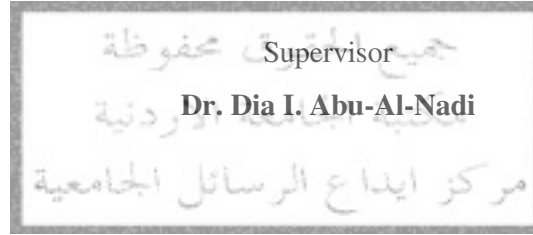


**AUTOMATED PERSONAL RECOGNITION SYSTEM BASED ON  
HUMAN IRIS ANALYSIS**

By  
**Taleb M. Al-**



**This Thesis was submitted in Partial Fulfillment of the Requirements for the  
Electrical Engineering/ Communications**

**Faculty of Graduate Studies  
The University of Jordan**

**January, 2005**

This Thesis (Automated Personal Recognition System Based on Human Iris Analysis) was successfully defended and approved on .....12/1/2005.....

Examination Committee

Signature

Dr. Dia Abu-Al-Nadi / Supervisor  
Communications

*Dia Abunnad*

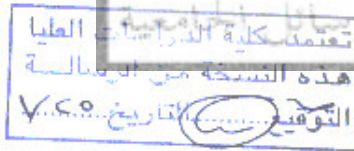
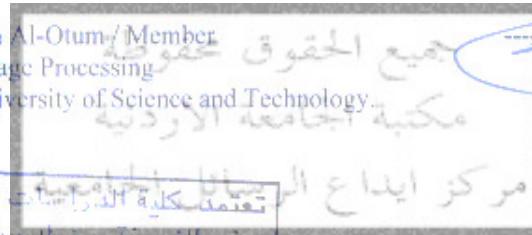
Prof. Jamil Ayoub / Member  
Communications Networks

*Jamil Ayoub*

Dr. Loay D. Khalaf / Member  
Communications

*Loay*

Dr. Hazem Al-Otum / Member  
Digital Image Processing  
Jordan University of Science and Technology



٢٠٠٥ /

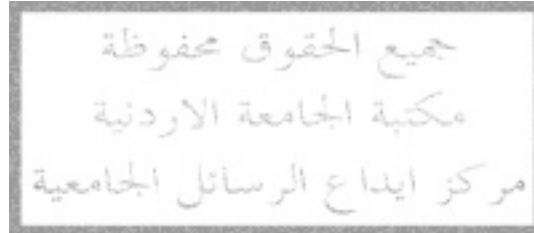
## DEDICATION

To my parents who dedicate their lives to secure a happy

To my truehearted teachers and instructors

To my valued classmates and friends

I present this humble work

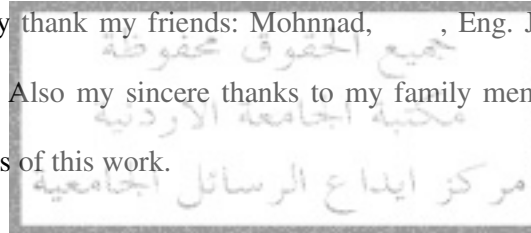


## ACKNOWLEDGMENT

I am greatly indebted to my instructor; Dr. Dia I. Abu-Al-Nadi, for his support and help. Also I want to thank the examination committee.

I would like to thank Libor Masek and Peter Kovesi for their useful published MATLAB Source Codes that are very useful for iris recognition researchers. Also I want to thank Chinese Academy of Sciences for their published CASIA iris database which makes this work easy.

I specially thank my friends: Mohnnad, Eng. Jarah, and Zaid for helping in various ways. Also my sincere thanks to my family members whose support is the key of the success of this work.



## TABLE OF CONTENTS

Committee Decision	ii
Dedication	iii
Acknowledgment	iv
Table of Contents	v
List of Tables	vii
List of Figures	viii
List of Abbreviations	xi
List of Appendices	xii
Abstract	xiii
<b>Introduction</b>	<b>1</b>
1. The Human Iris	3
1.1 Anatomy of the Human Iris	3
1.2 Medical Facts about the Human Iris	5
1.3 Reasons for Studying the Human Iris	7
2. Iris Recognition Technology	8
3. Literature Review of Iris Recognition Technology	12
4. The Objective of This Study	15
<b>Image Acquisition and Iris liveness Detection</b>	<b>17</b>
1. Introduction	17
2. Iris Image Acquisition Techniques	17
Image Acquisition System	18
2.2 Wildes <i>et al.</i> Iris Image Acquisition System	20
Iris Image Acquisition system	21
2.4 Lim <i>et al.</i> Iris Image Acquisition System	22
3. Iris Liveness Detection Techniques	23
<b>Iris Segmentation and Iris Image Quality Assessment</b>	<b>25</b>
1. Introduction	25
2. Iris Segmentation Techniques	25
2.1 D -Differential Operator	26
2.2 Hough Transform	27
2.3 Eyelashes and Noise Detection	28
3. Iris Image Quality Assessment Techniques	29
4. The Proposed Approach for Iris Segmentation and Iris Image Quality Assessment	30
<b>Iris Normalization</b>	<b>48</b>
1. Introduction	48
2. Iris Normalization Techniques	49
Rubber Sheet Model	49

2.2 Image Registration	50
2.3 Virtual Concentric Circles	52
3. Efficient Rubber Sheet Model	52
<b>Feature Extraction, Encoding, And Matching</b>	<b>57</b>
1. Introduction	57
2. Feature Extraction and Encoding techniques	57
2.1 Gabor Filters	58
2.2 Log-Gabor Filters	60
2.3 Laplacian of Gaussian Filters	62
2.4 Circular Symmetric Filters	63
2.5 Haar Wavelet	64
2.6 Zero-crossing of the Wavelet Transform	65
2.7 Multichannel Gabor Filters and Wavelet Transform	65
3. Matching Techniques	66
3.1 Hamming Distance	66
3.2 Normalized Correlation	66
3.3 Weighted Euclidean Distance	67
4. The Proposed Approach for Feature Extraction, Encoding and Matching	68
<b>Experimental Results and Conclusions</b>	<b>75</b>
1. Introduction	75
2. Software and Hardware Requirements	75
3. CASIA Iris Database	76
4. Iris Segmentation Results	78
5. Iris Image Quality Assessment Results	80
6. Pupil Oscillations Analysis	82
7. Iris Normalization Results	85
8. Feature Extraction, Encoding, and Matching Results	86
8.1 Algorithm Performance	86
8.2 Algorithm Parameters	92
9. Comparing with Existing Algorithms	95
10. Summary and Conclusions	101
11. Suggestions for Future Work	103
<b>References</b>	<b>105</b>
<b>Appendices</b>	<b>108</b>
<b>Abstract in Arabic</b>	<b>145</b>

## LIST OF TABLES

Table Number	Table Name	Page
Table 1	Different Compass Gradients	38
Table 2	Compass Gradients for a Different Direction	38
Table 3	The Relation Between the Noise Threshold and the Number of Rejected Images	81
Table 4	False Reject Rate (FRR) and False Accept Rate (FAR) for Different Threshold Points.	91
Table 5	Decidability Values for Various Average Block Sizes	95
Table 6	The Results of Comparing the proposed Method for Iris Segmentation with that of Masek	97
Table 7	Comparison of Identification Rates for Different Methods	99

## LIST OF FIGURES

Figure Number	Figure Name	Page
Figure 1	CASIA database. (b) Human eye.	3
Figure 2	Section through the human iris.	4
Figure 3	Anterior surface of the human iris.	4
Figure 4	Structure of the human iris in relation to the human organs according to the iridology.	8
Figure 5	Schematic diagram of entire iris recognition system.	9
Figure 6	Secure access to bank cash machine accounts using iris recognition technology.	11
Figure 7	The flow chart of the proposed iris recognition algorithm	16
Figure 8	acquisition system.	19
Figure 9	A schematic diagram of the Wildes <i>et al.</i> iris image acquisition system.	20
Figure 10	A perspective view of the automatic teller machine	22
Figure 11	A schematic diagram of Lim <i>et al.</i> iris image acquisition system.	23
Figure 12	An example to show the accuracy in locating the inner	31
Figure 13	The results of finding an approximate pupil region for image	32
Figure 14	The results of pupil boundary localization step in the iris segmentation process for image	33
Figure 15	.	34
Figure 16	Edge detection via gradient operators.	34
Figure 17	The results of outer boundary localization step in the iris from CASIA	37



	database	
Figure 18	One sample of failure images that can be detected by step one of iris image quality assessment	41
Figure 19	The results of eyelashes and eyelids isolation step in the iris segmentation process database	43
Figure 20	Illustration of calculation the noise percent of eye image	46
Figure 21	Samples of failure images that have not sufficient region for recognition	47
Figure 22		49
Figure 23	I sheet model for iris normalization	50
Figure 24	The iris mask for om CASIA database	53
Figure 25	The edge map of the iris mask	53
Figure 26	model	54
Figure 27	The result of mapping $(r, \theta)$ locations in the normalized iris region to $(x, y)$ locations in the origin iris region	55
Figure 28	The iris normalization result, with noise region intensities equal to the average of the free noise region	56
Figure 29	Noise mask that will be used in the feature extraction process	56
Figure 30	The even and odd symmetric components of the Gabor filter	59
Figure 31	Phase Quantization process used to encode iris patterns	60
Figure 32	Gabor and log-Gabor transfer functions viewed on both linear and logarithmic frequency scales	61
Figure 33	Three pairs(even and odd) of log Gabor wavelets all tuned to the same frequency, but having different bandwidths.	62
Figure 34	Conceptual diagram for organizing a feature vector using Haar wavelet transform	64

Figure 35	feature extraction and encoding	69
Figure 36	The region of interest used in the feature extraction process	71
Figure 37	Illustration of the matching process	74
Figure 38	Some samples of CASIA iris database	77
Figure 39	Segmentation results of various images from CASIA database.	79
Figure 40	The upper and lower sides of the iris region, with a noise region indicated by black color	80
Figure 41	Receiver Operating Characteristic (ROC) curves for a different noise thresholds.	81
Figure 42	The changes in the pupil radius and iris radius for the same eye image.	83
Figure 43	Displacements between the pupil center and the iris center for the same eye image	83
Figure 44	Pupil oscillations analysis for all images in the CASIA iris database	85
Figure 45	Nor database	86
Figure 46	Illustration of False Accept Rate and False Reject Rate for the distributions of intra-class and inter-class	88
Figure 47	Distributions of intra-class and interclass for the proposed algorithm.	90
Figure 48	The ROC curve for the proposed algorithm	92
Figure 49	The Receiver Operating Characteristic (ROC) curves for the angular direction test and the radial direction test.	94
Figure 50	ROC curves for different methods in the verification mode.	100
Figure 51	ROC curve for the proposed method in the verification mode.	100
Figure 52	Software architecture	110

## LIST OF ABBREVIATIONS OR SYMBOLS

PIN: Personal Identification Number.

ATM: Automated Teller Machine.

LCD: Liquid Crystal Display.

CCD: Charge Coupled Devices.

1D: One Dimension.

2D: Two Dimensional.

3D: Three Dimensional.

LVQ: Learning Vector Quantization.

AAD: Average Absolute Deviation.

HD: Hamming Distance.

WED: Weighted Euclidean Distance.

NaN: IEEE arithmetic representation for Not-a-Number.

FAR: False Accept rate.

FRR: False Reject Rate.

MR: Matching Ratio.

RAD: Rapid Application Development.

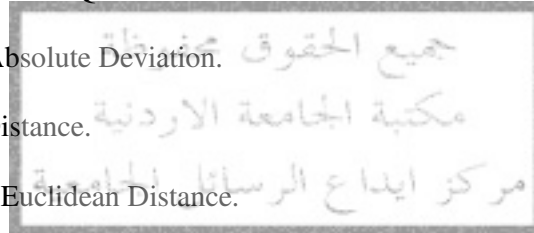
NLPR: National Laboratory of Pattern Recognition.

CASIA: Chinese Academy of Sciences, Institute of Automation.

ROC: Receiver Operating Characteristic.

CIR: Correct Identification Rate.

CAR: Correct Acceptance Rate.



## LIST OF APPENDICES

**Appendix A:** Software Architecture.

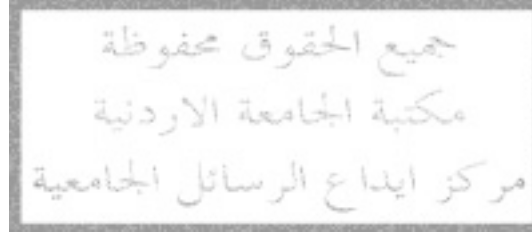
**Appendix B:** Program Source Code for Iris Segmentation.

**Appendix C:** Program Source Code for Iris Normalization.

**Appendix D:** Program Source Code for Iris Feature Extraction and Encoding.

**Appendix E:** Program Source Code for Iris Matching.

**Appendix F:** Program Source Code for Result Analysis.



# AUTOMATED PERSONAL RECOGNITION SYSTEM BASED ON HUMAN IRIS ANALYSIS

By

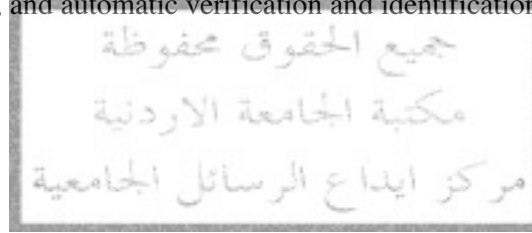
Supervisor  
**Dr. Dia I. Abu- Al-Nadi**

## ABSTRACT

Over the past decade, many biometric recognition approaches have been receiving extensive attention. Some of these approaches based on fingerprint, facial feature, hand geometry, signature, speech pattern, retina, iris pattern, etc. These approaches differ in the degree of intra-personal variation. Among these approaches, the most suitable and reliable approach with a small degree of intra-personal variation is that based on the iris pattern. Ophthalmologists and anatomists show that the visual appearance of the iris is unique to an individual and is stable with age. These characteristics make the human iris useful for personal identification.

This thesis presents a proposed algorithm for iris recognition that implements the Hough transform and the Compass gradients in a fast and efficient manner to locate the iris region in a captured eye image, then it assesses the quality of the iris image and accepts only the clear images. Then it unwraps the iris ring to a fixed size rectangular normalization. After that, the phase data from 1D Log-Gabor filter is extracted and encoded in an efficient manner that gives a small and suitable iris template. Finally, these templates are classified using a simple matching ratio, which is similar to the Hamming distance.

Experimental tests were performed and analyzed using MATLAB 6.1 program and CASIA iris database, which is a common iris database and specified for iris recognition researchers, and it contains 756 iris images. The proposed algorithm was tested in identification and verification modes. The results of these tests prove that the proposed algorithm has an encouraging performance; it achieves a segmentation rate equals to 99 %, in the identification mode it has a high identification rate equals to 99.81 %, and in the verification mode it has a high performance when comparing its Receiver Operating Characteristic (ROC) curve with that of the well known methods for Daugman, Wildes *et al*, and L. Ma *et al*. So the proposed algorithm can be characterized as a fast, reliable, and automatic verification and identification algorithm.



## Introduction

Human identification or verification is not a new topic by itself, but the methods for human identification and verification are changed according to the human activities requirements and the development of information technology. There are traditional methods for human identification and verification like: physical key, personal identification number (PIN), and a secret password. These methods suffer from various problems. For example, it can be lost, forgotten, or guessed. Today one of the important human activities requirements is a fast, reliable, and automatic human identification and verification system. So other types of methods based on biometric measurements are used, such as fingerprints, voiceprints, hand written signature, and hand shape. Unfortunately, these methods can be invasive; typically, the operator is required to make physical contact with a sensing device or otherwise take some special action as reciting a specific phonemic sequence (Wildes *et al.*, 1997). One possible alternative to these methods that has the potential to be less invasive is automated face recognition, but in face recognition, difficulties arise from the fact that the face is a changeable social organ displaying a variety of expressions, as well as being an active three Dimensional (3D) object whose image varies with viewing angle, illumination, and age. It has been shown that for facial images taken at least one year apart; even the best current algorithms have error rates of 43% to 50%. Against this intra-class (same faces) variability, inter-class (different faces) variability is limited because different faces have the same basic set of features, in the same canonical geometry (Daugman, 2002).

Automated iris recognition is yet another alternative for noninvasive verification and identification of people. Interestingly, the spatial patterns that are apparent in the human iris are highly distinctive to an individual (Flom, 1987), (Daugman, 1994, 2002),

(Wildes *et al.*, 1996, 1997), (Matsushita, 1999), (Kamada, 2004). Like the face, the iris is an overt body that is available for remote assessment.

Generally, the iris recognition system consists of two main parts; hardware part to capture the iris image of the person as an input to the system, and software part to analyze the iris image and compare it with that in a predetermined database, then according to the comparing result, it gives identification or verification result as an output of the system. Both parts are closely related with each other, and the performance of the entire system depends on both parts. So many efforts were done to improve the performance of both parts. The effort in this thesis is focused in the software part only.

In this thesis, a proposed algorithm for iris recognition is presented in some details. In this algorithm the main objective for any effort in iris recognition was considered. This objective is a fast, reliable, and automatic human identification and verification system.

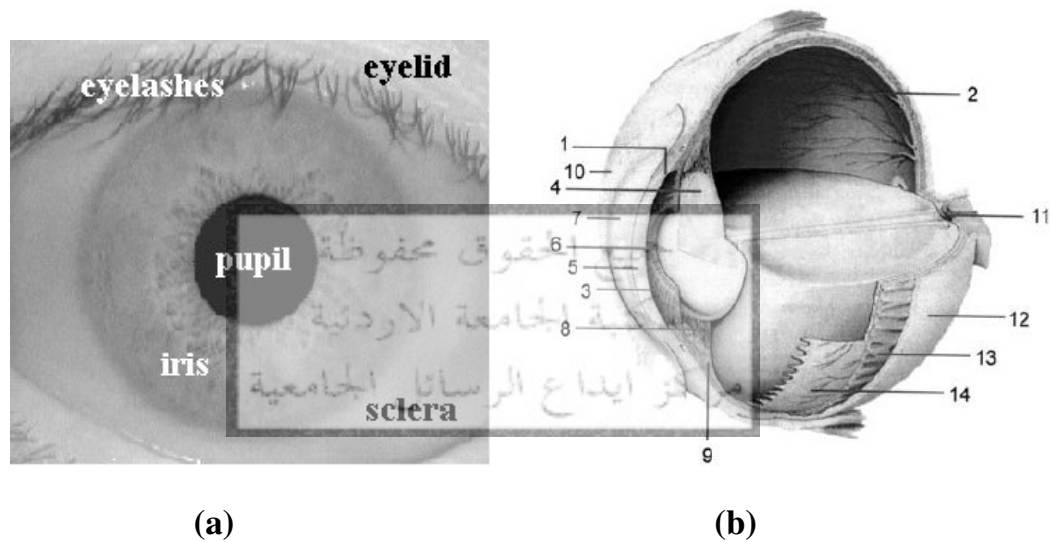
This thesis is divided into six parts: part one includes overview about the human iris, iris recognition technology, a literature review of iris recognition, and the objective of this study. In part two, the hardware part of the iris recognition system is presented, which is the iris image acquisition and iris liveness detection. The software part of the iris recognition system consists of: Iris segmentation or localization and iris image quality assessment are presented in part three, Iris normalization is presented in part four, and Iris feature extraction, encoding, and matching are presented in part five. Experiments, results analysis, and conclusions are discussed in part six.



## 1. The Human Iris

### 1.1 Anatomy of The Human Iris

The iris is the visible colored part of the eye and extends interiorly from the ciliary body, lying behind the cornea in front of the lens. It divides the anterior segment of the eye into anterior and posterior chambers which contain aqueous fluid secreted by the ciliary body (Waugh and Grant, 2001) as shown in Figure 1.

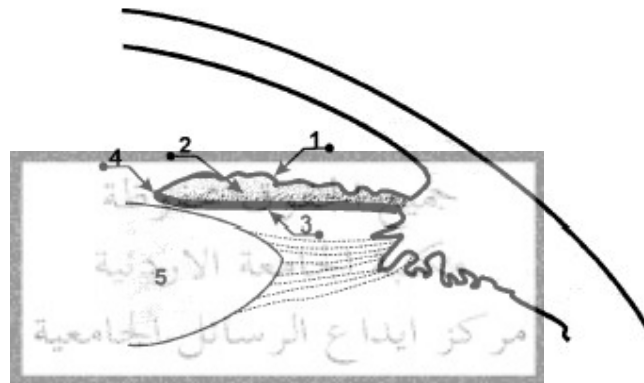


**Figure 1: (a) A front view of the human eye, image 088\_2\_3 from CASIA database. (b) Human eye (1-anterior chamber, 2-posterior chamber, 3- iris, 4- lens, 5- aqueous humour, 6- pupil, 7- cornea, 8- zonula, 9- ciliary body, 10- conjunctiva, 11- optic nerve, 12- sclera, 13- choroids, 14- retina. (Muron and Pospisil, 2000)**

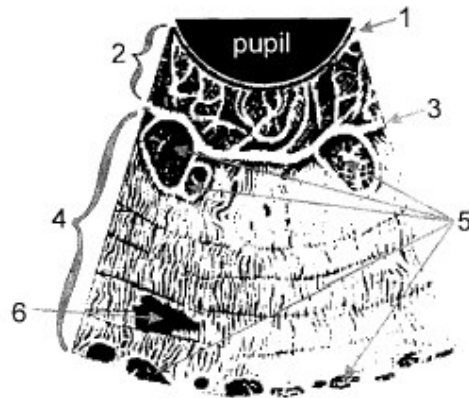
Section through the human iris is shown in Figure 2. From this figure we can see that the iris is composed of several layers:

1. **Anterior layer:** it is a visible layer, it bears the gaily-colored relief and it is very lightly pigmented due to genetically determined density of melanin pigment granules (Muron and Pospisil, 2000). It differs from the stromal layer in being more densely packed, especially with individual pigment cells called chromatophores (Wildes, 1997). Figure 3 shows the surface of this layer.

2. **Stroma layer:** muscles and vascularized stroma are found in this layer (Muron and Pospisil, 2000).
3. **Posterior layer:** it is invisible layer, consists of heavily pigmented epithelial cells that make it light tight (i.e., impenetrable by light) (Wildes, 1997). The surface of this layer is finely radiantly and concentrically furrowed with dark brown color. (Muron and Pospisil, 2000).



**Figure 2: Section through the human iris (Muron and Pospisil, 2000). (1- anterior layer, 2- stroma, 3- posterior layer, 4- pigment frill, 5- lens).**



**Figure 3: Anterior surface of the human iris (Muron and Pospisil, 2000). 1- Pigment frill, 2- pupillary area, 3- collarette, 4- ciliary area, 5- crypts, and 6- pigment spot)**

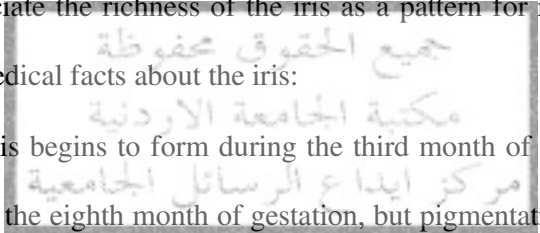
The visual appearance of the iris is a direct result of its multilayered structure. From Figure 3 we can see some of the visible features of the human iris, which are important to identify a person: pigment frill, collarette; which appears as a zigzag pattern, crypts, and pigment spot.

In addition to these features there are visible rare anomalies; due to aging or trauma, atrophic areas may appear on the iris, resulting in a "moth-eaten" texture. Tumors may grow on the iris, or congenital filaments may occur connecting the iris to the lens of the eye (Muron and Pospisil, 2000).

