IEEE transactions on pattern analysis and machine intelligence, vol. 25, No. 9, 2003.
 - [Zha04] Zhang, L., Zhang, D., “Characterization of Palm Print by Wavelet Signatures via Directional Context Modeling”, IEEE, vol. 34, No. 3, 2004.
 - [Zhe05] Zheng, L., He, X., “Classification Techniques in Pattern Recognition”, Conference proceedings , 2005.
- www.hpl.hp.com/hpjournal/dtj/vol7num4/vol7num4art5.pdf 1

الخلاصة

ان نظم التمييز تعتبر من احسن طرق المستخدمة و خصوصا في مجالات الحماية, وهناك عدة طرق للتمييز منها طرق تمييز اليد او عن طريق تمييز وجه الانسان, وغيرها. ان النظام المقترح هو نظام تمييز راحة اليد, هذا النظام يتضمن مرحلتين, المرحلة الاولى هي مرحلة التحضيرات والمرحلة الثانية هي مرحلة المطابقة. في مرحلة التحضيرات يتم اخذ نماذج من صور اليد اليمنى لعدد من الاشخاص. في كلا المرحلتين (مرحلة التحضيرات و المطابقة) يتم تحويل لون جميع الصور الملونة الى صور بالابيض و اسود باستخدام طريقة حد العتبة. وباستخدام عامل لابلاس يتم ايجاد حدود اليد. من اجل الحصول على منطقة راحة اليد يتم تحديد ترتيب نقاط حدود اليد وذلك باستخدام خوارزمية ترميز السلسلة, ثم نصل بعد ذلك الى اخر خطوة في مرحلة التحضيرات حيث يتم فصل الاصابع عن اليد وذلك عن طريق تحديد نقاط ارتباط الاصابع مع راحة اليد. بعد عملية فصل الاصابع يتم تطبيق خوارزمية برزنهايم من اجل ربط نقاط الارتباط للحصول على مخطط منطقة راحة اليد. يتم تحويل لون راحة اليد الى لون الابيض وذلك باستخدام خوارزمية حشو البذرة. من اجل الحصول على راحة اليد ملونة تم استبدال لون كل نقطة لونها ابيض بلون الموقع المقابل له في الصورة الملونة. بعد ذلك تم تطبيق تحويل هار على الصور من اجل الحصول على خصائص الصورة, و من اجل تحديد مميزات خصائص الصور تم استخدام نوعين من الخصائص المميزة وهي خصائص العزوم الثابتة والعزوم المعقدة حيث تخزن جميع خصائص الصور في قاعدة البيانات. تم حساب المتوسط والانحراف المعياري لهذه الصفات التي سوف تستخدم لاحقا في عملية المطابقة. وفي المرحلة الاخيرة وهي مرحلة المطابقة تم الحصول على اعلى نسبة للتطابق وذلك بقياس درجة التشابه باستخدام المقياس الاقليدي بين صفات خصائص اليد و بين قيم الخصائص المخزونة في قاعدة البيانات. تشير النتائج الى ان نسب التعرف باستخدام الخصائص المعقدة هي افضل من النسب التعريف باستخدام الخصائص الثابتة, حيث ان اعلى نسبة للتعرف تم الحصول عليها باستخدام مستوى من تحويل هار على الصورة بالنسبة للخصائص المعقدة هي ٨٨% اما بالنسبة للخصائص الثابتة فكانت النسبة ٧٨% , بينما باستخدام مستويان من تحويل هار على الصورة فقد كانت النسب بالنسبة لاستخدام الخصائص المعقدة هي ٩٦% اما للخصائص الثابتة هي ٧٦%.



جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

تميز راحة اليد باستخدام التحويل المويجي

رسالة

مقدمة الى كلية العلوم في جامعة النهرين كجزء من
متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

انس سنان مكي

(بكالوريوس جامعة النهرين ٢٠٠٤)

أشرف

د. لؤي أدور جورج