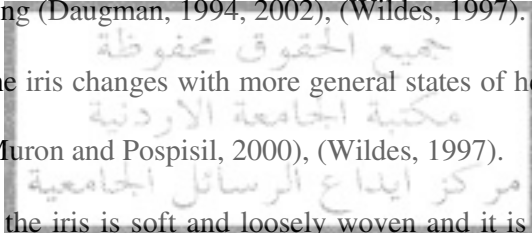
### 1.2 Medical Facts about Human Iris

To appreciate the richness of the iris as a pattern for recognition, it is useful to consider some medical facts about the iris:

- The human iris begins to form during the third month of gestation. The structure is completed by the eighth month of gestation, but pigmentation continues into the first year after birth. The iris grows from the ciliary body (Muron and Pospisil, 2000).
- The color of the iris is genetically determined and depends on the number of pigment cells present. Albinos have no pigment cells and people with blue eyes have fewer than those with brown eyes (Waugh and Grant , 2001).With blue irises resulting from a few pigment; long wavelength light penetrates and is absorbed by the pigment epithelium, while shorter wavelengths are reflected and scattered by the stroma (Daugman,2002).
- The structure of the iris unique to an individual and is stable with age. The left and right iris of a single person seems to be completely independent, and identical twins possess uncorrelated iris patterns. These facts come from two main evidence sources; the first source of evidence is clinical observations by the ophthalmologists and anatomists, the second source is developmental biology (Wildes, 1997).



- The most important function of iris is controlling the size of the pupil. Illumination, which enters the pupil and falls on the retina of the eye, is controlled by muscles in the iris. They regulate the size of the pupil and this is what permits the iris to control the amount of light entering the pupil. The change in the size results from involuntary reflexes and is not under conscious control (Muron and Pospisil, 2000). The iris is supplied by parasympathetic and sympathetic nerves. Parasympathetic stimulation constricts the pupil and sympathetic stimulation dilates it (Waugh and Grant, 2001).
- Pupil diameter undergoes small oscillations once or twice per second, even under uniform lighting (Daugman, 1994, 2002), (Wildes, 1997).
- Claims that the iris changes with more general states of health (iridology) have been discredited (Muron and Pospisil, 2000), (Wildes, 1997).
- The tissue of the iris is soft and loosely woven and it is called stroma (Muron and Pospisil, 2000).
- The pupil may not be exactly circular in shape and its deviation from the circle is a visible characteristics. Centers of the iris and the pupil may be different (Daugman, 1994, 2002), (Muron and Pospisil, 2000).
- The average diameter of the iris is 11 mm, and the pupil radius can vary from 0.1 to 0.8 of the iris radius (Daugman, 2002).
- Formation of the unique patterns of the iris is random and not related to any genetic factors (Wildes, 1997). The only characteristic that is dependent on genetics is the pigmentation of the iris, which determines its color.
- Iris imaging using the visible light shows the predominant texture, whereas in the near infrared (NIR) wavelengths used for imaging at a distance up to 1 meter, even darkly pigmented irises reveal rich and complex features (Daugman,2002).

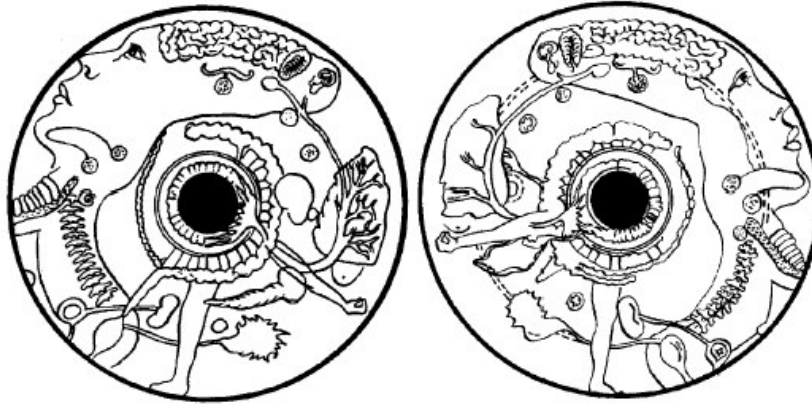


### 1.3 Reasons for Studying the Human Iris

The first reason for studying the human iris is the "IRIDOLOGY". Iridology was based on the belief that iris patterns reveal the state of human body organs and so the appearance of the human iris texture may be changed according to the state of human body organs. The chart in Figure 4 illustrates this belief and shows how each individual organ of the body is mapped into a specific region of each iris for the purposes of diagnosis. For example, near the pupil, there is the territory of digestion. This belief was popular in California and Rumania (Muron and Pospisil, 2000).

Many reviews published in medical journals reporting various scientific tests of iridology (Berggren, 1985. Cockburn, 1981. Worrall, 1984, 1985. Knipschild, 1988. Simon, Worthen, and Mitas, 1979), and they all dismiss iridology as a medical fraud. In particular, the review by Berggren (1985) concludes: "Good care of patients is inconsistent with deceptive methods, and iridology should be regarded as a medical fraud". So iridology is rejected and is called a medical fraud (Muron and Pospisil, 2000), (Wildes, 1997).

The second reason for studying the human iris is identification of a person, it is against the first reason; it based on the medical fact that the structure of the iris unique to an individual and is stable with age.



**Figure 4: Structure of the human iris in relation to the human organs according to the Iridology (Muron and Pospisil, 2000).**

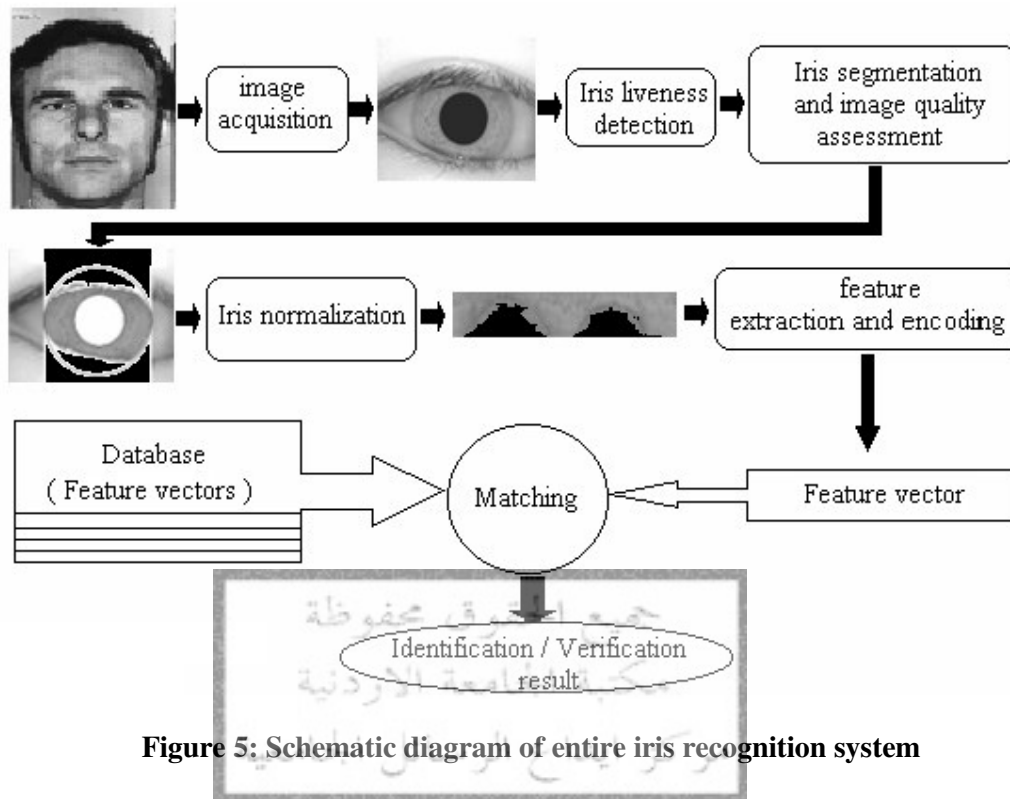
## 2. Iris Recognition Technology

Generally, the iris recognition system consists of two main parts; hardware part to capture the iris image of the person as an input to the system, and software part to analyze the iris image and compare it to that in a database, then according to the comparing result it gives identification or verification result as an output of the system. Both parts are closely related with each other, and the performance of the entire system depends on both parts. So many efforts are done to improve the performance of both parts.

For the hardware part, capturing an iris image of high quality (clear texture) is one of the major challenges for practical applications. But for the software part, it can be divided into these steps:

1. Iris liveness detection .( Hardware may be shared with this step)
2. Iris segmentation and iris image quality assessment.
3. Iris normalization.
4. Iris feature extraction, encoding, and matching.

Schematic diagram of entire iris recognition system is shown in Figure 5.



**Figure 5: Schematic diagram of entire iris recognition system**

Using the human iris for identification has some advantages, but also some disadvantages. These can be summarized as follows:

♣ **Advantages:**

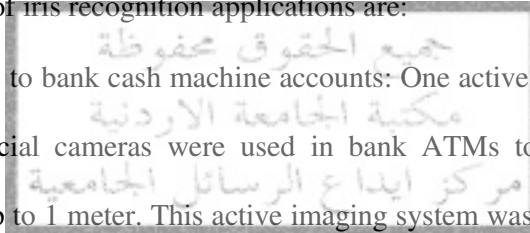
- Iris is stable throughout the life.
- Iris is visible from a distance.
- Iris is a highly protected, internal organ of the eye.
- Iris patterns possess a high degree of randomness.
- Limited genetic penetration.
- Changing pupil size provides a natural test against artifice.
- As it is a planar object its image is relatively insensitive to angle of illumination, and changes in viewing angle cause only a fine transformations.

♣ **Disadvantages:**

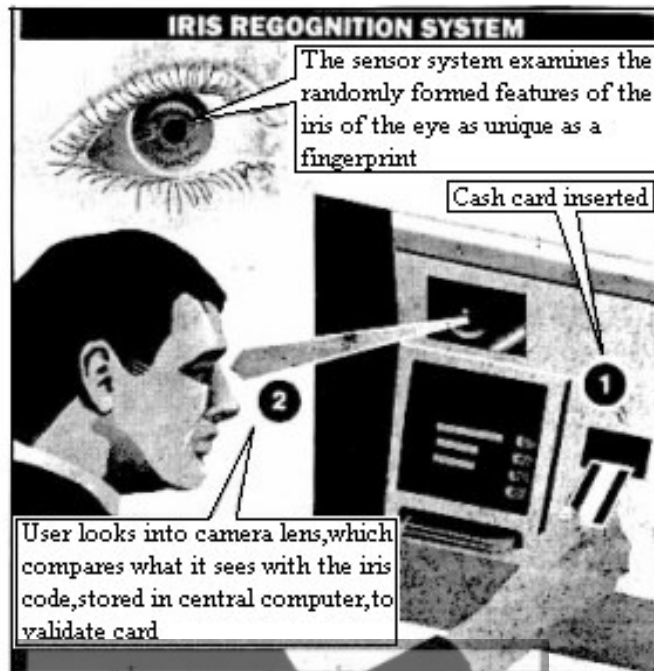
- Iris is a small target (1cm) to acquire from a distance.
- Iris is a moving target.
- Iris locates behind a curved, wet, and reflecting surface.
- Iris is obstructed by eyelashes, eyelids, and reflections.
- Illumination should not be very visible or bright. As humans are very sensitive about their eyes.

Iris recognition can be used in any existing use of keys, cards, PINs, passwords, or any security tools. The iris is a living security tool (e.g. living password, living passport). Some of iris recognition applications are:

- Secure access to bank cash machine accounts: One active imaging system developed in 1996; special cameras were used in bank ATMs to capture iris images at a distance of up to 1 meter. This active imaging system was installed in cash machines both by NCR Corp in UK and by Diebold Corp in USA in successful public trials in several countries during 1997 - 1999. To illustrate this application see Figure 6.







**Figure 6: Secure access to bank cash machine accounts using iris recognition technology.**

- Information security applications; "Panasonic AuthentiCam" digital camera is a small imaging device used for handheld, desktop, e-commerce, and other information security applications.
- For physical security, a building entry and portal control camera featuring auto-zoom and auto-focus called "IrisAccess" has been developed by the Korean conglomerate LG.
- Document-free air travel, check-in, international border crossing, and security procedures based on iris recognition kiosks in airports have been developed by EyeTicket Corp in USA. Several airports worldwide have recently installed the Daugman algorithms for security and immigration control, including London Heathrow Airport, Amsterdam Schiphol Airport, Frankfurt, and Charlotte Airports in USA.

### 3. Literature Review of Iris Recognition Technology

In the year 1880, Albert Bertillon was the first person to notice that the human iris can be used in identification of a person (Muron and Pospisil,2000).And the first use of the iris recognition was in the Parisian penal system to distinguish inmates(Wildes,1997). The idea of using iris patterns for personal identification was originally documented in an ophthalmology textbook by James Doggarts in 1949, and may have been proposed as early as 1936 by ophthalmologist Frank Burch. In the science fiction (Wildes, 1997). In the year 1987, two American ophthalmologists, Leonard Flom and Aran Safir patented a proposed concept of automated iris recognition, but this concept does not appear, however, that this team ever developed and tested a working system (Wildes, 1997). Early work toward actually realizing a system for automated iris recognition was carried out at Los Alamos National Laboratories, CA (Wildes, 1997). Other research into automated iris recognition was carried out in North America and Europe; however, these efforts have not been well documented to date (Wildes, 1997). Most of real works on iris recognition were done in the last decade.

Daugman (1994) developed actual algorithm for iris recognition. This algorithm is the base for all current iris recognition systems. The Daugman algorithm is patented and the rights were owned by Iridian Technologies Company. This algorithm finds the iris in a live video image of a person's eye. It defines a circular pupillary boundary between the iris and the pupil portions of the eye, and it defines another circular boundary between the iris and the sclera portions of the eye. The algorithm fits the circular contours via Integro-differential operator, this operator can be defined as follows:

$$\max(r, x_o, y_o) = \left\{ \frac{\partial}{\partial r} \int_0^{2\pi} I(r * \cos \theta + x_o, r * \sin \theta + y_o) \right\} \quad (1.1)$$

Where  $(x_o, y_o)$  denotes the center of the searched circular boundary, and  $r$  its radius. Then it normalizes the iris ring to a rectangular block of a fixed size. After that it finds the iris code according to the real and imaginary parts of two dimensions (2D) Gabor filters outputs. And by using the hamming distance compares the code with stored iris codes. This system is sensitive to illumination variations. Daugman employed carefully designed devices for image acquisition to ensure that the iris is located at the same location within the image, and the images have the same resolution and are glare free under fixed illuminations. However, these requirements are not always easy to be satisfied especially in practical applications.

Wildes *et al.* (1996) developed iris recognition system that uses two cameras. The first one is with low resolution and the second one with high resolution. The obtained image from the second camera is processed to extract the iris region by using a binary edge map and Hough transform. In the next step, the reduced image is filtered by Laplacian of Gaussian filters to get a Laplacian pyramid constructed with four different resolution levels to represent the iris texture. Then the normalized correlation between the acquired and data base representations is used for pattern matching.

By this system the problem of specular reflection of the light source that appears in the image acquisition step is eliminated. This allows for more of the iris detail to be available for subsequent processing. And the coupling of a low light level camera with diffuse camera allows for a level of illumination that is entirely unobjectionable to human operators. This system is sensitive to illumination variations; Wildes system employed carefully designed devices for image acquisition to ensure that the iris is located at the same location within the image, and the images have the same resolution

and are glare free under fixed illuminations. However, these requirements are not always easy to be satisfied especially in practical applications. This system is not used for identification, only for verification. No commercial version of this system is available.

Boles and Boashash (1998) proposed an algorithm for iris feature extraction using zero-crossing representation of one dimension (1D) wavelet transform. This work focused on iris representation and matching algorithm without an image acquisition.

Matsushita (1999) developed an iris identification system. This system is used to identify customers. At first, the camera identifies the head of a customer, finds the position of the eyes, zooms up and photographs the irises. The computer extracts only a portion of the iris data that is significant to identify the customer.

Zhu *et al.* (2000) adopted multichannel Gabor filters and 2D wavelet transform to capture both global and local details in an iris.

Lim *et al.* (2001) decomposed an iris image into four levels using 2D Haar wavelet transform and quantized the fourth-level high frequency information to form an 87-bit code. A modified competitive learning neural network (LVQ) was used for classification.

Tisse *et al.* (2002) implemented the gradient decomposed Hough transform / Integro-differential operators combination for iris localization and the analytic image (a combination of the original image and its 2D Hilbert transform) to extract pertinent information from iris texture. Similar to the algorithm by Daugman, they sampled binary emergent frequency images to form a feature vector and used Hamming distance for matching.

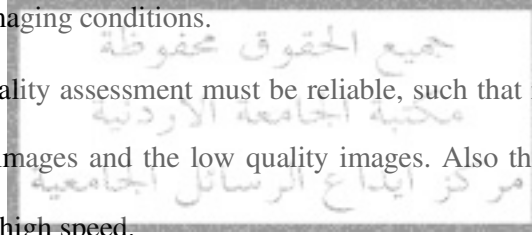
Ma *et al.* (2002, 2003) adopted multichannel Gabor filters and circular symmetric filters for iris feature extraction.

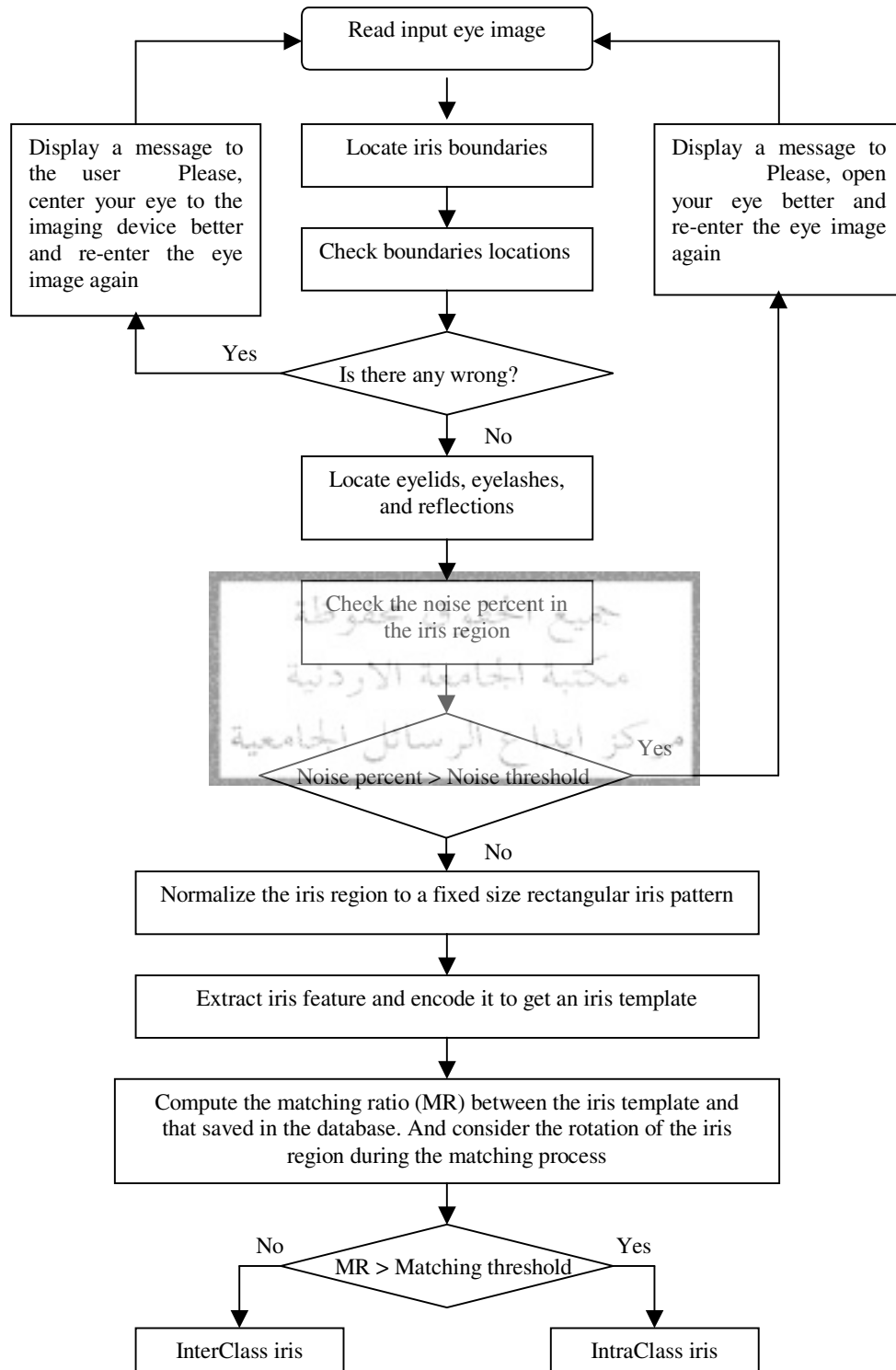
#### 4. The Objective of this Study

The objective of this study is to propose an iris recognition algorithm that has the main characteristics of any practical recognition system. These characteristics are the high speed, reliability, and automation.

The flowchart of the proposed algorithm is shown in Figure 7. And to satisfy the desired characteristics, the following must be achieved:

- Accurately, automatically, and with high speed, iris segmentation process must locate the inner and outer boundaries of the iris, and isolate the eyelids, reflections and eyelashes. Also must be reliable, such that it is insensitive to the noise that may come from imaging conditions.
- Iris image quality assessment must be reliable, such that it can classify correctly the high quality images and the low quality images. Also the assessment process must be done with high speed.
- Iris normalization must be an efficient method such that it can map the iris region to a fixed size without affecting seriously the original iris texture. The result of normalization must be suitable with the feature extraction process. Also the mapping process must be done with high speed.
- Feature extraction and encoding processes must be accurate and efficient such that the resultant iris template (feature vector) has a small size and represents the iris texture correctly. Also these operations must be done with high speed.
- Correctly and with high speed, matching process must find the degree of similarity between two iris templates. Such that a clear separation between the intra-class (same irises) and inter-class (different irises) can be found.





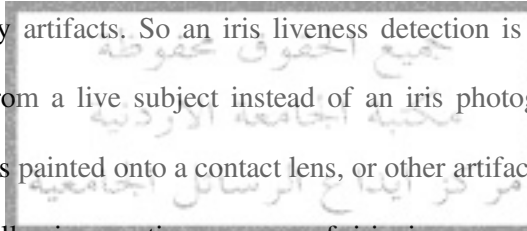
**Figure 7: The flowchart of the proposed iris recognition algorithm**

## Image Acquisition and Iris Liveness Detection

### 1. Introduction

Generally, image acquisition is the first step in the hardware part of any iris recognition system. And it is one of the major challenges of automated iris recognition in practical applications. It is not easy to capture an iris image with a high quality that can be used for recognition. So an accurate and clever technique is required to capture the iris region with a high quality.

Unfortunately, illegal uses are not limited to the traditional personal identification methods (fingerprints, voiceprints, signatures, and etc.), but also the iris can be marred by artifacts. So an iris liveness detection is required to ensure that an input image is from a live subject instead of an iris photograph, a video playback, a glass eye, fake iris painted onto a contact lens, or other artifacts.



In the following sections, some of iris image acquisition and iris liveness detection techniques are presented.

### 2. Iris Image Acquisition Techniques

Capturing a high quality iris image for recognition is one of the major challenges of automated iris recognition in practical applications. This will be clear after studying some important issues of iris image acquisition:

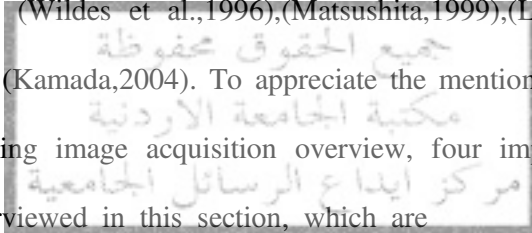
- Iris is a small target; about 1 cm in diameter.
- To get a correct recognition, acquired iris images must have a sufficient resolution.
- It is important to have good contrast in the interior iris pattern without resorting to a level of illumination that annoys the operator.
- Humans are very sensitive about their eyes.
- Acquired images must contain the most iris region as much as possible. So these images must be well framed (i.e., centered) without unduly constraining the operator

(i.e., preferably without requiring the operator to employ an eye piece, chin rest, or other contact positioning).

- Artifacts noise in the acquired images (e.g., due to specular reflections, optical aberrations, etc.) should be eliminated as much as possible.
- Iris is obstructed by eyelashes and eyelids.

When designing an imaging acquisition system, we should consider three main aspects (Ma *et al.*, 2003): Lighting system, positioning system, and Physical capturing system.

Much works have been done on iris image acquisition: (Flom,1987), (Daugman,1994), (Wildes *et al.*,1996),(Matsushita,1999),(Lim *et al.*,2001)( Ali and Hassanien,2003) (Kamada,2004). To appreciate the mentioned challenges and for the sake of completing image acquisition overview, four important image acquisition systems are overviewed in this section, which are



Wildes *et al.*

system, , and Lim *et al.* system. As our efforts in this thesis is focused in the software part, we only overview these systems without details.

acquisition system is shown in

Figure 8 (Wildes, 1997). This system can be summarized as follows:

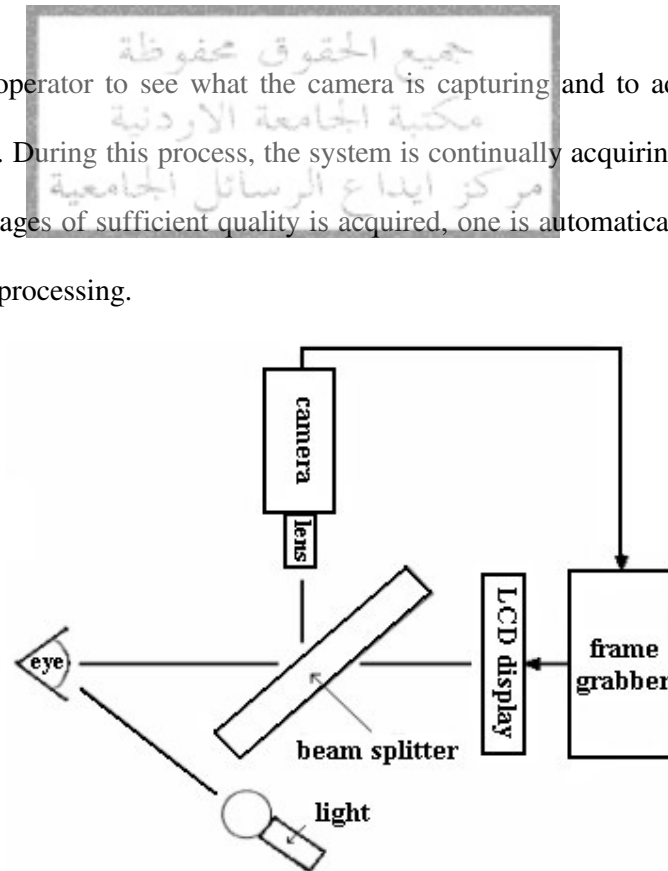
- The Daugman system captures images with the iris diameter typically between 100 and 200 pixels from a distance of 1546 cm using a 330-mm lens.
- Video rate capture is exploited in this system.
- The overall spatial resolution and focus that results from this design appear to be sufficient to support iris recognition.
- It is simple and compact system. By careful positioning of the light source below the operator, reflections of the point source on eyeglasses can be avoided in the imaged



iris. Without placing undue restriction on the operator, however, it has not been iris region. Therefore, this design requires that the region of the image where the point source is seen (the lower quadrant of the iris) must be omitted during matching since it is dominated by artifact.

- The light used in this design is a visible light. And the camera used in this design is a monochrome video camera with 8 bit gray level resolution.
- This system requires the operator to self position his eye region in front of the camera. It provides the operator with live video feedback via a miniature liquid crystal

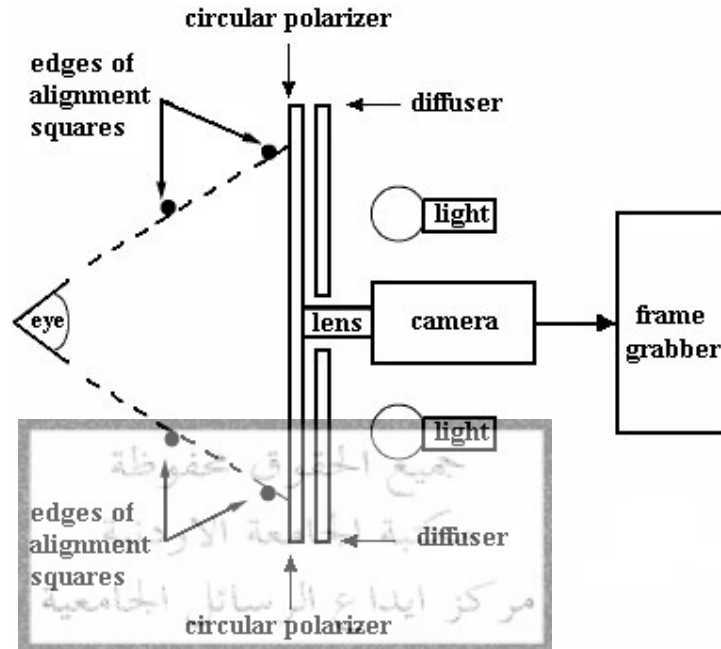
allows the operator to see what the camera is capturing and to adjust his position accordingly. During this process, the system is continually acquiring images. Once a series of images of sufficient quality is acquired, one is automatically forwarded for subsequent processing.



**Figure 8: A schematic diagram of the Daugman image-acquisition system (Wildes, 1997)**

## 2.2 Wildes *et al.* Iris Image Acquisition System

A schematic diagram of the Wildes *et al.* iris image acquisition system is shown in Figure 9 (Wildes, 1997).



**Figure 9: A schematic diagram of the Wildes *et al.* image-acquisition system (Wildes, 1997).**

This system can be summarized as follows:

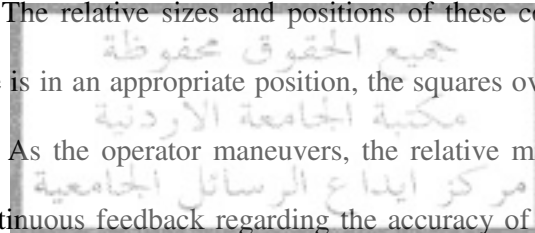
- Wildes *et al.* system images the iris with approximately 256 pixels across the diameter from 20 cm using an 80-mm lens.
- Video rate capture is exploited in this system.
- The overall spatial resolution and focus that results from this design appear to be sufficient to support iris recognition.

- certain advantages result.

First, the use of matched circular polarizer at the light source and the camera essentially eliminates the specular reflection of the light source. This allows for more of the iris detail to be available for subsequent processing. Second, the

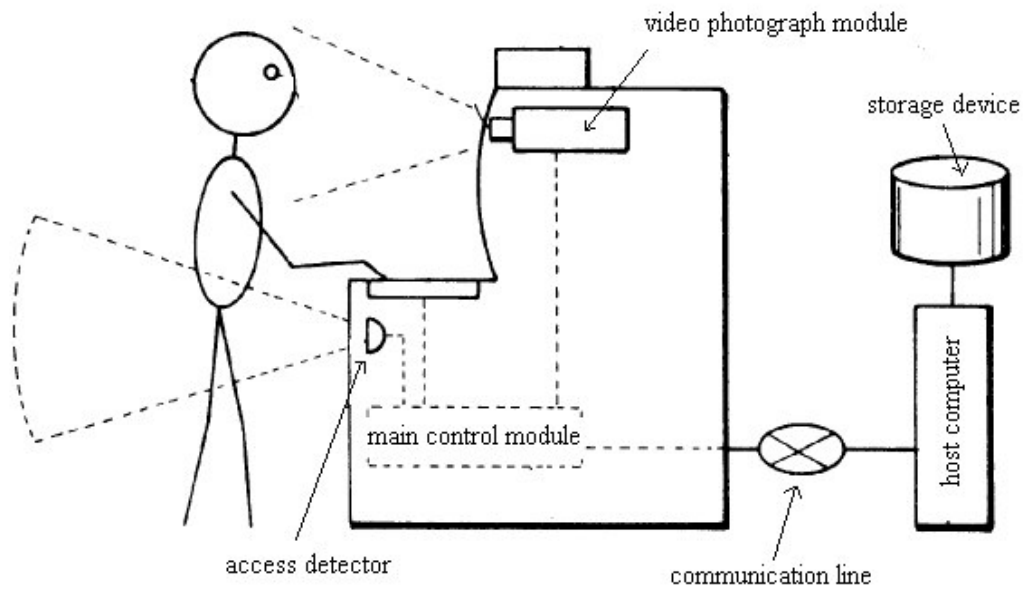
coupling of a low light level camera (a silicon intensified camera) with a diffuse illuminant allows for a level of illumination that is entirely unobjectionable to human operators.

- The light used in this design is a visible light. And the camera used in this design is a monochrome video camera with 8 bit gray level resolution.
- This system requires the operator to self position his eye region in front of the camera. The system provides a reticle to aid the operator in positioning. In particular, a square contour is centered around the camera lens so that it is visible to the operator. Suspended in front of this contour is a second, smaller contour of the same shape. The relative sizes and positions of these contours are chosen so that when the eye is in an appropriate position, the squares overlap and appear as one to the operator. As the operator maneuvers, the relative misalignment of the squares provides continuous feedback regarding the accuracy of the current position. Once the operator has completed the alignment, he/she activates the image capture by pressing a button.



### 2.3

A perspective view of the automatic teller machine according to the Matsushita image acquisition system is shown in Figure10 (Matsushita, 1999). The access detector detects that a customer is approaching the Automated Teller Machine (ATM), and calculates the distance to, and the position of the customer. Then, the video photograph module finds the face of the customer, identifies the position of the eyes, and informs the main control module that iris data may be obtained. Upon receiving this information, the main control module zooms up the camera and photographs the irises. Then the iris data sent to the host computer to complete the software processes.



**Figure 10: A perspective view of the automatic teller machine according to the Matsushita image acquisition system (Matsushita, 1999).**

#### 2.4 Lim *et al.* Iris Image Acquisition System

A schematic diagram of Lim *et al.* iris image acquisition system is shown in Figure 11 (Lim *et al.*, 2001). An image surrounding human eye region is obtained at a distance from a Charge Coupled Devices (CCD) camera without any physical contact to the device. To acquire more clear images through a CCD camera and minimize the effect of the reflected lights caused by the surrounding illumination, two halogen lamps are arranged as the surrounding lights. The size of the image acquired under these circumstances is 320x240 pixels.

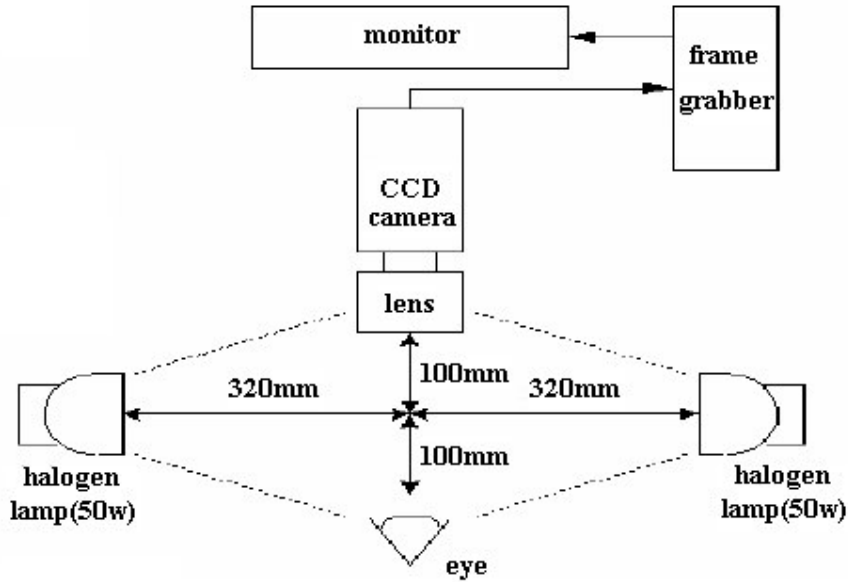


Figure 11: A schematic diagram of Lim *et al.* iris image acquisition system (Lim *et al.*, 2001)

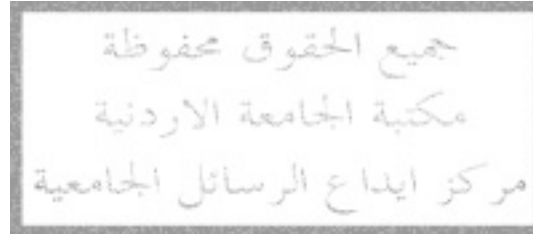
### 3. Iris Liveness Detection Techniques

Biometrics such as fingerprints, voiceprints, signatures, or handprints all have significant drawbacks. All are easily forged and can be counterfeited. Fingerprints or handprints require physical contact, and they also can be counterfeited and marred by artifacts. Electronically recorded voiceprints are susceptible to changes in a person's voice.

Really, these illegal uses are not limited to the previous personal identification methods, but also the iris can be marred by artifacts. So the iris liveness detection aims to ensure that an input image is from a live subject instead of an iris photograph, a video playback, a glass eye, fake iris painted onto a contact lens, or other artifacts. However, efforts on iris liveness detection are still limited, though iris liveness detection is highly desirable. Daugman (1994) suggested that an important feature of a living eye is that the pupil diameter undergoes small oscillations once or twice per second, even under uniform lighting. So a photograph of an iris, or contact lens imprinted with an iris

image, would not exhibit such variation in time, so monitoring the pupil diameter over

suggestion and also he exploited the fact that the iris reacts (changes in the pupil diameter) very quickly to changes in illumination and monitoring the reaction to a controlled illuminant could provide an evidence on iris liveness. Kamada (2004) exploited the eyeball movements for detecting whether iris information from a living body or not. How to utilize the optical and physiological characteristic of the live eye to implement effective liveness detection remains to be an important research topic (Ma *et al*, 2003).

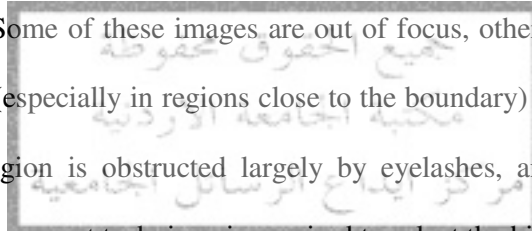


## Iris Segmentation and Iris Image Quality Assessment

### 1. Introduction

The iris is an annular part between the pupil (inner boundary) and the sclera (outer boundary) as shown in the Figure 1.a . Without placing constraints on the human operator the acquired image will not contain only the region of interest; but also some unuseful parts such as: pupil , eyelids, eyelashes, also, specular reflections can occur within the iris region. A technique is required to isolate and exclude these unuseful parts and get the region of interest, in other words we need an iris segmentation algorithm.

Unfortunately, not all the segmented irises are clear, suitable, and sharp enough for recognition. Some of these images are out of focus, others contain many noticeable interlacing lines (especially in regions close to the boundary) caused by eye motion, and in others, iris region is obstructed largely by eyelashes, and eyelids. Therefore, iris image quality assessment technique is required to select the high quality images.



In the following sections, some of the most well known techniques for iris segmentation and iris image quality assessment are presented. In the last section, an efficient and fast proposed approach for iris segmentation and iris image quality assessment is presented and discussed in some details.

### 2. Iris Segmentation Techniques

In this section, the most well known techniques for iris segmentation are presented. These techniques are: -differential Operator and Hough transform which was used by Wildes and others. And also we present Kong and Zhang method for eyelashes detection.

Both Daugman and Wildes model the inner and outer boundaries with circular contours. And the upper and lower eyelids with arcs (Daugman, 2002), (Wildes, 1997).

The two systems differ mostly in the way that they search the contours parameters. These techniques will be illustrated in the following sections.

## 2 Integro-Differential Operator

Daugman (1994) defined an Integro-differential Operator to find the parameters defining the circular inner and outer boundaries of the iris and the arcs of the upper and lower eyelids. This operator is defined as

$$\max_{(r, x_o, y_o)} \left| G_{\sigma}(r) * \frac{\partial}{\partial r} \oint_{r, x_o, y_o} \frac{I(x, y)}{2\pi r} ds \right| \quad (3.1)$$

where  $I(x, y)$  is the eye image,  $G_{\sigma}(r) = (1/\sqrt{2\pi\sigma})e^{-((r-r_o)^2/2\sigma^2)}$  is a radial Gaussian smoothing function with center  $r_o$ ,  $ds$  is an element of the contour of the circle given by a radius  $r$ , and center  $(x_o, y_o)$ . The division by  $2\pi r$  is to normalize the integral. The operator searches iteratively for a maximum contour integral derivative with increasing radius at successively finer scales of analysis through the three space parameters of radius and center coordinates  $(r, x_o, y_o)$  defining a path of contour integration. The upper and lower eyelids are localized by changing the path of the contour integration from a circular to an arc.

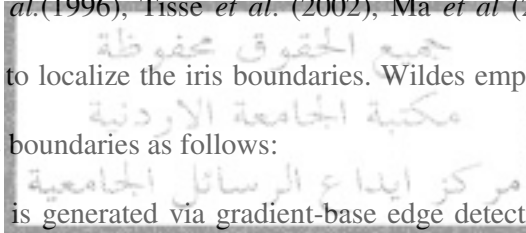
There are a number of problems with the Integro-differential Operator; the algorithm can fail where there is noise in the eye image, such as from reflections, since it works on a local scale. And as every point in an image  $I(x, y)$  is used as a test center, a high computational load is required.



## 2.2 Hough Transform

Hough transform is a standard technique that can be used to find the parameters of simple geometric shapes as circle, curves, and lines in an image (Gonzalez and Woods, 2002). Implementation of this technique is summarized in two steps; firstly, an edge map of the image is found by a suitable edge detector technique. Secondly, voting in Hough space for the parameters of the wanted shape (e.g. circle, curve, or line) is done through each edge point. The edge point votes for a certain shape if this point is located on the edge map of this shape. Mathematically, if this point satisfies the mathematical model for this shape.

Wildes *et al.*(1996), Tisse *et al.* (2002), Ma *et al* (2003), and others used the Hough transform to localize the iris boundaries. Wildes employed the Hough transform to localize the iris boundaries as follows:



1. An edge map is generated via gradient-base edge detection; this operation consists of thresholding the magnitude of the image intensity gradient. The derivatives in the horizontal direction are weighted for detecting the eyelids, and the derivatives in the vertical direction are weighted for detecting the outer circular boundary of the iris. The motivation for this is that the eyelids are usually horizontally aligned, and also the eyelids edge map will corrupt the circular iris boundary edge map if using all gradient data. Taking only the vertical gradients for locating the iris boundary will reduce influence of eyelids when performing circular Hough transform in the next step. All of these will give accurate and efficient iris segmentation.
2. Voting in Hough space for the parameters of the inner circle (inner boundary of the iris) and the outer circle (outer boundary of the iris) is done through each edge point. These parameters are the pupil center  $(x_p, y_p)$  and pupil radius  $r_p$ , and the iris center

$(x_i, y_i)$  and iris radius  $r_i$  which are able to model the inner and outer boundaries of the iris by the following circles equations:

$$\text{The inner circle equation: } (x_e - x_p)^2 + (y_e - y_p)^2 = r_p^2 \quad (3.2)$$

$$\text{The outer circle equation: } (x_e - x_i)^2 + (y_e - y_i)^2 = r_i^2 \quad (3.3)$$

Where  $(x_e, y_e)$  are the edge points that located on the outer or inner boundary .

There are number of problems with the Hough transform; it requires threshold value to be chosen for edge detection, and this is not easy in practical applications. This problem is not found in the Daugman Integro-differential operator. And Hough transform is computationally intensive. And thus may not be suitable for real time applications.

### 2.3 Eyelashes and Noise Detection

Kong and Zhang distinguish between two types of eyelashes (Masek, 2003):

- Separable eyelashes : are detected using 1D Gaussian smoothing function, since the convolution of the pixels of these eyelashes with this function results in low values, then by thresholding the results of convolution process, we can distinguish between separated eyelashes pixels and others.
- Multiple eyelashes: are detected using the variance of intensity in a small window around each pixel, if the variance of intensity is smaller than a threshold, this pixel is considered as multiple eyelashes pixel.

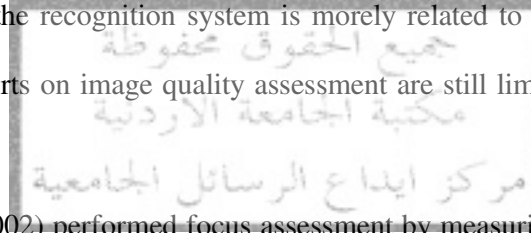
Specular reflections are detected using a certain threshold, as these reflections have a higher intensity values than other regions in the images.

### 3. Iris Image Quality Assessment Techniques

Unfortunately, not all the segmented irises are clear, suitable, and sharp enough for recognition. Some of these images are out of focus that result when the subject is far from the focus plane of the camera, others contain many noticeable interlacing lines (especially in regions close to the boundary) caused by eye motion, and in others, iris region is obstructed largely by eyelashes, and eyelids, these result when the subject opens his/her eye partially. Therefore, iris image quality assessment technique is required to select a suitable image of high quality.

Image quality assessment is an important topic for iris recognition as the performance of the recognition system is more related to the quality of the captured images. The efforts on image quality assessment are still limited. Some of these efforts are:

- Daugman (2002) performed focus assessment by measuring the total high frequency power in the 2D Fourier spectrum of an image. And seeking to maximize this quantity either by moving an active lens or by providing audio feedback to subjects to adjust their positions. Image passing a minimum focus criterion were then analyzed to find the iris.
- Zhang and Salganicoff analyzed the sharpness of the boundary between the pupil and the iris to determine whether an image is in focus (Ma *et al*, 2003).
- Ma *et al.*(2003) classified the low quality images to three classes:
  1. Out of focus images, which result when the subject is far from the focus plane of the camera.
  2. Images have little useful information, which are obstructed largely by eyelashes, and eyelids, these result when the subject opens his/her eye partially.
  3. Motion blurred images which result when the subject is moving during imaging.



Then they analyzed each class in the frequency domain by using 2D Fourier transform of two local regions in the iris ring, and the results were as follows:

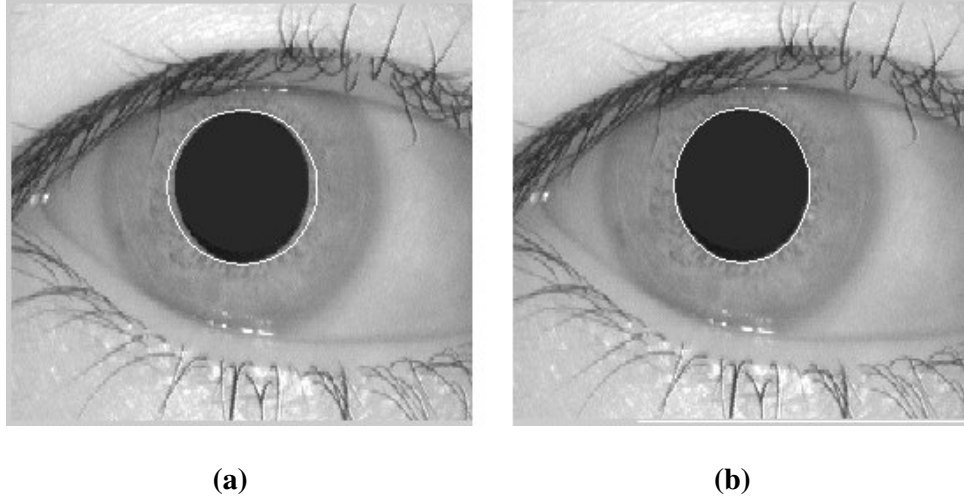
- The first class is greatly dominated by low frequency components.
- The second class contains significant middle and high frequency components.
- The third class contains vertical high frequency components in Fourier spectrum.
- Clear and properly focused iris image has relatively uniform frequency distribution.

According to these results they defined a quality descriptor based on 2D Fourier spectrum to distinguish between the low and high quality images.

#### 4. The Proposed Approach for Iris Segmentation and Iris Image Quality Assessment.

To propose an approach for iris segmentation and iris image quality assessment, many important factors must be considered. These factors are speed, reliability, and accuracy. In the proposed algorithm these factors were the main objectives.

Before presenting the steps of the proposed algorithm, an important point about how to model the inner boundary of the iris must be discussed. Both the inner boundary and the outer boundary of the iris can approximately be taken as two circles as in all previous efforts (Daugman, 1994), (Wildes *et al.*1996), (Zhu *et al.* 2000), (Lim *et al.*, 2001), (Tisse *et al.*2002), (Ma *et al.*, 2002, 2003), (Ali and Hassanien, 2003). But based on the fact that the pupil may not be exactly circular in shape and its deviation from the circle is a visible characteristic as discussed in part one, and to be more accurate, the proposed algorithm approximates the inner boundary as ellipse, as shown in Figure12. Modeling the inner boundary with ellipse has two advantages: the accuracy in the segmentation process and it may be useful when exploiting the changes in the pupil size for iris liveness detection as illustrated in part two. As these changes are very small, it may disappear when modeling the inner boundary with a circle as shown in Figure12.



**Figure 12: An example to show the accuracy in locating the inner boundary for**  
**(a) The inner boundary approximated as circle, (b) The inner boundary approximated as ellipse.**

The proposed algorithm for iris segmentation and image quality assessment is described and discussed in the following steps: *Note: all the weights, thresholds, filter sizes and margins in the following steps are chosen experimentally*

### **Step 1:**

Find approximately the inner boundary center  $(x_o, y_o)$  and its radiuses;  $r_x$  (pupil radius in the horizontal direction) and  $r_y$  (pupil radius in the vertical direction) as follows:

- Filter the image by 20x20 median filter.
- For each row in the filtered image, change the values of the pixels that have the minimum value in that row to one and the other pixels to zero.
- Filter the resultant image by 35x35 median filter.
- Get the row and column that has maximum number of white pixels. Then find the center  $(x_o, y_o)$  and the radiuses  $(r_x, r_y)$  as follows:

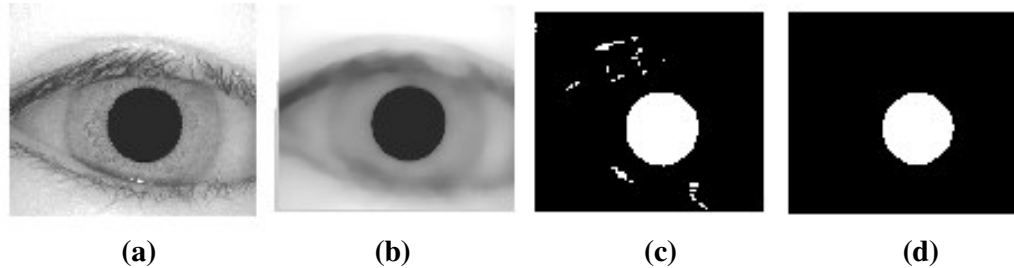
$x_o$  = The index of this column.

$y_o$  = The index of this row.

$r_x$  = The number of white pixels in this row / 2.

$r_y$  = The number of white pixels in this column / 2.

The results are shown in Figure 13.



**Figure 13: The results of finding an approximate pupil region from CASIA database.(a) The original image,(b) The result of step 1.a ,(c) The result of step 1.b,(d) The result of step 1.c.**

#### Discussion:

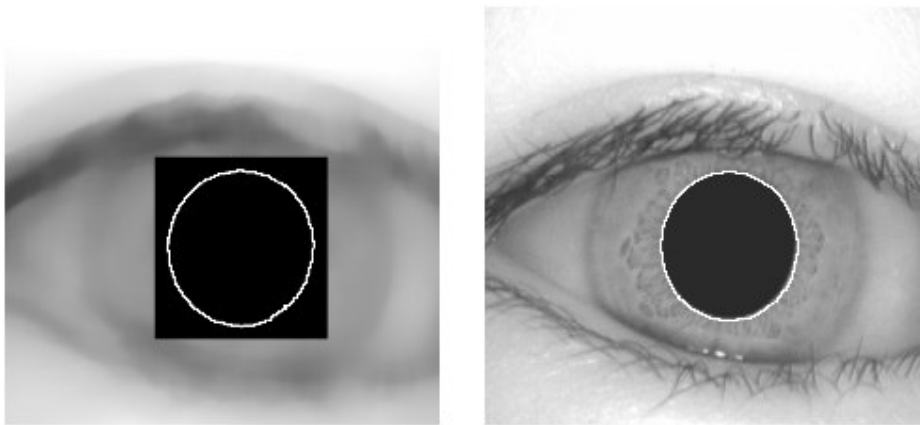
The main idea in this step is that the pupil is generally the darkest region in the image. And by noting that the minimum values in the rows that pass through the pupil region are located in the pupil region, because the regions that share the pupil with these rows are the sclera and parts of iris region which are very light regions. But in other rows the minimum values will be distributed separately in each row, and can be eliminated by a median filter.

Median filtering is a nonlinear operation often used in image processing to reduce "salt and pepper" noise. Median filtering is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges (Gonzalez and Woods, 2002), (Lewis, 1990). Each output pixel contains the median value in the  $m$ -by- $n$  neighborhood around the corresponding pixel in the input image. When we use it here, we get an image that consists generally of five homogenous regions as shown in Figure 13.b. These regions are the pupil, iris, eyelids, sclera, and the surrounding skin region. Among these regions the pupil is the darkest one, and this is what the main idea in this step is based on.

**Step 2:**

Find the exact parameters (center and radiuses) of the inner boundary as follows:

- a. In the rectangular region in the resultant image of step1.a, that is determined by the limits  $(x_o \pm (r_x + \Delta), y_o \pm (r_y + \Delta))$ , where  $\Delta$  is a small margin equal 10 pixels, find the edge map for the inner boundary using the sobel edge detector . This small margin is used to ensure that the selected region contains all the edge map of the inner boundary.
- b. Apply the elliptical Hough transform to find the exact center of the elliptical model of the inner boundary within a small square that is determined by the limits  $(x_o \pm r_x/10, y_o \pm r_y/10)$ , and to find the exact radiuses within a small margin  $(r_x \pm 3, r_y \pm 3)$ . It is expected that the parameters (center and radiuses) of the ellipse exist in these margins. And as these margins are small, a reduction in the computations will occur. The results are shown in Figure 14.



(a)

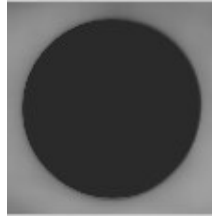
(b)

**Figure 14: The results of pupil boundary localization step in the iris segmentation process**

**(a)The result of step2.a, (b) The result of step2.b**

**Discussion:**

To detect the edge map of the inner boundary, we can use any edge detector as Sobel, Canny, Laplacian of Gaussian (LOG), Roberts, and Prewitt. All of these edge detectors give the same result, because the edge map of the inner boundary is sharp and clear as shown in the Figure 15.

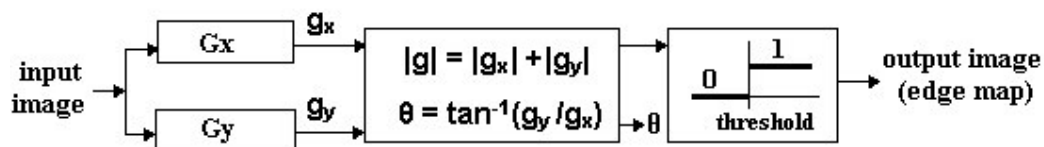


**Figure 15: The pupil region**

However, the Sobel edge detector is chosen, as it is a simple and popular edge detector. It finds edges by two steps; firstly, it approximates the derivative of the image in the vertical and horizontal directions by doing a convolution between the image and two kernels;  $G_x$  for the horizontal gradient and  $G_y$  for the vertical gradient, where  $G_x$  and  $G_y$  are defined as follows (Jain, 1989):

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Secondly, at each point in the image, it compares the sum of magnitudes of both convolution results of the horizontal and vertical gradients with a certain threshold. If the sum is larger than the threshold, this point is considered as edge point. This process is illustrated in Figure 16.



**Figure 16: Edge detection via gradient operators**



As mentioned before that the inner boundary is modeled with ellipse, which is represented as:

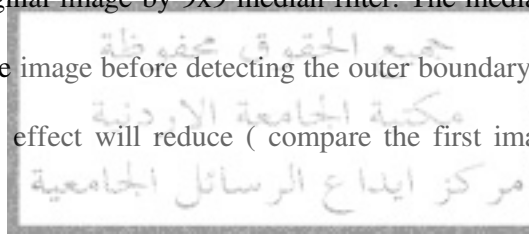
$$\frac{(x-x_i)^2}{b^2} + \frac{(y-y_i)^2}{a^2} = 1 \quad (3.4)$$

Where  $(x_i, y_i)$  is the center of the ellipse and  $b^2, a^2$  are the minor and major axis respectively. These parameters (center and axis) can be found by applying the elliptical Hough transform which was introduced in section 3.2.2.

### **Step 3:**

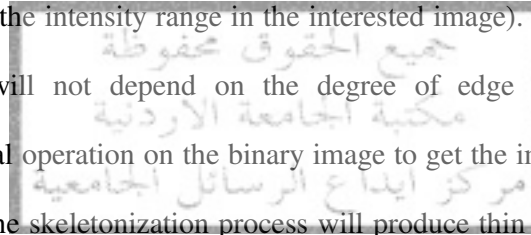
Find the exact parameters (center and radius) of the outer boundary as follows:

- Filter the original image by 9x9 median filter. The median filter is used in this step to enhance the image before detecting the outer boundary; by using the median filter the eyelashes effect will reduce ( compare the first image with the last image of Figure17) ,



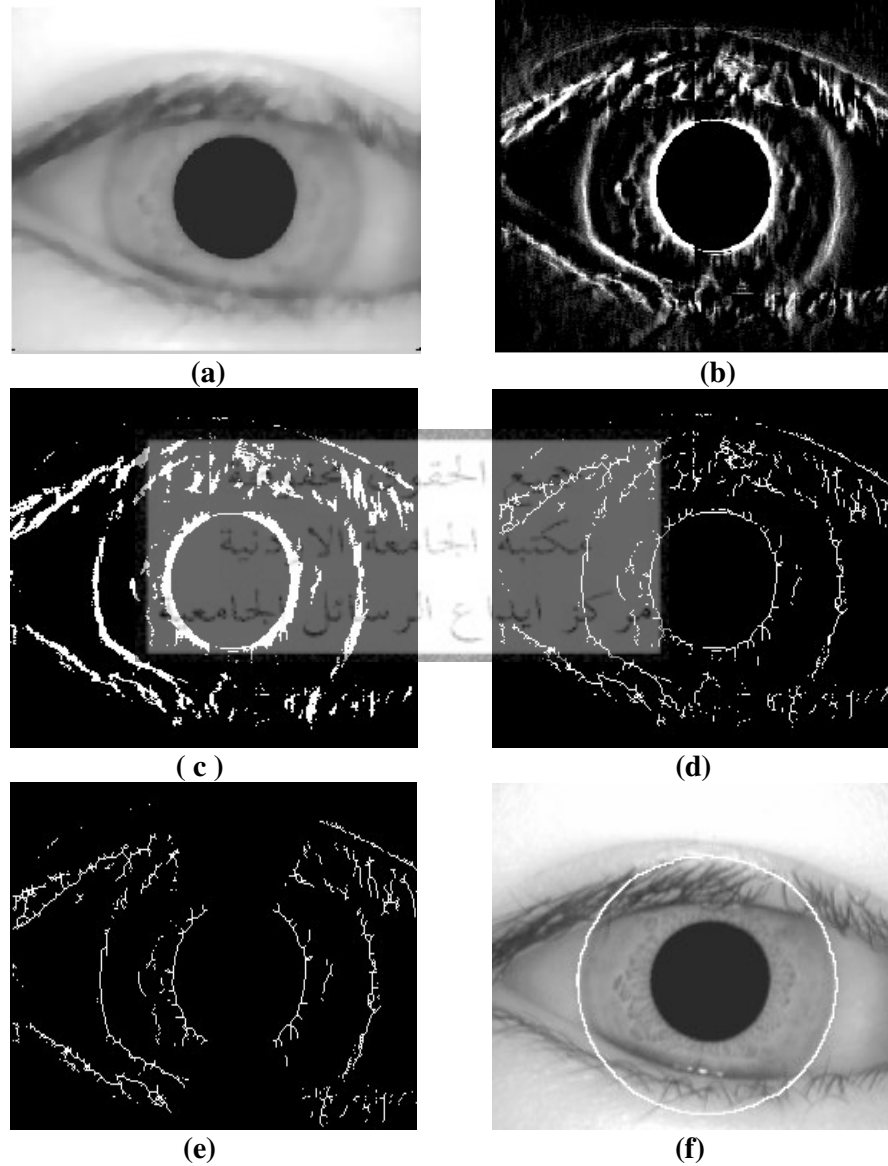
- Find the edge map of the image. The traditional method to find the edge map is to use one of the known edge detectors as Sobel, Canny, Laplacian of Gaussian (LOG), Roberts, and Prewitt. But the direct use for these detectors in this step has two obstructions; the first obstruction is that the outer boundary of the iris is a ramp edge (smooth edge) not a digital edge (sharp edge) as for the inner boundary as shown in Figure 12. And as this edge smoothing affects seriously the derivatives (first and second derivatives) that are used by the mentioned detectors to detect the edges a serious consideration must be taken before using these detectors (Gonzalez and Woods, 2002). The second obstruction is that these detectors need a certain threshold, but in practical it is not easy to find a fixed threshold that can be applied for all images; some of images have a clear outer boundary but others have not that, may be due to imaging conditions. So a proposed technique to reduce these issues

can be applied as follows: do a convolution between the 3-compass kernel multiplied also with the same weight. This weight is used here to focus the outer boundary and reveal it better as shown in Figure 17.b. Then convert the image to binary image by taking the mid point of the intensity range in the image as a threshold for the two binary levels (1 and 0). This binary conversion is important to apply the Hough transform in the step d, also the binary threshold in this conversion can be considered as an adaptive threshold (as this threshold is the mid point of the intensity range in the interested image). So the output of the binary conversion will not depend on the degree of edge smoothing. Then perform morphological operation on the binary image to get the image skeleton, which is the edge map. The skeletonization process will produce thin edges, so this will give the correct location of the outer boundary and reduce the computations of the Hough transform.



- c. Consider the center  $(x_o, y_o)$  as a reference point, and keep only the edge pixels that have angles between - consider the other edge pixels because it is usually affected by eyelids and eyelashes, also the kept pixels are enough to determine the outer boundary with some reduction in the computations.
- d. Apply the circular Hough transform to find the exact center and radius of the circular model of the outer boundary. The center is found within a small square that is determined by the limits  $(x_o \pm r_y/5, y_o \pm r_y/5)$ , these limits are verified experimentally; see part 6, section 6.6. And the exact radius is found within the expected iris radius range  $(> r_x + 10$  and  $> r_y + 10)$ , this range is chosen to avoid the

circular paths as collarette, which appears as a zigzag pattern. The results are shown in Figure 17.



**Figure 17:**The results of outer boundary localization step in the iris segmentation process for result of convolution with Compass kernels,(c) The binary image,(d)The result of step3.b (image skeleton),(e) The result of step3.c,(f) The result of step3.d

**Discussion:**

For the Compass operator, it measures gradients (by the convolution process) in a selected number of directions (Jain, 1989). Table 1 shows four different Compass gradients.

**Table 1: Different compass gradients (Jain, 1989)**

$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$ (Roberts)	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$ (Prewitt)
$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$ (Kirsch)	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ (Sobel)

An anti clockwise circular shift of the eight boundary elements of these masks gives a  $45^\circ$  rotation gradient direction. For example, the eight compass gradients corresponding to the first operator (that is used in this work) of Table 1 are illustrated in Table 2.

**Table 2: Compass gradients for different directions**

$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$ (N)	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$ (NE)	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$ (E)	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ (SE)
$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ (S)	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$ (SW)	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$ (W)	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$ (NW)

(Where: N: North, E: East, W: West, S: South)

Compass edge detector uses these kernels for edge detection by comparing at each point in the image the maximum of the gradients that result from these kernels with a certain threshold. If the maximum is larger than the threshold this point will be considered as edge point.

Here in this work the Compass edge detector is not used as it was discussed before. But we used the East and West Compass kernels weighted by a certain weight to focus the outer boundary parts that are located in the 3- - , respectively. These parts are the clearest parts of the outer boundary. Then we converted the resultant image to a binary image by a simple thresholding technique. And then morphological technique is used to get the image skeleton which is as edge map.

Morphology is a technique of image processing based on shapes (Jain, 1989), (Lewis, 1990). The value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, we can construct a morphological operation that is sensitive to specific shapes in the input image. Many morphological operations can be used to perform common image processing tasks, such as contrast enhancement, noise removal, thinning, skeletonization, filling, and segmentation. Here we apply the skeletonization process to get the skeleton of the binary image, which is as the edge map of the image.

Most morphological operations can be defined in terms of two basic operations, erosion and dilation (Jain, 1989). Suppose the object  $X$  and the structuring element  $B$  are represented as sets in two dimensional space. Let  $B_x$  denote the translation of  $B$  so that its origin is located at  $x$ . Then the erosion of  $X$  by  $B$  is defined as the set of all points  $x$  such that  $B_x$  is included in  $X$ , that is, Erosion:

$$X \ominus B = \{ x : B_x \subset X \} \quad (3.5)$$

Similarly, the dilation of  $X$  by  $B$  is defined as the set of all points  $x$  such that  $B_x$  hits  $X$ , that is, they have a nonempty intersection, Dilation:

$$X \oplus B = \{ x : B_x \cap X \neq \emptyset \} \quad (3.6)$$

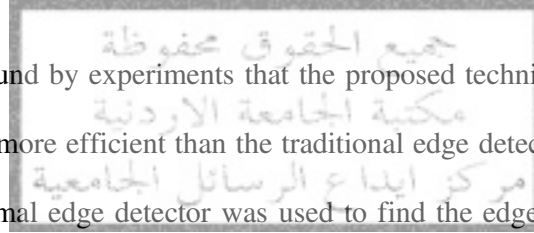
The skeleton is defined as the set of points whose distance from the nearest boundary is locally maximum (Jain, 1989). The skeleton of image  $A$  can be expressed in terms of erosions and openings as follows:

$$S(A) = \bigcap_{k=0}^K S_k(A) \quad (3.7)$$

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B \quad (3.8)$$

$$(A \ominus kB) \circ B = ((A \ominus kB) \ominus B) \oplus B, \quad \circ: \text{opening} \quad (3.9)$$

where  $B$  is a structuring element  $A \ominus kB$  indicates  $k$  successive erosions of  $A$  with the structuring element  $k$  times, and  $K$  is the last iterative step before  $A$  erodes to an empty set.



It was found by experiments that the proposed technique in step 3.b for finding the edge map is more efficient than the traditional edge detectors. Canny edge detector, which is an optimal edge detector was used to find the edge map, but the results were not good as for the proposed technique. There was a problem in the threshold of Canny edge detector, it was not easy to choose a certain threshold that can be applied to all images in the database. But for the proposed technique, it is insensitive to this problem as explained in step 3.b.

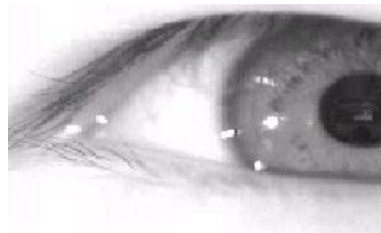
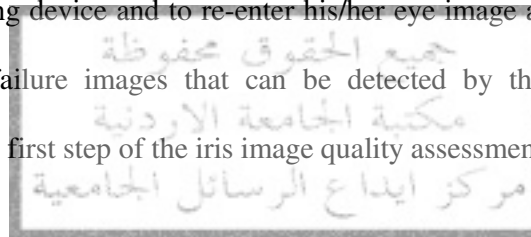
The threshold problem for Canny edge detector is not found only in this work, but also in Masek work (Masek,2003), who used the same database(CASIA database) function, which allowed for weighting of the gradients.

Here, the Hough transform is applied in a certain fast manner based on the fact that the distances from each point on the circumference of a circle to its center are equal; for each proposed center in the predetermined square region, the distance from it to each edge pixel is found, then the proposed center that has a large number of equal

distances are considered as the exact center of the outer boundary , and one of these equal distances is the radius of the outer boundary. This implementation is faster than the traditional implementation of the Hough transform which requires a range of proposed radiuses and centers and then voting in Hough space for these parameters is done through each edge point.

#### **Step 4:**

Check the results of step 2 and 3 by using the dimensions of the original images and the expected radiuses of the inner and outer boundaries. If there is any wrong result, stop the algorithm and display a message as feedback to the user asking him to center his/her eye to the imaging device and to re-enter his/her eye image again. Figure 18 shows one sample of the failure images that can be detected by this step. This step can be considered as the first step of the iris image quality assessment.



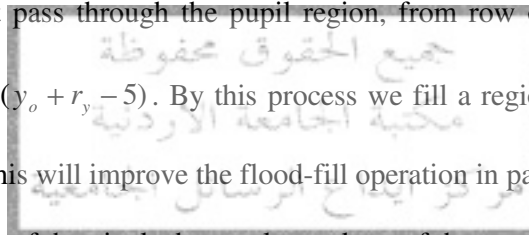
**Figure 18: One sample of failure images that can be detected by step one of the iris image quality assessment (Lim *et al.*, 2001)**

#### **Step 5:**

Isolate the eyelashes, reflections, and eyelids as follows:

- To isolate the eyelashes and reflections we use a simple thresholding technique. Where Eyelashes regions are the pixels that have values less than 100, and reflections regions are the pixels that have values larger than 240.
- To isolate the upper and lower eyelids we follow these steps:

- a. Convolve the upper side of the image (result of step3.a) with north compass kernel multiplied with a certain weight (experimentally, it equals 30), and the lower side with south compass kernel multiplied with the same weight.
- b. Convert the image to a binary image (using a large threshold about 0.8 of the full range of intensity in the image).
- c. Filter the image by 1x5 median filter then filter the result with 5x1 median filter. These filters will disconnect the separated eyelashes that connected to the eyelids. And this will improve the flood-fill operation in part (f) of this step.
- d. Find the complement of the resultant image, and change all pixel values to one for the rows that pass through the pupil region, from row of index  $(y_o - r_y + 5)$  to the row of index  $(y_o + r_y - 5)$ . By this process we fill a region that we are sure it must be filled, so this will improve the flood-fill operation in part (f) of this step.
- e. Put the values of the pixels that are located out of the outer boundary to zero.
- f. Perform a flood-fill operation., and find the upper and lower edge pixels( searching starts from center row of the pupil in the white region, see Figure19.f )
- g. Fit a polynomial of power four to the edge pixels (upper and lower separately). Then if the peak of this polynomial is very near to the inner boundary by a distance less than 0.25 of the distance between the outer and inner boundaries, fit a polynomial of power two instead of power four. It is found by experiments that these models for the eyelids are very suitable. The results shown in Figure 19.





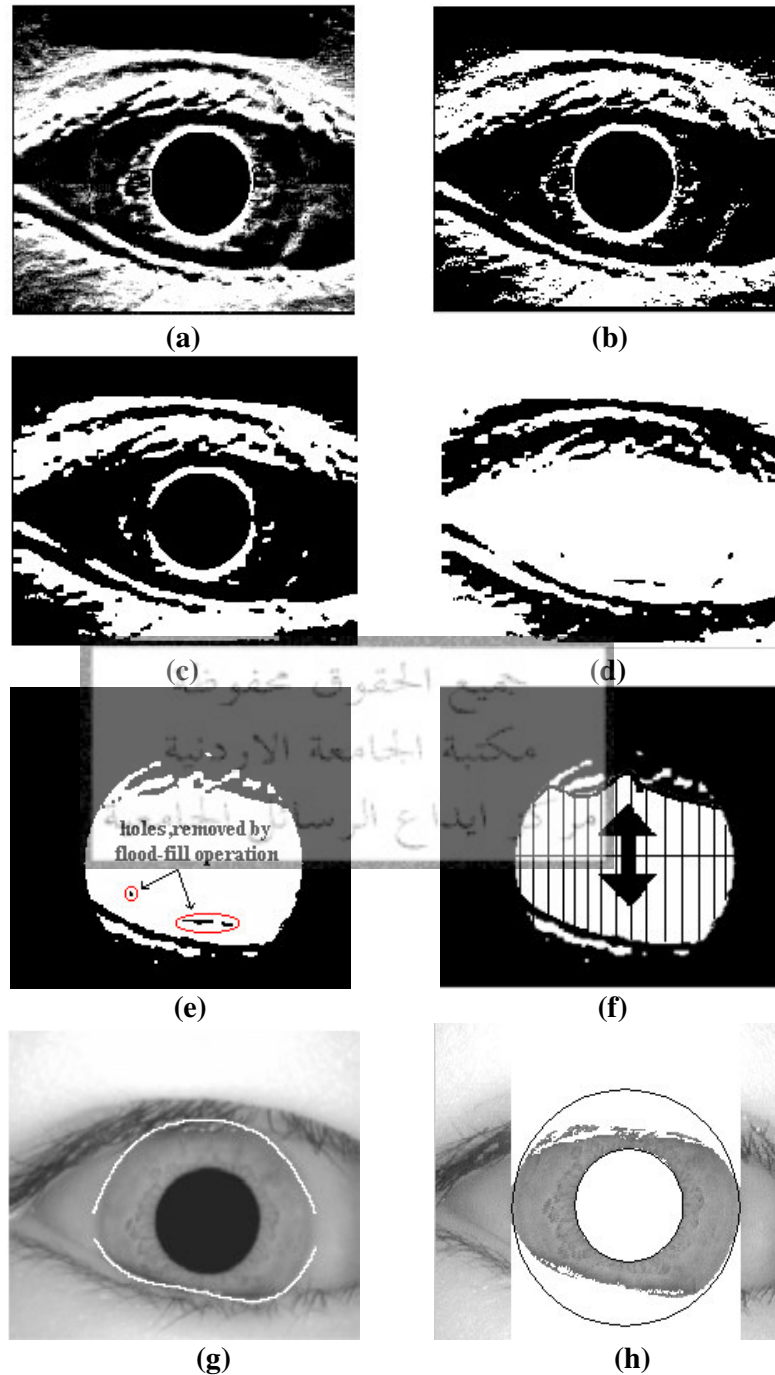


Figure 19: The results of eyelashes and eyelids isolation step in the iris segmentation process for step5.a,(b) The result of step5.b,(c)The result of step5.c,(d) The result of step5.d, (e) The result of step5.e, (f)The result of step5.f,(g) The result of step 5.g,(h) The final result of eyelids and eyelashes isolation.

### Discussion:

Here flood-fill operation is used, which is one of morphological image processing (Gonzalez and Woods, 2002). For binary images, flood-fill operation changes connected background pixels (0s) to foreground pixels (1s), stopping when it reaches object boundaries. For grayscale images, it brings the intensity values of dark areas that are surrounded by lighter areas up to the same intensity level as surrounding pixels. So it removes regional minima that are not connected to the image border. A common use of the flood-fill operation is to fill "holes" in images. For example, suppose you have an image, binary or grayscale, in which the foreground objects represent spheres. In the image, these objects should appear as disks, but instead are donut shaped because of reflections in the original photograph. Before doing any further processing of the image, you may want to first fill in the "donut holes" using flood-fill operation. Here flood-fill operation is used to fill the irrelevant holes that appear in the image as shown in Figure 19.e. These holes must be removed because they affect the searching process for finding the upper and lower edges as shown in Figure 19.f . Filling holes in image  $A$  can be done as follows(Gonzalez and Woods, 2002):

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k$$

Where  $X_0$  = initial point inside the boundary,  $B$  is the structuring element, and  $A^c$  is the complement of  $A$ . The algorithm terminates at iteration step  $k$  if  $X_k = X_{k-1}$ . The final result is the union of  $X_k$  and  $A$  .

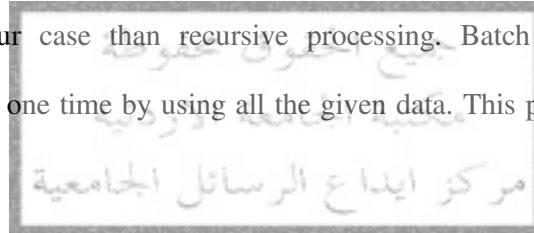
The process of fitting a polynomial to given data points requires an estimation of the coefficients of the polynomial using a method of estimation as a least-squares estimation method or it is called least-squares curve fitting( Melsa and Cohn,1978).

Least-squares estimation is based on arranging the given data in a linear model called a linear observation model, which is defined as

$$Z = H\theta + n \quad (3.11)$$

Where  $Z$  is called the measurement vector,  $H$  is called the observation matrix,  $\theta$  is called the parameter vector, and  $n$  is called the measurement noise vector.

Estimation of the parameter vector  $\hat{\theta}$ , which represents the polynomial coefficients in this case can be found by batch or recursive processing. Recursive processing requires an initial values then it estimates the parameter vector for each additional new data point. Batch processing is simpler, faster, and more suitable to implement in our case than recursive processing. Batch processing estimates the parameter vector one time by using all the given data. This process can be summarized as follows:



$$\hat{\theta} = (H^T H)^{-1} H^T Z \quad (3.12)$$

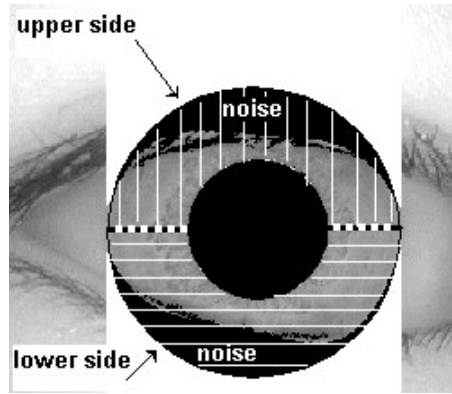
Where  $H^T$  is the transpose of matrix  $H$ .

#### **Step 6:**

Assess the quality of the iris image by calculating the noise percent in the upper and lower sides of the segmented iris region, separately. The noise percent is calculated as follows:

$$\text{noise \%} = (\text{noise area} / \text{interested area}) \times 100\% \quad (3.13)$$

Where the interested area is the upper or the lower side of the iris region as shown in Figure 20.

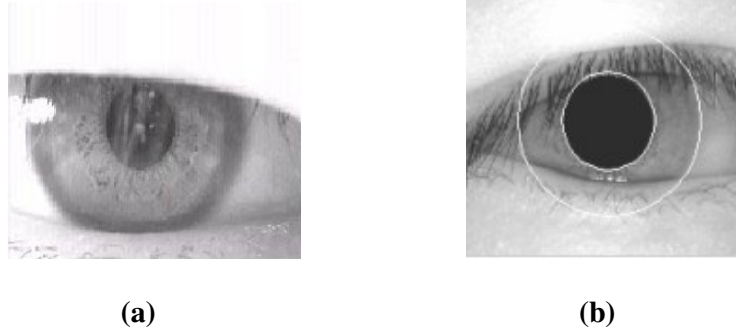


**Figure 20: Illustration of calculation the CASIA database**

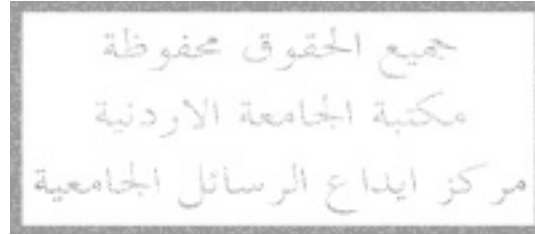
Then compare these percents to a certain threshold (we used 50%, as will be illustrated in part 6). If one or both of the noise percents exceed this threshold, the algorithm considers that there is no sufficient region for recognition as shown in Figure21, and return a message to the user asking him to open his/her eye morely and to re-enter his/her eye image a gain. This step can be considered as the second step of image quality assessment.

To ensure always both sides have a sufficient useful information, the calculations and comparing of the noise percents( upper and lower) are done separately; suppose that for a person an iris image is captured with a high noise percent in the lower side equals to 90 % and a low noise percent in the upper side equals to 5 %. In other session the same iris is captured with a low noise percent in the lower side equals to 5%. And a high noise percent in the upper side equals to 90 %. So there is only 20% (10 % from each side) of the iris region used for comparing the two irises, and this region is not sufficient for comparing. But if the both percent are considered as a one percent the results in both sessions are the same and equal to 47.5 % ; consider the entire iris area is  $I$ , so the lower side area = the upper side area =  $0.5*I$ . In the first session the area of the noise in the upper side =  $0.05*(0.5*I)$ , and the noise area in the lower side =  $0.90*(0.5*I)$ . So the total noise area =  $0.95*0.5*I = 0.475*I$ . So the total noise percent =

$(0.475 * I / I) * 100\% = 47.5 \%$ . The same result will be for the second session. And this result may be an accepted noise percent. So a wrong comparison will be done.



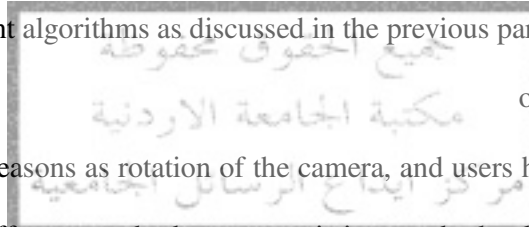
**Figure 21:** Samples of failure images that have not sufficient region for recognition. (a) Failure image (Lim *et al.*, 2001), (b) image 101\_1\_3 from CASIA database.



## Iris Normalization

### 1. Introduction

For any successful iris recognition system, it must be able to solve the problems of image shift, rotation, scaling, and noise. For shift problem, the iris region may be located in different locations in the acquired eye image due to the eyeball movements or the person movements. For noise problem, the iris image may be affected by different types of noise as eyelashes, eyelids, specular reflections, and images may be out of focus or a motion blurred images. These problems (shift and noise problems) can be solved by using a good design imaging system with good iris localization and image quality assessment algorithms as discussed in the previous parts.



on may be rotated, this can be due to many reasons as rotation of the camera, and users head tilts. This problem can be solved by different methods; most existing methods shift the iris template by a different rotation angles during matching (Daugman,1994)(Tisse *et al.*, 2002), other methods solve this problem by registering the input images with different rotation angles before matching process(Ma *et al.*2002,2003).

For scaling problem, the iris region size may differ between the images of different persons and also may differ between the images of the same person, this is due to many reasons as the stretching of the iris caused by pupil dilation at different levels of illumination, and variations of the imaging distance. A certain normalization process can solve this problem. This normalization process must produce iris regions, which have the same constant dimensions for all persons, so that two images of the same iris under different conditions will have characteristic features at the same spatial location.

In the next section, some of well known iris normalization techniques are presented. The last section presents an efficient implementation for the common and

technique is used in our work.

## 2. Iris Normalization Techniques

The most popular techniques for iris normalization are: model, Image registration, and Virtual concentric circles.

Daugman (1994) proposed an iris normalization technique, which is called

$(x, y)$  to polar image coordinates  $(r, \theta)$  according to

$$x(r, \theta) = (1-r)x_p(\theta) + rx_l(\theta) \quad (4.1)$$

$$y(r, \theta) = (1-r)y_p(\theta) + ry_l(\theta) \quad (4.2)$$

Where  $r$  lies on  $[0,1]$  and  $\theta$  lies on  $[0,2\pi]$ , while  $(x_p(\theta), y_p(\theta))$  and  $(x_l(\theta), y_l(\theta))$  are the coordinates of the pupillary (inner) and limbic (outer) boundaries in the direction  $\theta$  respectively. Figure 22 illustrates this technique.

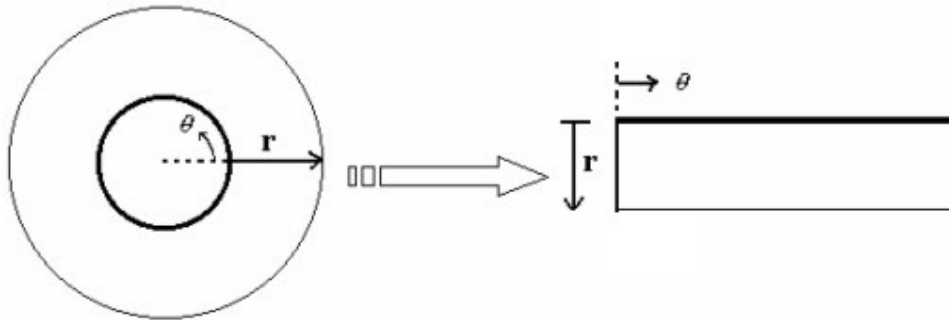


Figure 22

et model for iris normalization

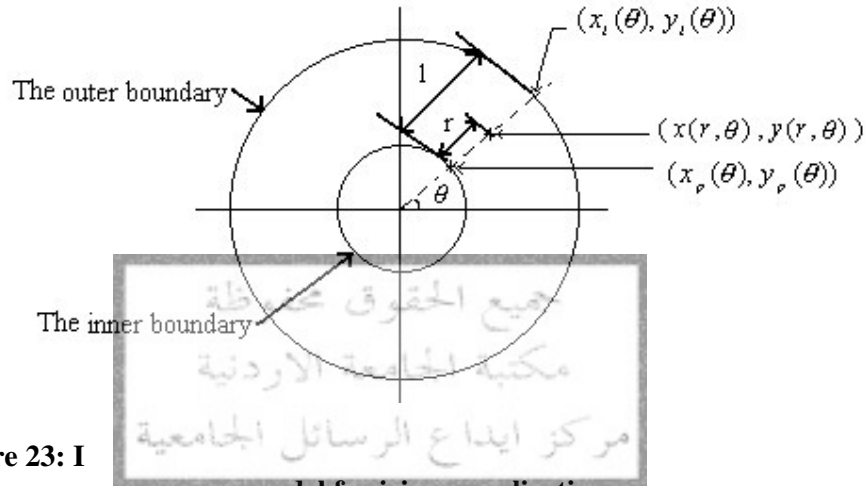
to Figure 23. From this figure we can derive the following at any  $\theta$  and  $r$ :

$$\cos(\theta) = \frac{x_l(\theta) - x_p(\theta)}{1} = \frac{x(r, \theta) - x_p(\theta)}{r} \Rightarrow x(r, \theta) = (1-r)x_p(\theta) + rx_l(\theta) \quad (4.3)$$

$$\sin(\theta) = \frac{y_l(\theta) - y_p(\theta)}{1} = \frac{y(r, \theta) - y_p(\theta)}{r} \Rightarrow y(r, \theta) = (1-r)y_p(\theta) + ry_l(\theta) \quad (4.4)$$

Now, it is

sheet model (Eq.4.1 and Eq.4.2) are same.



**Figure 23: I**  
**model for iris normalization.**

## 2.2 Image Registration

Image registration is the process of aligning two or more images of the same scene. Typically, one image, called the base image, is considered the reference to which the other images, called input images, are compared. The object of image registration is to bring the input image into alignment with the base image by applying a spatial transformation to the input image.

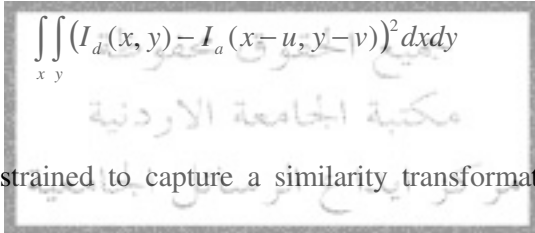
A spatial transformation maps locations in one image to new locations in another image. Determining the parameters of the spatial transformation needed to bring the images into alignment is key to the image registration process.

Image registration is often used as a preliminary step in other image processing applications. For example, we can use image registration to align satellite images of the



earth's surface. After registration, we can compare features in the images to see how a river has migrated, how an area is flooded.

Wildes *et al.* (1996) employed the image registration technique to compensate for both scaling and rotation. This approach geometrically warps a newly acquired image  $I_a(x, y)$  into alignment with a selected database image  $I_d(x, y)$  according to mapping function  $(u(x, y), v(x, y))$  such that the image intensity value of the new image are close to that of the corresponding points in the reference image. So the mapping function  $(u(x, y), v(x, y))$  must be chosen to minimize

$$\int \int_{x y} (I_d(x, y) - I_a(x-u, y-v))^2 dx dy \quad (4.5)$$


While being constrained to capture a similarity transformation of image coordinates  $(x, y)$  to  $(x', y')$ , that is

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - sR(\Phi) \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.6)$$

Where  $s$  is a scaling factor and  $R(\Phi)$  is a matrix representing rotation by  $\Phi$ . In implementation, given a pair of iris images  $I_a$  and  $I_d$ , the warping parameters  $s$  and  $\Phi$ , are recovered via an iterative minimization procedure. This technique works differently matching time, rather than performing normalization and saving the result for later comparisons. So this technique is not suitable for identification mode, only it can be used for verification mode.

### 2.3 Virtual Concentric Circles

Boles and Boashash (1998) applied a specific iris normalization technique. To compare two irises using this technique, firstly, one of these irises is considered as a reference iris and the other is resized to the size of the reference iris. Then concentric circular samples centered at pupil center are extracted. Then the sample length is fixed by using a certain normalization resolution, so that the number of data points extracted from each iris is the same. Then these data points are processed to extract features from it. This system works differently to the other techniques, since the normalization at matching time, rather than performing normalization and saving the result for later comparisons.

### 3. Efficient Implementation

After studying the previous techniques for iris normalization in the previous



feature extraction; most of feature extraction methods can be applied easily after this normalization technique. And also this technique is popular; it was used in many works: (Daugman,1994),(Zhu *et al.*,2000), (Lim *et al.*,2001), (Tisse *et al.*,2002), ( Ma *et al.*, 2002,2003) and (Ali and Hassanien,2003).

ization, an important issue must be considered, which is the pupil can be non-concentric to the iris. So when applying Equations (4.1) and (4.2) and considering the center of the pupil as the reference point, it is not easy to find the values of  $(x_i(\theta), y_i(\theta))$ , see Figure 23.

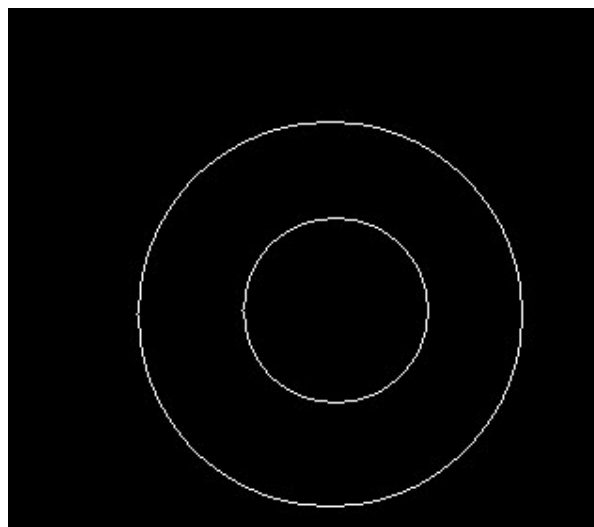
A proposed approach to implement this technique considers this issue, and solves it correctly. This implementation is illustrated in the following steps:

- a. By using the results of segmentation stage, construct an iris mask as shown in Figure 24.



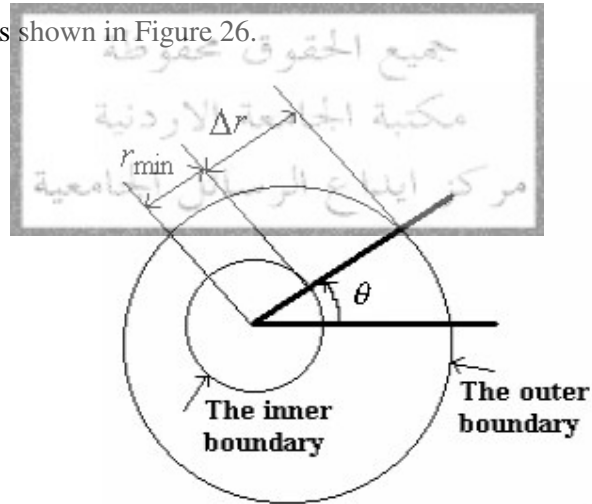
**Figure 24: The**

- b. Find the edge map of the mask image by using Canny edge detector (to ensure that the edge points are connected), the result as shown in Figure 25.



**Figure 25: The edge map of the iris mask.**

- c. Find the angles of each edge point from the horizontal line that passes through the pupil center (consider the pupil center as the reference point).
- d. Divide  $\theta$  range  $[0, 2\pi]$ , to the wanted angular resolution (the length of the resulting normalized iris region, see Figure 22), so a number of  $\theta$  s equals to the wanted angular dimension and in the range  $[0, 2\pi]$  are obtained.
- e. For each  $\theta$ , search for the edge points that have angles equal or approximately equal to this  $\theta$ . Then find the radial distances between these points and the pupil center, then take the difference between the maximum and the minimum of these radial distances. Record the minimum radial distance  $r_{\min}$  and the difference radial distance  $\Delta r$  as shown in Figure 26.



**Figure 26**

**finding  $r_{\min}$  and  $\Delta r$  for each  $\theta$ . Note that the pupil can be non-concentric to the iris.**

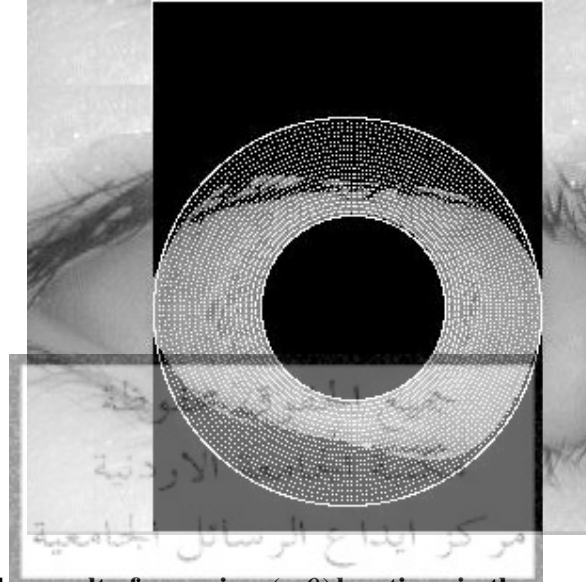
- f. By using the following Equations

$$x(r, \theta) = [r \times \Delta r(\theta) + r_{\min}(\theta)] \cos(\theta) \quad (4.7)$$

$$y(r, \theta) = [r \times \Delta r(\theta) + r_{\min}(\theta)] \sin(\theta) \quad (4.8)$$

find the locations  $(x, y)$  in the iris image for the corresponding points in the normalized iris region that is determined by  $(r, \theta)$ , where the range of  $r$  is  $[0, 1]$  and divided to the

number of wanted radial resolution (the width of the normalized iris region, see Figure 22), and the range of  $\theta$  is  $[0, 2\pi]$ , and divided to the number of wanted angular resolution (the length of the normalized iris region, see Figure 22). These locations in the iris image will be as shown in Figure 27.



**Figure 27: The result of mapping  $(r, \theta)$  locations in the normalized iris region to  $(x, y)$  locations in the origin iris region.**

- g. Find the intensities values of these locations  $(x, y)$  by using linear interpolation that uses the gray levels of the four nearest neighbors to calculate the gray level of the non-integer coordinates  $(x, y)$  by using the relationship (Gonzalez and Woods, 2002):

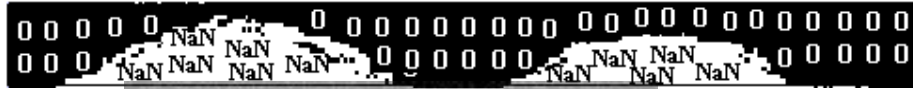
$$f(x, y) = ax + by + cxy + d \quad (4.9)$$

where  $f(x, y)$  is the gray level value, and the four coefficients are easily determined from the four equations in four unknowns that can be written using the four known neighbors of  $(x, y)$ . Then reflect these intensities to the corresponding  $(r, \theta)$  locations to get the normalized iris as shown in Figure 28. and change the values of the noise regions to the average value of the free noise regions as shown in Figure 28; this reduce the effect of noise in the feature extraction process.

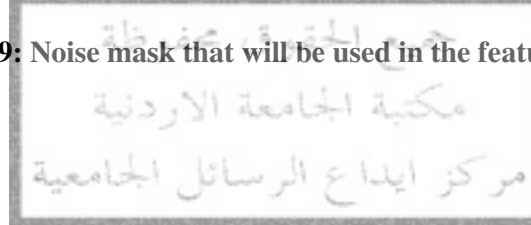


**Figure 28: The iris normalization result with noise region intensities equal to the average of the free noise region.**

- h. Find another array called noise mask array that determines the noise regions with NaN, where NaN is the IEEE arithmetic representation for Not-a-Number, and it is used here to represent the noise. And other regions are represented with 0. This will be used in the feature extraction process. The result as shown in Figure 29.



**Figure 29: Noise mask that will be used in the feature extraction process**



## Feature Extraction, Encoding, and Matching

### 1. Introduction

After iris normalization process, a fixed size iris pattern having a particularly interesting structure and provides abundant texture information is obtained. The next step is to extract the significant information from the iris pattern and encode it to get an iris template or what is called feature vector.

After an iris template (feature vector) is produced by the feature extraction and encoding processes, a matching metric is required to measure the similarity between two iris templates. After the matching process we can distinguish between two ranges of matching values; the first range contains the matching values of the same eyes and this range is called intra-class comparisons. And the second range contains the matching values of different eyes and this range is called inter-class comparisons.

In the next two sections, some of the well known techniques for feature extraction, encoding, and matching are presented. In the last section an efficient and fast proposed approach for feature extraction, encoding, and matching is presented.

### 2. Feature Extraction and Encoding Techniques

Some of the most well known techniques for feature extraction and encoding are: Gabor filters, log-Gabor filters, Laplacian of Gaussian filters, circular symmetric filters, Haar wavelet, Zero-crossing of the wavelet transform, multichannel Gabor filters and wavelet transform.

## 2.1 Gabor Filters

In recent years, Gabor filters based methods have been widely used in computer vision, especially for texture analysis. It is shown that the functional form of Gabor filters conforms closely to the receptive profiles of simple cortical cells, and Gabor filtering is an effective scheme for image representation (Ma. *et al.*,2002)

S organization of the primary visual cortex in mammalian brains, an enormous amount of experimental and theoretical research has greatly advanced our understanding of this area and the response properties of its cells. On the theoretical side, an important insight has been advanced by Marcelja and Daugman that simple cells in the visual cortex can be modeled by Gabor functions. Gabor elementary functions are Gaussians modulated by sinusoidal functions. The original Gabor elementary functions, in the form proposed by Gabor, are generated with a fixed Gaussian while the frequency of the modulating wave varies. Daugman generalized the Gabor function to two dimensions (2D) form to model the orientation-selective simple cells (Lee, 1996).

Daugman (1994) used a 2D version of Gabor filters in order to encode iris pattern. The 2D Gabor filter over the image domain  $(x, y)$  is represented as

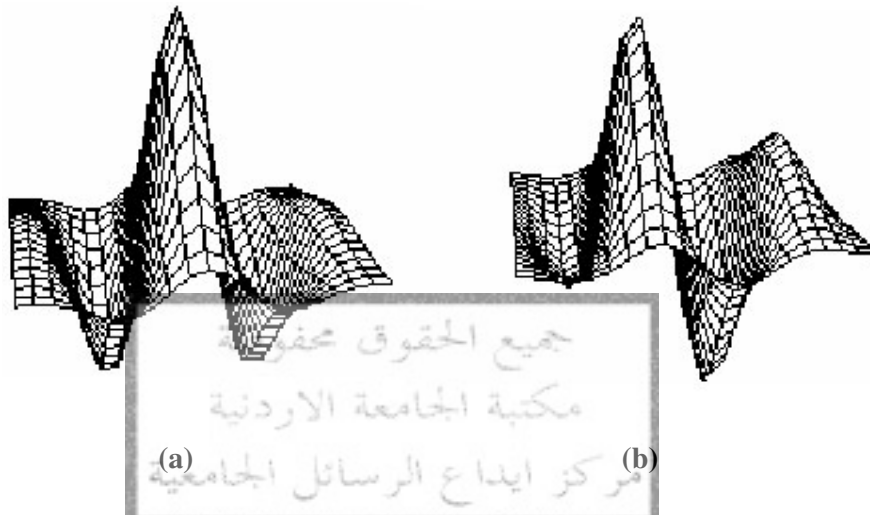
$$G(x, y) = \frac{1}{2\pi\sigma\beta} e^{-\pi \left[ \frac{(x-x_o)^2}{\sigma^2} + \frac{(y-y_o)^2}{\beta^2} \right]} e^{i[\zeta_o x + \nu_o y]} \quad (5.1)$$

Where  $(x_o, y_o)$  is the center of the filter in the spatial domain and  $(\zeta_o, \nu_o)$  is the center of the filter in the frequency domain.  $\sigma$  and  $\beta$  are the standard deviations of the elliptical Gaussian along  $x$  and  $y$  respectively.

From equation (5.1) the Gabor filter is constructed by modulating a sine/cosine wave with an elliptical Gaussian. The center frequency of the filter is specified by the frequency of the sine/cosine wave, and the bandwidth of the filter is specified by the

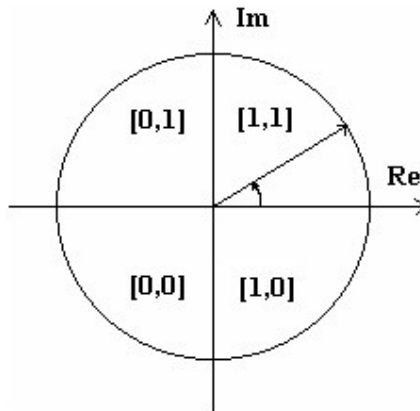


width of the Gaussian. In other words, Gabor filter consists of two components; the real component specified by a cosine modulated by Gaussian and it is known as even symmetric component, and an imaginary component specified by a sine modulated by a Gaussian and it is known as odd symmetric component as show in Figure 30.



**Figure 30: (a) The even symmetric component of the Gabor filter (b) The odd symmetric component of the Gabor filter**

ses its representation by quantizing the local phase angle in to four levels according to whether the real and imaginary filter outputs are positive or negative . These four levels are represented using two bits of data, so each pixel in the normalized iris pattern corresponds to two bits of data in the iris template as shown in Figure 31. A total of 2,048 bits are calculated for the template, and an equal number of masking bits are generated in order to mask out corrupted regions within the iris. This creates a compact 256-byte template, which allows for efficient storage and comparison of irises (Daugman, 1994, 2002).



**Figure 31: Phase Quantization process used to encode iris patterns.**

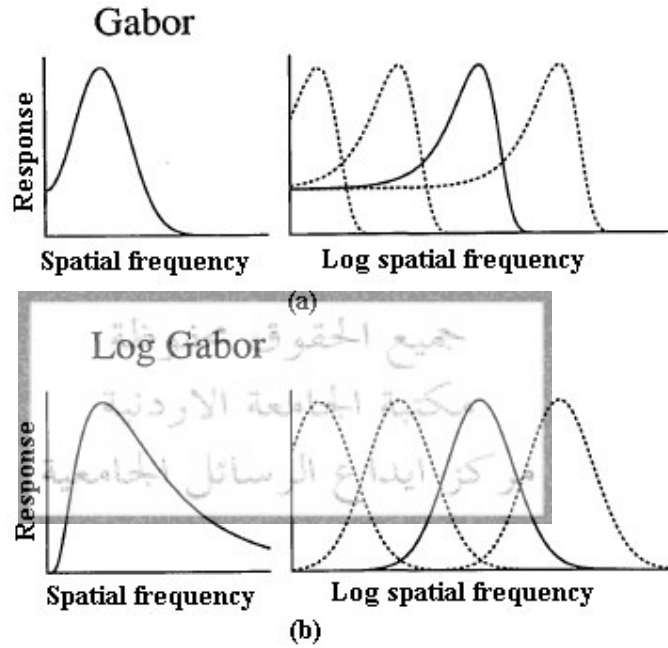
## 2.2 Log-Gabor Filters

Hawken and Parker suggested that the Gabor function fails to capture the precise form of the spatial-frequency tuning curves in monkey cortical cells (Field, 1987). Gabor functions miss in this fit primarily because they fail to capture the relative symmetry of the tuning curves on a log axis. The log axis is the standard method for representing the spatial-frequency response of visual neurons. Figure 32.(a) shows a one-dimensional representation of the spacing of the Gabor function on a log axis. With the bandwidths constant in octaves, the Gabor functions over-represent the low frequencies. Furthermore, with a  $1/f$  falloff in the amplitude, most of the input to the Gabor will be provided by the low frequency tails of the functions. This will, in essence, produce a correlated and redundant response to the low frequencies .

So Gabor filter has a disadvantage ,which is the DC component when the bandwidth is larger than one octave (an octave is a frequency change by a factor of 2 ), so its bandwidth is limited approximately to one octave ,and this indicates that the Gabor filters are not optimal if one is seeking broad spectral information with maximal spatial localization .

Field (1987) proposed log-Gabor filters as an alternative to the Gabor filters, which solves the disadvantage of the Gabor filters as shown in Figure 32.(b). For

bandwidths of less than 1 octave, the two functions produce similar results. At bandwidths of larger than 1 octave, the redundancy at the low frequencies becomes apparent for Gabor filters, but the frequency response of the log Gabor filters permits a more compact representation than the Gabor filters when the bandwidths are larger than 1 octave.



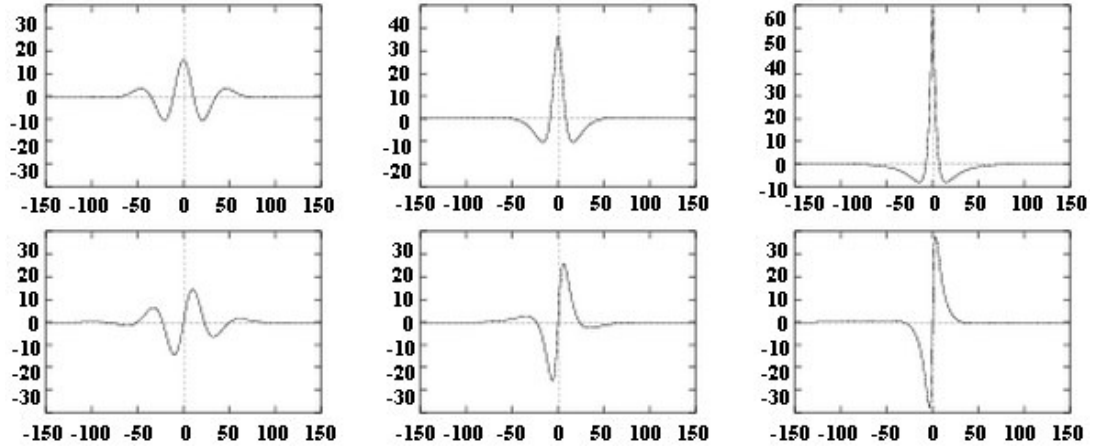
**Figure 32: Gabor and log-Gabor transfer functions viewed on both linear and logarithmic frequency scales.**

The frequency response of the log-Gabor filter is given as:

$$G(f) = \exp\left(\frac{-(\log(f/f_o))^2}{2(\log(\sigma/f_o))^2}\right) \quad (5.2)$$

Where  $f_o$  is the center frequency, and  $\sigma/f_o$  gives the bandwidth of the filter.

The shape of the Log-Gabor function in the spatial domain is similar to Gabor functions though their shape becomes much sharper as the bandwidth is increased as shown in Figure 33. The shapes of Gabor and log-Gabor functions are almost identical for bandwidths less than one octave.



**Figure 33: Three pairs (even and odd) of log Gabor wavelets all tuned to the same frequency, but having different bandwidths.**

### 2.3 Laplacian of Gaussian Filters

Wildes *et al.* (1996) constructed a Laplacian pyramid with four different resolution levels of filtered iris image in order to generate a compact iris template.

These iris images are filtered by using Laplacian of Gaussian filters, which are given as

$$\nabla^2 G = -\frac{1}{\pi\sigma^4} \left( 1 - \frac{\rho^2}{2\sigma^2} \right) e^{-\rho^2/2\sigma^2} \quad (5.3)$$

Where  $\rho$  is the standard deviation of the Gaussian. Construction of Laplacian pyramid is done using the difference between the filtered images, and down / up sampling technique.

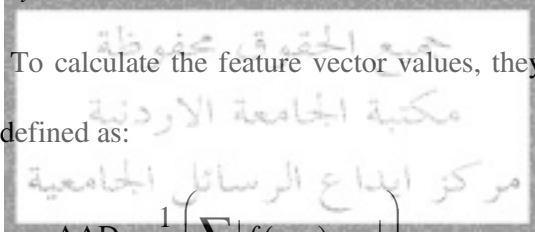
## 2.4 Circular Symmetric Filters

Ma. *et al.* (2002) extracted features from the rectangular normalized iris region by using a circular symmetric sinusoidal function which is represented in spatial domain as:

$$G(x, y, f) = \frac{1}{2\pi\delta_x\delta_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\delta_x^2} + \frac{y^2}{\delta_y^2}\right)\right] M(x, y, f) \quad (5.4)$$

$$M(x, y, f) = \cos\left[2\pi f\left(\sqrt{x^2 + y^2}\right)\right] \quad (5.5)$$

Where  $M(x, y, f)$  is the modulating function,  $f$  is the frequency of the sinusoidal function,  $\delta_x$  and  $\delta_y$  are the space constants of the Gaussian envelope along the  $x$  and  $y$  axis respectively. To calculate the feature vector values, they use the average absolute deviation (AAD) defined as:



$$AAD = \frac{1}{N} \left( \sum_N |f(x, y) - m| \right) \quad (5.6)$$

Where  $N$  is the number of pixels in the image block,  $m$  is the mean, and  $f(x, y)$  is the value at point  $(x, y)$ .

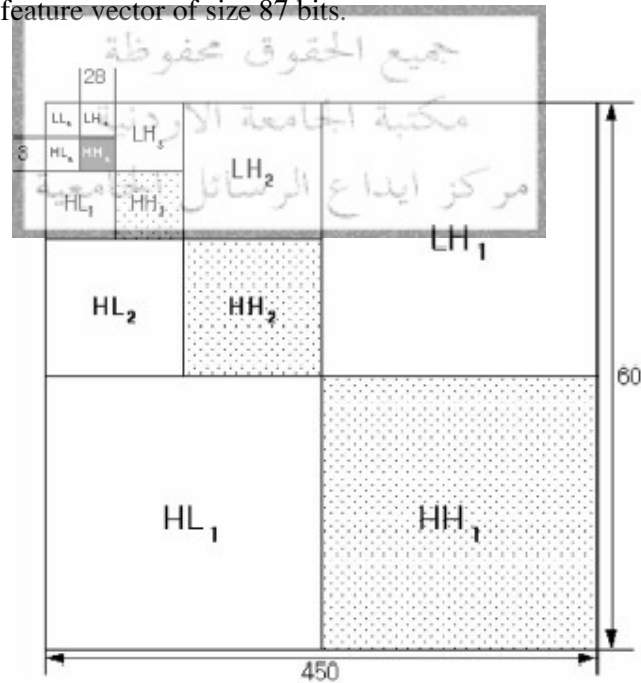
Ma *et al.* (2003) used the same method for feature extraction, but the feature values are the mean  $m$  and the average absolute deviation  $\sigma$  of the magnitude of each filtered block were defined as:

$$m = \frac{1}{n} \sum_w |F_i(x, y)|, \quad \sigma = \frac{1}{n} \sum_w ||F_i(x, y) - m| \quad (5.7)$$

Where  $w$  is size of the block in the filtered image,  $n$  is the number of pixels in the block  $w$ , and  $m$  is the mean of the block  $w$ . these feature values are arranged to form a 1D feature vector:  $V = [m_1, \sigma_1, m_2, \sigma_2, \dots, m_{768}, \sigma_{768}]^T$

## 2.5 Haar Wavelet

Lim *et al.* (2001) applied Haar wavelet transform to the rectangular normalized iris region (whose size equals to 60x450 pixels) four times in order to get the HH sub image of the high-pass filter of the fourth transform, which has a size of 3x28 as shown in Figure 34. This sub-image has real values between -1.0 and 1.0. These values are quantized into binary value by simply converting the positive value into 1 and the negative value into 0. Therefore we can get from this sub-image 84 values. In addition to these values, three values are calculated from each average value for the three remaining high-pass filter areas (HH<sub>1</sub>, HH<sub>2</sub>, and HH<sub>3</sub>). Therefore the iris image is represented by a feature vector of size 87 bits.



**Figure 34: Conceptual diagram for organizing a feature vector using Haar wavelet transform**

## 2.6 Zero-crossing of the Wavelet Transform

Boles and Boashash (1998) used the Zero-crossing of the wavelet transform to represent the normalized virtual concentric circles that have the features of the iris texture in closed rings. The wavelet transform decomposes the signals (virtual circles) in to a set of signals at different resolution levels. To reduce the noise effect on the finer resolution levels, and also to reduce the number of computations required, only a few low resolution levels were taken.

To illustrate the zero-crossing representation, let us denote the zero-crossing representation of a signature  $f$  at a particular resolution level  $j$  by  $Z_j f$ . In addition, let  $P_j = \{ p_j(r) ; r = 1, 2, \dots, R_j \}$  be a set containing the locations of zero-crossing points at level  $j$ , where  $R_j$  is the number of zero crossings of the representation at this level. Then, the representation  $Z_j f$  can be uniquely expressed in the form of a set of ordered complex numbers whose imaginary  $[\rho_j]_f$  and real  $[\mu_j]_f$  parts indicate the zero-crossing position and magnitude of  $Z_j f$  between two adjacent zero-crossing points, respectively.

## 2.7 Multichannel Gabor Filter and Wavelet Transform

Zhu *et al.* (2000) used two texture analysis methods to extract features from the normalized block of iris image. These methods are Multichannel Gabor filtering and 2-D wavelet transform. They used two types of Gabor filters (even and odd), which can be represented as follows:

$$h_e(x, y) = g(x, y) \cdot \cos[2\pi f(x \cos \theta + y \sin \theta)] \quad (5.8)$$

$$h_o(x, y) = g(x, y) \cdot \sin[2\pi f(x \cos \theta + y \sin \theta)] \quad (5.9)$$

Where  $g(x, y)$  is 2D Gaussian function,  $f$  is the center frequency, and  $\theta$  is the orientation of the filter. The mean and the standard deviation of each filtered image (at certain  $f$  and  $\theta$ ) represent iris features.

In addition to the Gabor filters, they used 2D wavelet transform whose wavelet basis is the Daubechies wavelet of order 4. As the information at the finer resolution level is strongly affected by noise, only five low resolution levels are used. The mean and the standard deviation of each resolution level represent iris features.

### 3. Matching Techniques

Some of the most well known techniques for iris template matching are:

Hamming distance, Normalized Correlation, and Weighted Euclidean Distance.

#### 3.1 Hamming distance

Daugman (1994) employed the Hamming distance to find the dissimilarity between two iris templates  $A$  and  $B$ . This matching metric is defined as:

$$HD = \frac{1}{N - \sum_{i=1}^N Am_i (OR) Bm_i} \sum_{j=1}^N A_j (XOR) B_j (AND) Am_j^c (AND) Bm_j^c \quad (5.10)$$

Where  $Am$ ,  $Bm$  are the noise masks (noise region = 1s, other region = 0s) of the two iris templates  $A$  and  $B$ , respectively.  $Am^c$  and  $Bm^c$  are the complements of the noise masks. And  $N$  is the number of bits in the iris template.

#### 3.2 Normalized Correlation

Wildes *et al* (1996) found the similarity between two iris image arrays  $p_1[i, j]$  and  $p_2[i, j]$  by using the normalized correlation which can be defined as:

$$\frac{\sum_{i=1}^n \sum_{j=1}^m (p_1[i, j] - \mu_1)(p_2[i, j] - \mu_2)}{nm\sigma_1\sigma_2} \quad (5.11)$$

$$\mu_k = (1/nm) \sum_{i=1}^n \sum_{j=1}^m p_k[i, j] \quad , k = 1, 2 \quad (5.12)$$



$$\sigma_k = \sqrt{(1/nm) \sum_{i=1}^n \sum_{j=1}^m (p_k[i, j] - \mu_k)^2}, k=1, 2 \quad (5.13)$$

Where  $p_1$  and  $p_2$  are the two images of size  $n \times m$ .  $\mu_1$  and  $\sigma_1$  are the mean and the standard deviation of  $p_1$ , respectively.  $\mu_2$  and  $\sigma_2$  are the mean and the standard deviation of  $p_2$ , respectively.

Normalized correlation accounts for local variations in image intensity. In implementation, the correlations are performed discretely over small blocks of pixels (8x8) in each of the four spatial frequency bands that are instantiated in the Laplacian pyramid representations. These operations result in multiple correlation values for each band. Subsequent processing combines the block correlations within a band into a single value. In sum, this yields a set of four goodness-of-match values.

### 3.3 Weighted Euclidean Distance

Zhu *et al.* (2000) used the following Weighted Euclidean Distance classifier to identify the unknown iris.

$$WED(k) = \sum_{i=1}^N \frac{(f_i - f_i^{(k)})^2}{(\delta_i^{(k)})^2} \quad (5.14)$$

Where  $f_i$  is the  $i$ th feature of the unknown iris,  $f_i^{(k)}$  and  $\delta_i^{(k)}$  are the  $i$ th feature and its standard deviation of iris  $k$ ,  $N$  is the total number of features extracted from a single iris. Features of an unknown iris are compared with those of a set of known irises. It is identified as iris  $k$  if the Weighted Euclidean Distance is minimum.

Ma *et al* (2002) used another form of the Weighted Euclidean Distance as follows:

$$WED(k) = \sqrt{\sum_{i=1}^{BN} A_i \sum_{j=1}^N (f_{(i,j)}^k - f_{(i,j)})^2} \quad (5.15)$$

Where  $A_i$  is the  $i$ th weighting coefficient determined experimentally,  $BN$  and  $N$  are the number of sub-images (in the compared image) and the total number of features extracted from each sub-image, respectively.  $f_{(i,j)}$  and  $f_{(i,j)}^k$  are the  $j$ th feature component of the  $i$ th sub-image of the unknown iris and that of iris indexed by  $k$ . Features of an unknown iris are compared with those of irises in database. It is identified as iris indexed by  $k$  if the weighted Euclidean distance mentioned above is minimum at  $k$  and this minimum is also less than a reasonable threshold.

#### 4. The Proposed Approach for Feature Extraction, Encoding and Matching

After studying the previous techniques in the previous sections, Log-Gabor filters for feature extraction is chosen to use it in this work. The main motivation for this choice is the good results that were found by Libor Masek, who used Log-Gabor filters for feature extraction (Masek, 2003). The objectives of Masek were developing

human

extraction can be summarized as follows: The 2D normalized iris pattern, which is 1D signals, and these 1D signals are convolved with 1D log-Gabor filter. The rows of the 2D normalized pattern are taken as the 1D signals, each row corresponds to a circular ring on the iris region. The angular direction is taken rather than the radial one, which corresponds to columns of the normalized pattern, since maximum independence occurs in the angular direction (also this was found by Ma et al, 2003). Then phase data from 1D Log-Gabor filter was extracted and quantized to four levels to encode the unique pattern of the iris. Then the iris templates are compared to find the matching between them by using the Hamming distance. This process is illustrated in Figure 35.

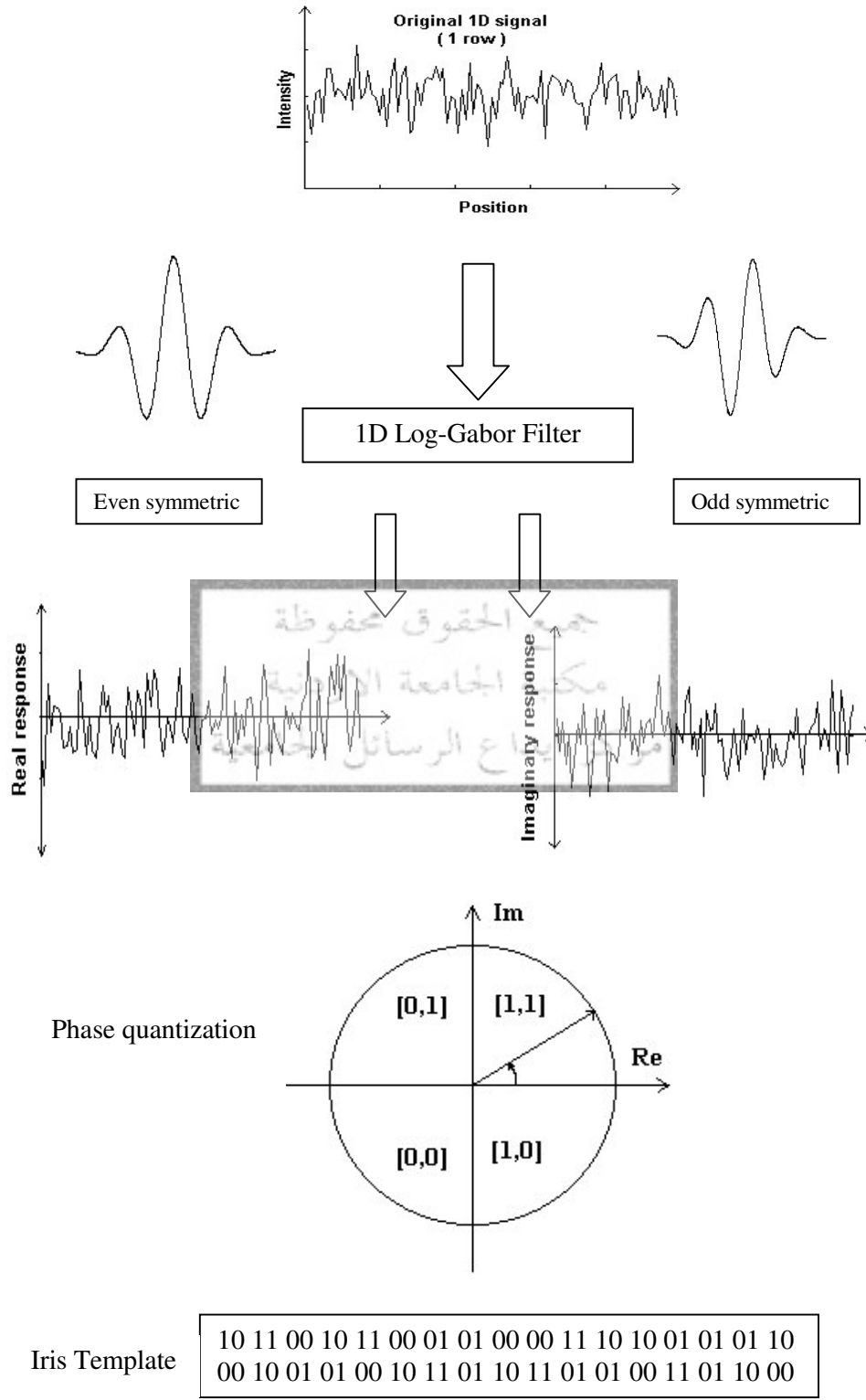


Figure 35: An illustration of Masek process for feature extraction and encoding.

After many experiments on CASIA iris database, Masek reached to the following important findings:

- A good resolution for the 2D normalized iris pattern is 20x240 pixels.
- The encoding process only required one 1D Log-Gabor filter to provide accurate recognition, since the open literature mentions the use of multi-scale representation in the encoding process.
- For CASIA iris database the optimum center wavelength of the Log-Gabor filter is 18 pixels ( $f_o = 1 / \text{center wavelength}$ ). And the optimum bandwidth  $\sigma/f_o$  is 0.5.
- For CASIA iris database the number of shifts required to compensate iris image rotation is 8 shifts, where one shift is defined one shift to the left, followed by one shift to the right, and in one shift two bits are moved.

The final result of Masek was that the encoding process produces a bitwise template with size 20x480 bits , and a corresponding noise mask with the same size (20x480 bits), which corresponds to corrupt areas within the iris pattern. From this result, the size of the bitwise iris template is very large (20x480 = 9600 bits), also the same size for the noise mask. This means that for each Hamming distance calculation, we need to shift both iris template(9600 bits) and noise mask (9600 bits) to the left by 8 shifts and to the right by 8 shifts ,in each shift we move two bits. And after each shift we calculate the Hamming distance by using Equation (5.10) and the minimum Hamming distance is chosen as a matching result. So many computations required to calculate only one Hamming distance, and a large size of memory is required to save the iris template (feature vector) and its noise mask. This reduces the speed of the system, which is one of the important aims that all the recognition systems try to increase it as possible. The speed factor can be one of the obstructions for applying the system in practical applications. Also for the required memory size; usually, for

practical applications the database contains hundreds or thousands of different classes. So the size of the iris template or the feature vector must be small as possible.

ecognition

system for practical applications, but to verify both the uniqueness of the human iris and also its performance as a biometric.

The proposed approach is

efficient technique to reduce the feature vector to a proper size such that the proposed algorithm can be applied in practical applications. The proposed approach can be summarized in the following steps:

- In experiments, it is found that the top-most 75 percent section (15x240) of an unwrapped iris image provides the most useful texture information. Also this was found by (Ma *et al.*2002). So take only the upper section 15x240 as a region of interest as shown in the Figure 36. Also do the same for the noise mask array that was performed in part 4.

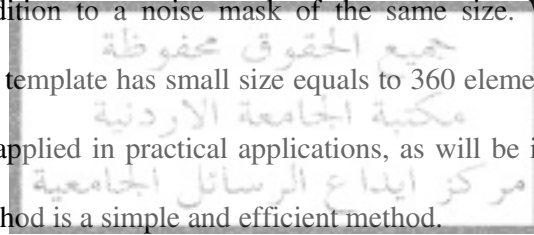


**Figure 36: The region of interest used in the feature extraction process**

- Filter the 2D normalized iris pattern, whose size equals to 15x240 pixels, by the Log-Gabor filter using the same manner and parameters used by Masek. So the result will be 15x240 array of complex numbers; the real parts are the outputs of the even symmetric filter and the imaginary parts are the outputs of the odd symmetric filter.
- Calculate the average of each 5x2 block in the filtered image. So the result will be 3x120 array of complex numbers.
- Calculate the angle of each complex number. So the result will be 3x120 array of real numbers (angles).

- Quantize the angles to four decimal levels (1,2,3,4); the angles in the first quarter quantized to 1, in the second quarter quantized to 2, in the third quarter quantized to 3, and in the fourth quarter quantized to 4. So the result will be 3x120 array of integer numbers (1, 2, 3, and 4).
- Resize the noise mask, where the noise region represented by NaNs and the other regions by zeros, to 3x120 pixels. Then add it to the previous result. By knowing that the result of any mathematical operation with NaN will be NaN, the result will be 3x120 iris template with these contents (1, 2, 3, 4, and NaN).

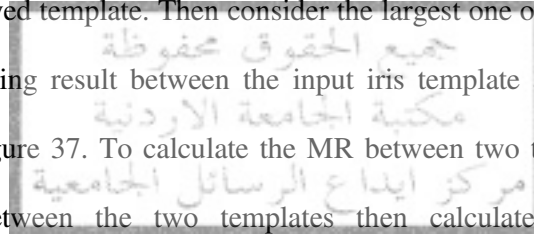
So, as illustrated previously that the iris template produced by Masek has size of 9600 bits in addition to a noise mask of the same size. Whereas, by the proposed approach the iris template has small size equals to 360 elements with a high recognition rate and can be applied in practical applications, as will be illustrated in the part 6. So the proposed method is a simple and efficient method.



Now, after the iris template (the feature vector) is obtained by feature extraction and encoding processes, a matching metric is required to measure the similarity between two iris templates. After matching process we can distinguish between two ranges of matching values; the first range contains the matching values of the same eyes and this range is called intra-class comparisons. The second range contains the matching values of different eyes and this range is called inter-class comparisons.

For template matching process a simple approach that based on calculating a simple matching ratio between the iris templates is proposed. This approach can be summarized by the following steps:

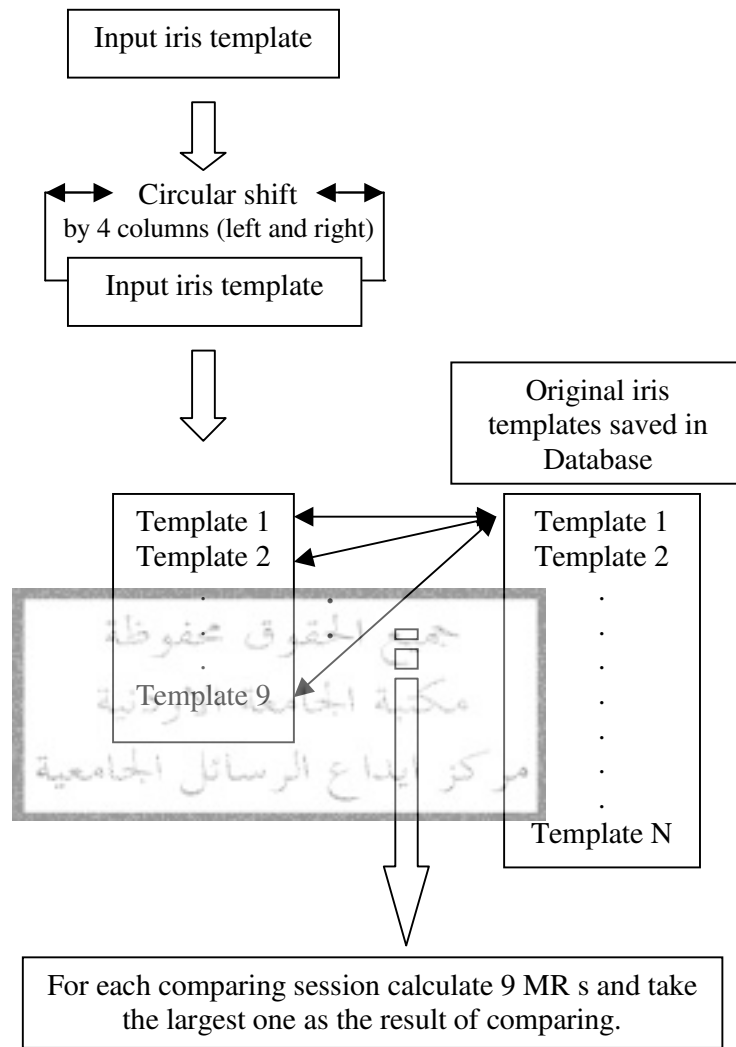
- Sometime there is a rotation in the iris ring due to the user head tilts, so the solution of this problem is done by constructing nine circular rotated templates (circular shift by 4 columns to the right and 4 columns to the left gives 8 templates and the original template). These templates are constructed only at matching time for each new input iris template that will be compared to the saved templates in the database. So in the database only the original template is saved ( not the rotated templates) as shown in Figure 37.
- Compare the input iris template with each saved iris template in the database by calculating nine matching ratios between the nine templates of the input template with each saved template. Then consider the largest one of the Matching Ratio (MR) as the matching result between the input iris template and the saved template as shown in Figure 37. To calculate the MR between two templates, do a subtraction operation between the two templates then calculate the Matching ratio as:



$$MR = \frac{N_z}{N_{nonNaN}} \quad (5.16)$$

Where  $N_z$  is the number of zeros in the subtraction result, and  $N_{nonNaN}$  is the total number of non NaN components in the subtraction result. We must note here that any mathematical operation with NaN the result will be NaN. And this is what we want; the noise region will remain after the subtraction process.

The final step is the classification between the Matching ratios of the intra-class (same irises) and inter-class (different irises), this is done by finding a suitable threshold between the Matching ratio distributions of intra and inter classes .when this threshold is chosen a compromising between False Accept rate (FAR) and False Reject Rate (FRR) must be considered as will be illustrated in the next part. Usually, we want both rates FAR and FRR to be very small as possible.



**Figure 37: Illustration of the matching process**



## Experimental Results and Conclusions

### 1. Introduction

To show the performance of the proposed algorithm, this part presents the experimental results for each stage of the algorithm, which are: iris segmentation and image quality assessment, iris normalization, feature extraction, encoding, and matching. Also the results are analyzed and compared with that of the well known algorithms as that for Daugman, Wildes *et al.*, Boles and Boashash, and Ma *et al.*

At the end of this part, conclusions about the whole proposed algorithm and some useful suggestions for future works are presented.

### 2. Software and Hardware Requirements

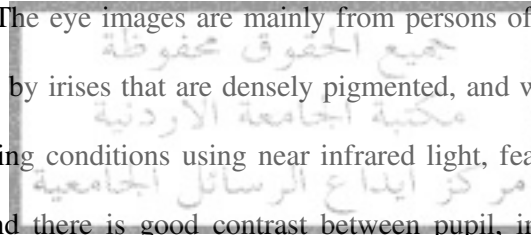
To implement the proposed algorithm, a rapid application development (RAD) approach is employed in order to get the results quickly. MATLAB provides an excellent RAD environment, with its image processing toolbox, digital signal processing toolbox, statistical toolbox, and wavelet toolbox. In addition to these utilities, there are many tools that can help in analyzing the results. Also it provides high level programming methodology.

MATLAB 6.1 running on a computer system with 300MHz Pentium II was used in implementation of the proposed algorithm. Really, both the software and hardware that are used have a low speed, since the MATLAB is an interpreted language, and 300MHz Pentium II is a low speed computer system, so using them is not suitable for practical applications.

### 3. CASIA Iris Database

To test the proposed algorithm CASIA iris images database is used, this database is from National Laboratory of Pattern Recognition (NLPR), Institute of Automation (IA), Chinese Academy of Sciences (CAS). This Academy provides iris database for iris recognition researchers. Most of the previous works used this database and advised to use it.

CASIA Iris database (ver 1.0) includes 756 iris images from 108 eyes (hence 108 classes). Images of each class were taken through two sessions with one month interval between sessions. Three samples are collected in the first session and four in the second session. The eye images are mainly from persons of Asian decent, whose eyes are characterized by irises that are densely pigmented, and with dark eyelashes. Due to specialized imaging conditions using near infrared light, features in the iris region are highly visible and there is good contrast between pupil, iris and sclera as shown in Figure 38.



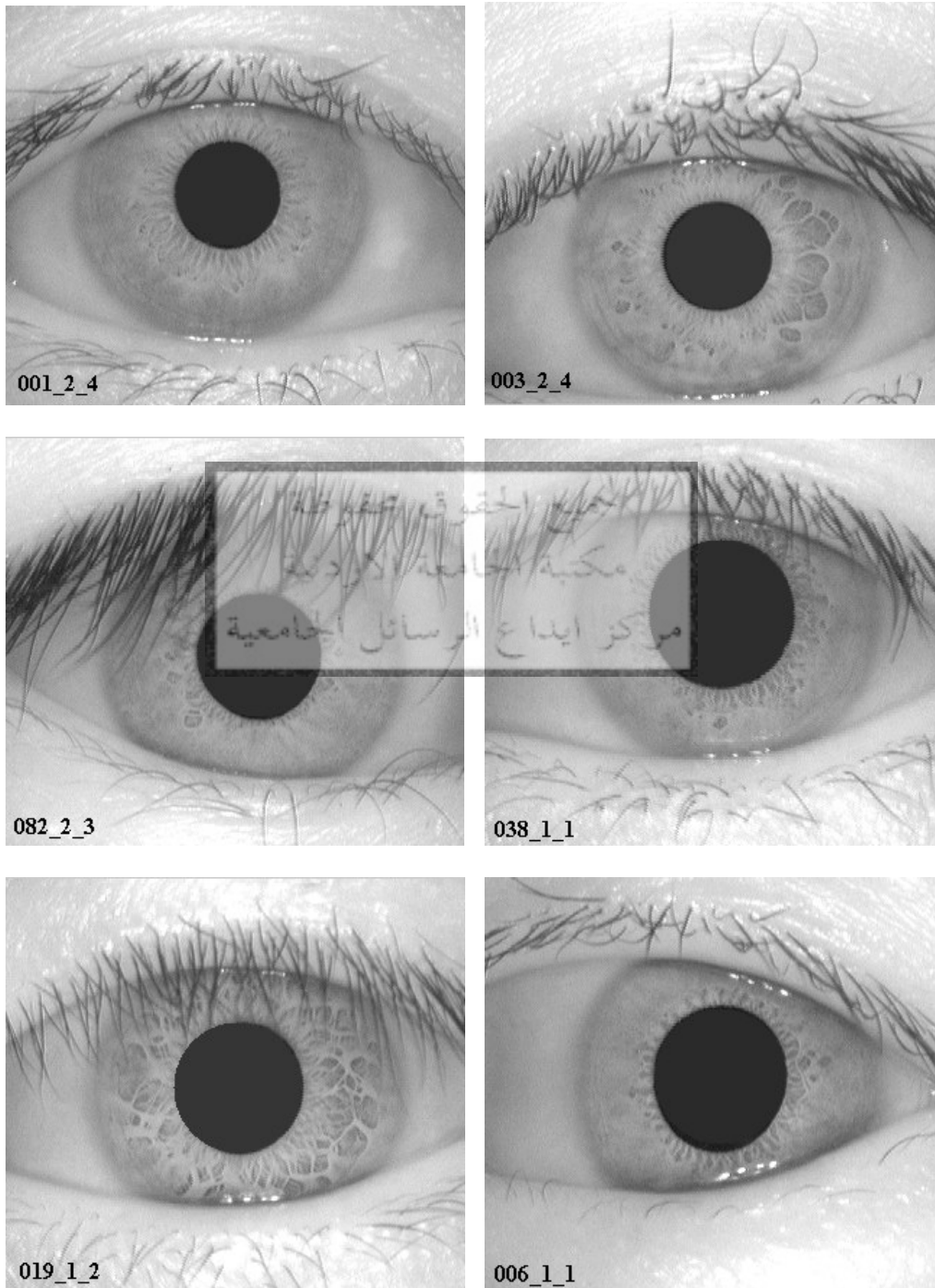


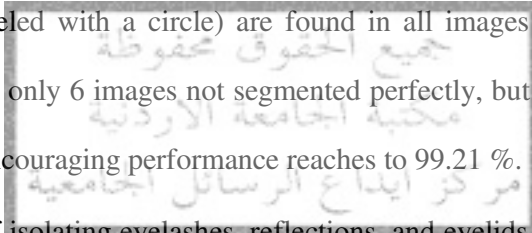
Figure 38: Some samples of CASIA iris database

#### 4. Iris Segmentation Results

The visual results for the implementation of each step of the proposed algorithm for iris segmentation were presented in part 3. But the statistical results for each step are:

- The results of finding accurate parameters (center and radiuses) for the inner boundary were encouraging; by this step the accurate center and radiuses for the inner boundary (as it is modeled with an ellipse) are found in all images (756 images) of CASIA database with no significant error.
- The results of finding accurate parameters (center and radius) for the outer boundary were encouraging; by this step the accurate center and radius for the outer boundary (as it is modeled with a circle) are found in all images (756 images) of CASIA database with only 6 images not segmented perfectly, but it has small errors. So this step has an encouraging performance reaches to 99.21 %.
- The results of isolating eyelashes, reflections, and eyelids, were encouraging; by this step the eyelashes, reflections, and eyelids are isolated in all images (756 images) of CASIA database with only 4 images not segmented perfectly, but they have small errors. So this step has an encouraging performance reaches to 99.47%.

From these results it is clear that the proposed algorithm for iris segmentation is a successful algorithm. Some of iris segmentation results are shown in Figure 39.



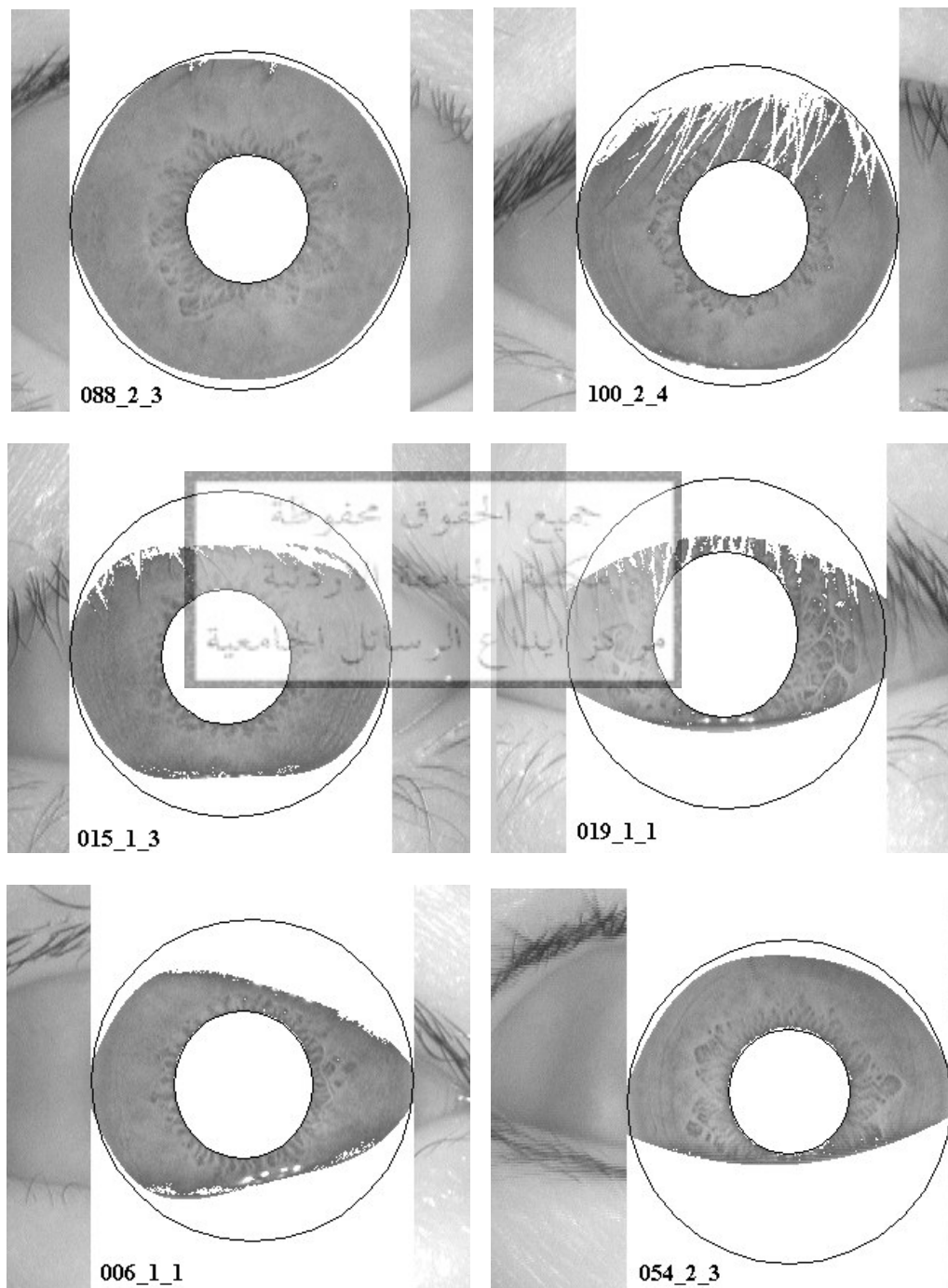
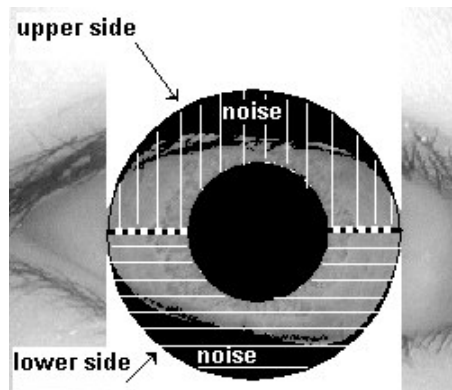
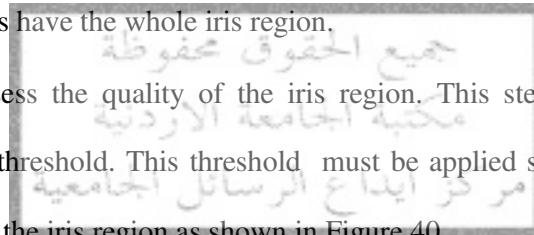


Figure 39: Segmentation results for various images from CASIA database.

## 5. Iris Image Quality Assessment Results

As illustrated in part 3, the proposed algorithm for iris image quality assessment is represented by two steps:

- In step one; the results of locating the inner and outer boundaries are checked by using the dimensions of the original images and the expected radiuses of the inner and outer boundaries. If there is any wrong result, the algorithm will stop and display a message as a feedback to the user asking him/her to center his/her eye to the imaging device better and to re-enter his/her eye image again. By this step the acquired eye images are ensured to have the whole iris region. For CASIA database all eye images have the whole iris region.
- Step two assess the quality of the iris region. This step requires a certain noise percent as a threshold. This threshold must be applied separately to the upper and lower side of the iris region as shown in Figure 40.

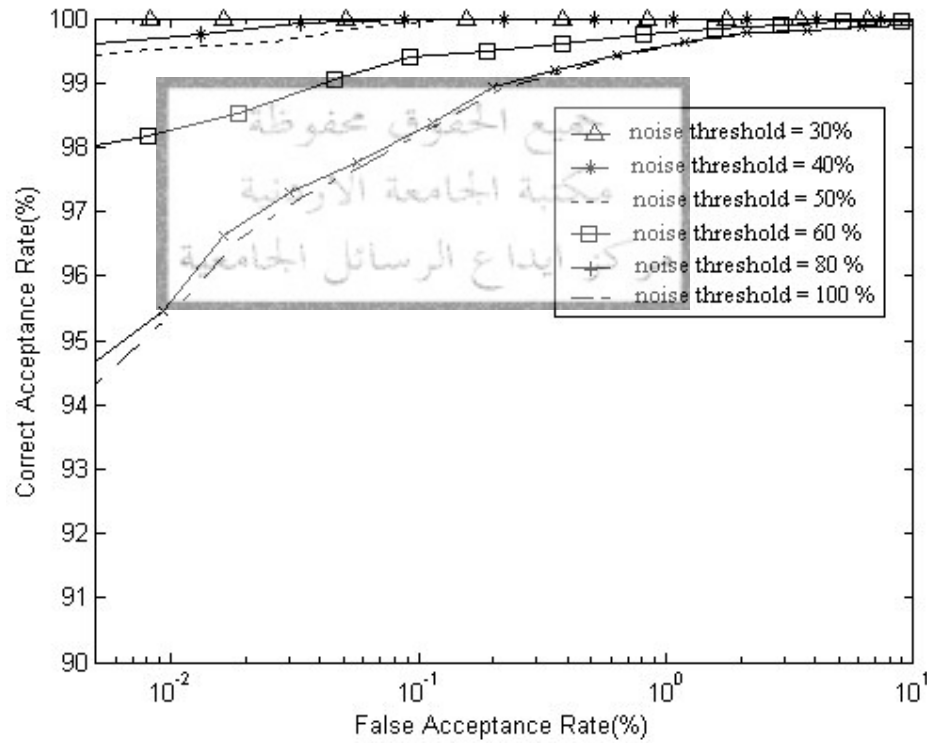


**Figure 40: The upper and lower sides of the iris region, with a noise region indicated by black color.**

Figure 41 shows the Receiver Operating Characteristic (ROC) curves for the proposed algorithm with a different noise threshold. The ROC curve is a plot of correct acceptance rate against false acceptance rate (Melsa and Cohn, 1978). It shows the

overall performance of a system. The ideal ROC curve is a step function at the zero false acceptance rate.

From Figure 41 and Table 3 it is clear that if the noise threshold decreases, both overall performance of the system and the number of rejected images will increase. And this is as expected; when the noise threshold decreases, the algorithm will accept only the images that have a noise percent less than the threshold, so it accepts only the images which have a small noise and this will increase the recognition rate or the overall performance and in the same time the number of rejected images will increase.



**Figure 41: Receiver Operating Characteristic (ROC) curves for different noise thresholds**

**Table 3: The relation between the noise threshold and the number of rejected images.**

Noise Threshold	30%	40%	50%	60%	80%	100%
Number of rejected images	446	259	136	65	8	0

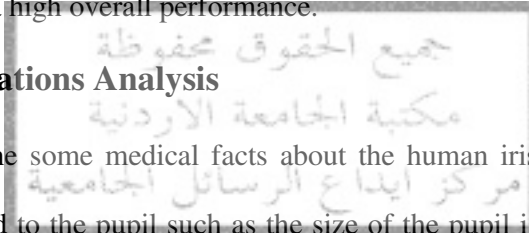
Therefore, to compromise between the overall performance of the system and the number of rejected images, administrators must clearly understand the value of the information or systems to be protected when the system is applied in the real applications.

From Figure 41 and Table 3, it is clear that 30% as noise threshold gives a perfect performance but the number of rejected images are very high. But 50% is more suitable noise threshold to show the high performance of the proposed algorithm with a large number of accepted images. By using 50% as the noise threshold we get 620 accepted images (about 82% of all images), and 136 images were rejected (about 18% of all images) with a high overall performance.

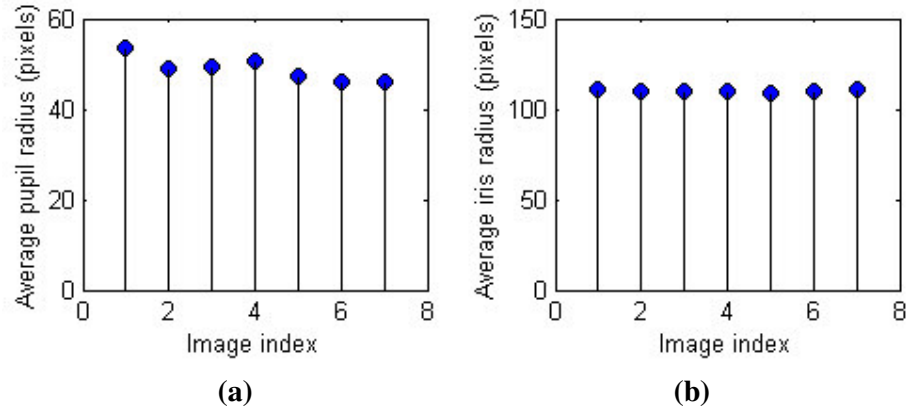
## 6. Pupil Oscillations Analysis

In part one some medical facts about the human iris were presented, some of these facts related to the pupil such as the size of the pupil is changed according to the light intensity, centers of the iris and the pupil are different, and the pupil radius can vary from 0.1 to 0.8 of the iris radius. In this section these facts are verified.

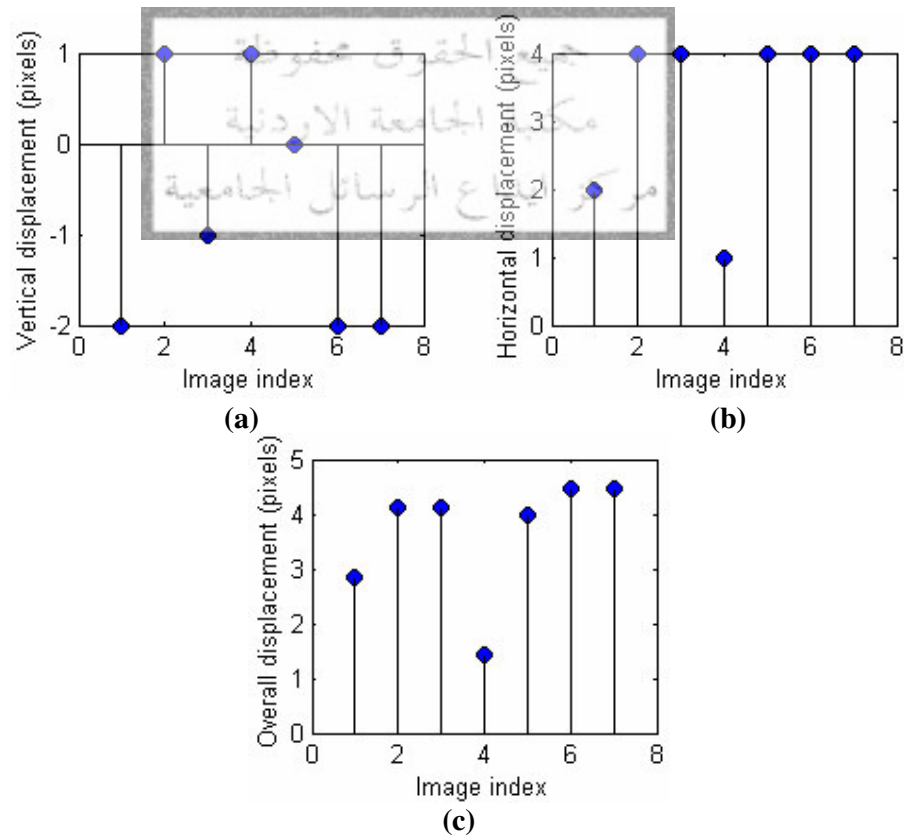
Figure 42 shows the changes in the pupil and iris radiuses for the same eye image. The changes in the pupil radius due to the changes in the light intensity, but the small changes in the iris radius due to variations in the imaging distance. Figure 43 shows the displacements between the pupil center and the iris center for the same eye image.







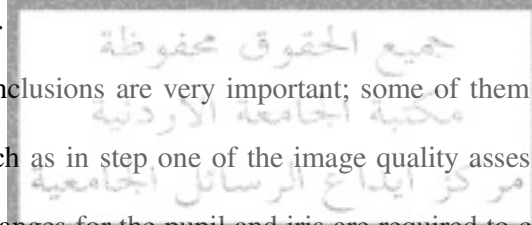
**Figure 42: (a) The changes in the pupil radius due to the changes in the light intensity (b) The small changes in the iris radius due the variations in the imaging distance. Both (a) and (b) are for the image 100 from the CASIA database**



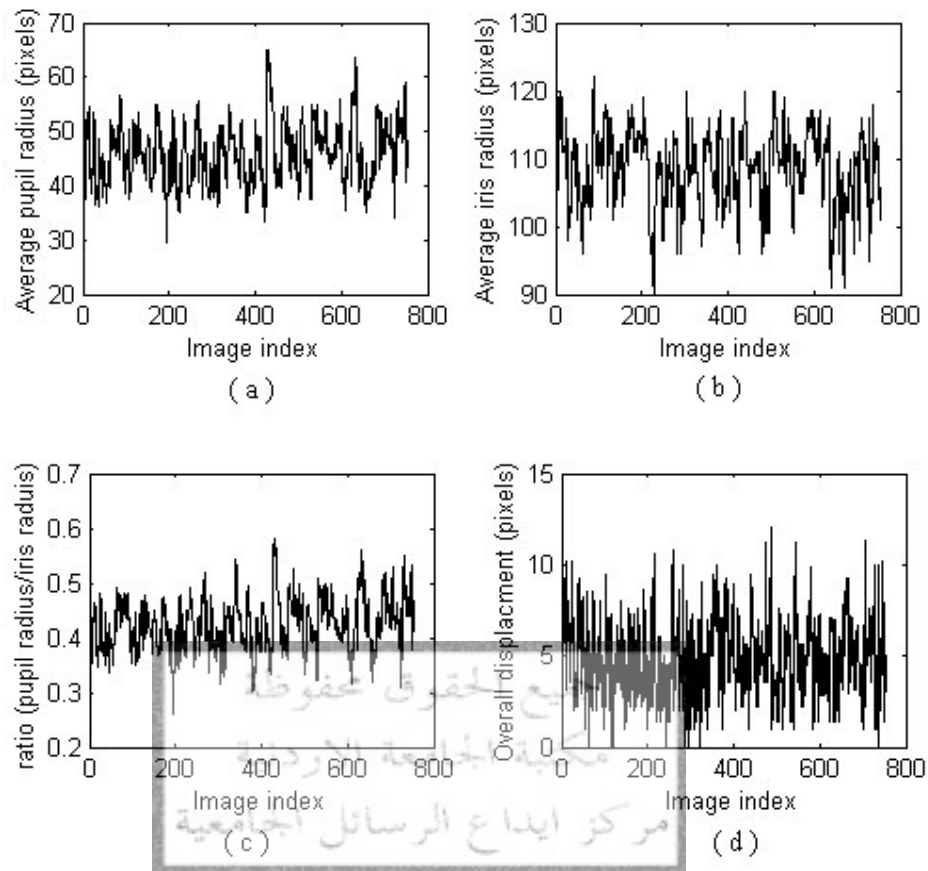
**Figure 43: (a) The vertical displacement of the pupil center relative to the iris center, (b) The horizontal displacement of the pupil center relative to the iris center, (c) The overall displacement of the pupil center relative to the iris center. All of these are for image 100 from CASIA database.**

The next analysis stage was done on all images of CASIA database. And the results are shown in Figure 44. Form this figure the followings can be concluded:

- The average pupil radius equals to 46 pixels.
- The average iris radius equals to 108 pixels. conclude
- The pupil radius is in the range [30, 65] pixels.
- The iris radius is in the range [90, 125] pixels.
- The average ratio of the pupil radius to the iris radius equals to 0.42
- The ratio of the pupil radius to the iris radius is in the range [0.25 , 0.6]
- The overall displacement of the pupil center relative to the iris center is in the range [0, 12] pixels.



These conclusions are very important; some of them are used in many steps of the algorithm such as in step one of the image quality assessment algorithm; where an expected radius ranges for the pupil and iris are required to check the results for finding the inner and outer boundaries. Also some of these findings are used to verify the assumption in the step of finding the outer boundary; where an approximate margin around the pupil center is required when the circular Hough transform is applied to find the outer boundary. The assumption for this margin was a small square that is determined by the limits  $(x_o \pm r_y/5, y_o \pm r_y/5)$ , in other words this assumption means that the distance between the pupil center and the iris center is approximately in the range  $[0, (\text{maximum average pupil radius}/5)]$ , and according to the results this range equals to  $[0, (60/5 = 12)]$ . So this assumption is true when comparing this range with the found range of the overall displacement, which is [0, 12].



**Figure 44: (a) average pupil radius for all images of CASIA database. (b) Average iris radius for all images of CASIA database. (c) Ratios of the pupil radius to the iris radius for all images of CASIA database. (d) Overall displacements between the pupil center and the iris center for all images of CASIA database**

## 7. Iris Normalization Results

The iris normalization process was a successful and an efficient process, one sample of the results is shown in Figure 45. By the normalization process two important issues are solved, which are the scale issue, and non-concentric issue between the pupil and the iris. The first issue is solved by using a fixed dimension (20x240) to all normalized irises, and the second issue is solved by the proposed technique for

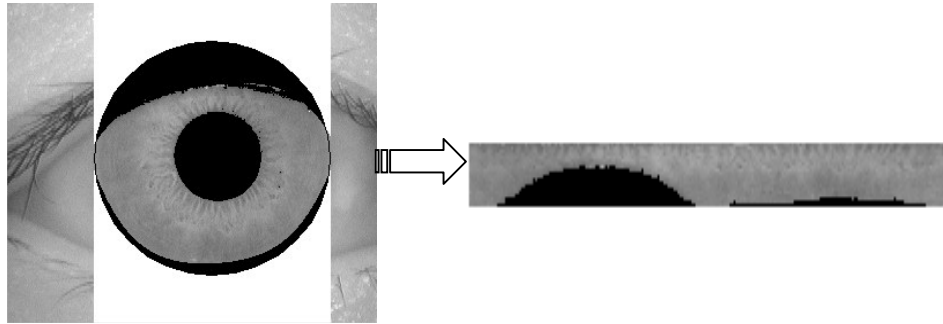
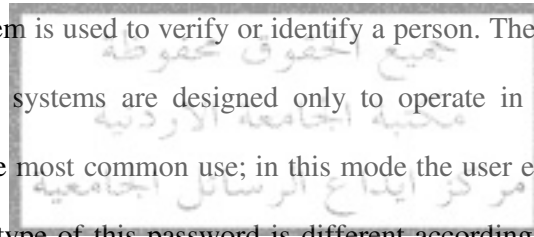


Figure 45: Normalization result

## 8. Feature Extraction, Encoding, and Matching Results

### 8.1 Algorithm Performance

To test the performance of any recognition system, it is important to distinguish whether the system is used to verify or identify a person. These are a different operation modes, some of systems are designed only to operate in one mode. Generally, the verification is the most common use; in this mode the user enters his/her username and a password, the type of this password is different according to the used system, in iris recognition system the password is the iris pattern. In traditional systems the password may be a certain word. In this mode only one-to-one comparison is performed.



But identification mode is more complicated; a one-to-many comparisons of images are performed, the system seeks to determine who the user is without information from the user. For example the face recognition system is commonly used for identification (Ali and Hassanien, 2003); the system captures an image of the user face and looks for matching in its database.

Here the performance of the proposed algorithm is tested in the two modes:

1. **Identification mode (i.e., one-to-many matching):** In this mode the performance of the algorithm is measured by the Correct Identification Rate (CIR); the ratio of the number of samples being correctly classified to the total number of test samples (or the unknown samples that we want to classify it). For experimental tests, 101

different iris samples (or 101 different classes) out of 618 iris samples are considered as a training samples (or known samples saved in the database as references for the different classes), and the rest (517 iris samples) as test samples (or unknown samples that we want to classify it to their classes). Then each test sample is compared to all training samples, and the test sample is identified and classified to the correct class according to the largest Matching Ratio (MR) and this MR must be larger than a certain threshold (0.427), this threshold is chosen according to the verification mode results that will be discussed next. The final result was: 516 test samples are identified correctly out of 517 test samples. Which is an encouraging Correct Identification Rate **CIR= 99.81 %**.

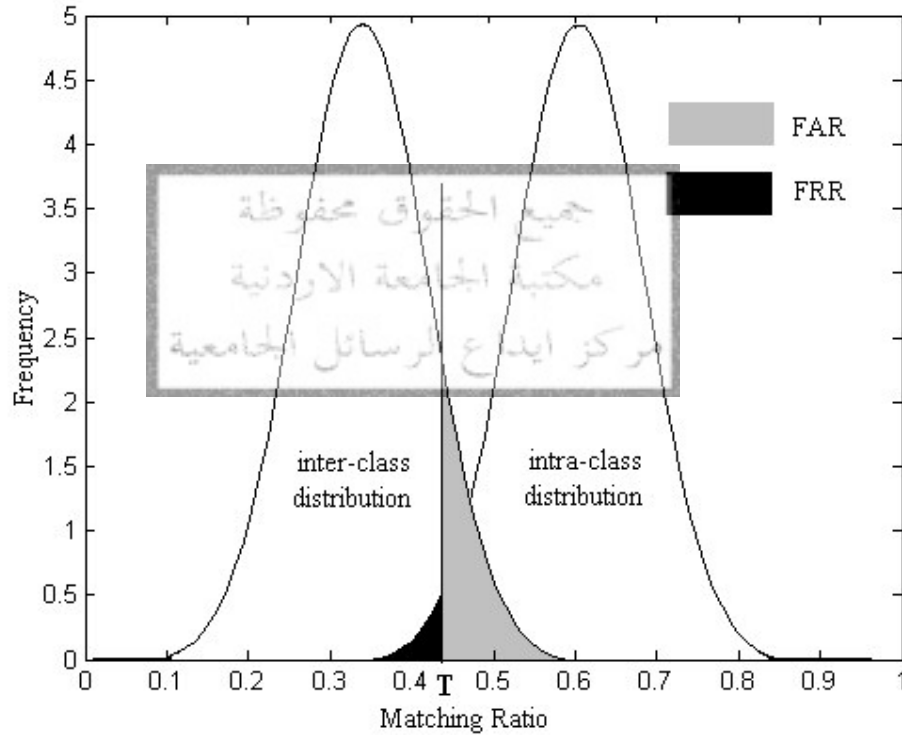
2. **Verification mode (i.e., one - to one matching):** In this mode, the results are presented statistically in two ways; the matching ratio distributions for the inter-class and intra-class, and the Receiver Operating Characteristic (ROC) curves.

- Figure 46 illustrates the Matching ratio distributions of the intra-class (same irises) and the inter-class (different irises). From this figure, there are two types of errors related to these distributions: False Reject Rate (FRR), also called type I error (Melsa and Cohn, 1978), measures the probability of reject the genuine users that were defined to the system previously. And False Accept Rate (FAR), also called type II error (Melsa and Cohn, 1978), it measures the probability of accept wrongly the imposter users as a genuine users. These errors are defined as

$$FRR = \frac{\int_0^T P_{Intra}(x)dx}{\int_0^1 P_{Intra}(x)dx} \quad (6.1)$$

$$FAR = \frac{\int_0^1 P_{Inter}(x) dx}{\int_0^1 P_{Intra}(x) dx} \quad (6.2)$$

Where  $P_{Intra}$  is the intra-class distribution,  $P_{Inter}$  is the inter-class distribution, and  $T$  is the threshold between the distributions. This threshold effect on both errors so when choosing it, both errors must be considered.



**Figure 46: Illustration of False Accept Rate and False Reject Rate**

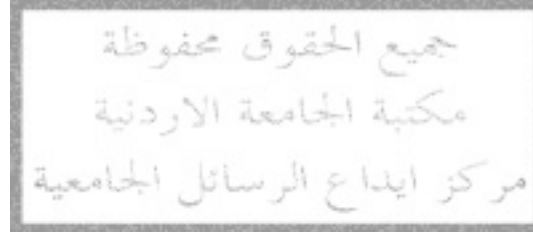
A better performance metric can be calculated for these distributions is the

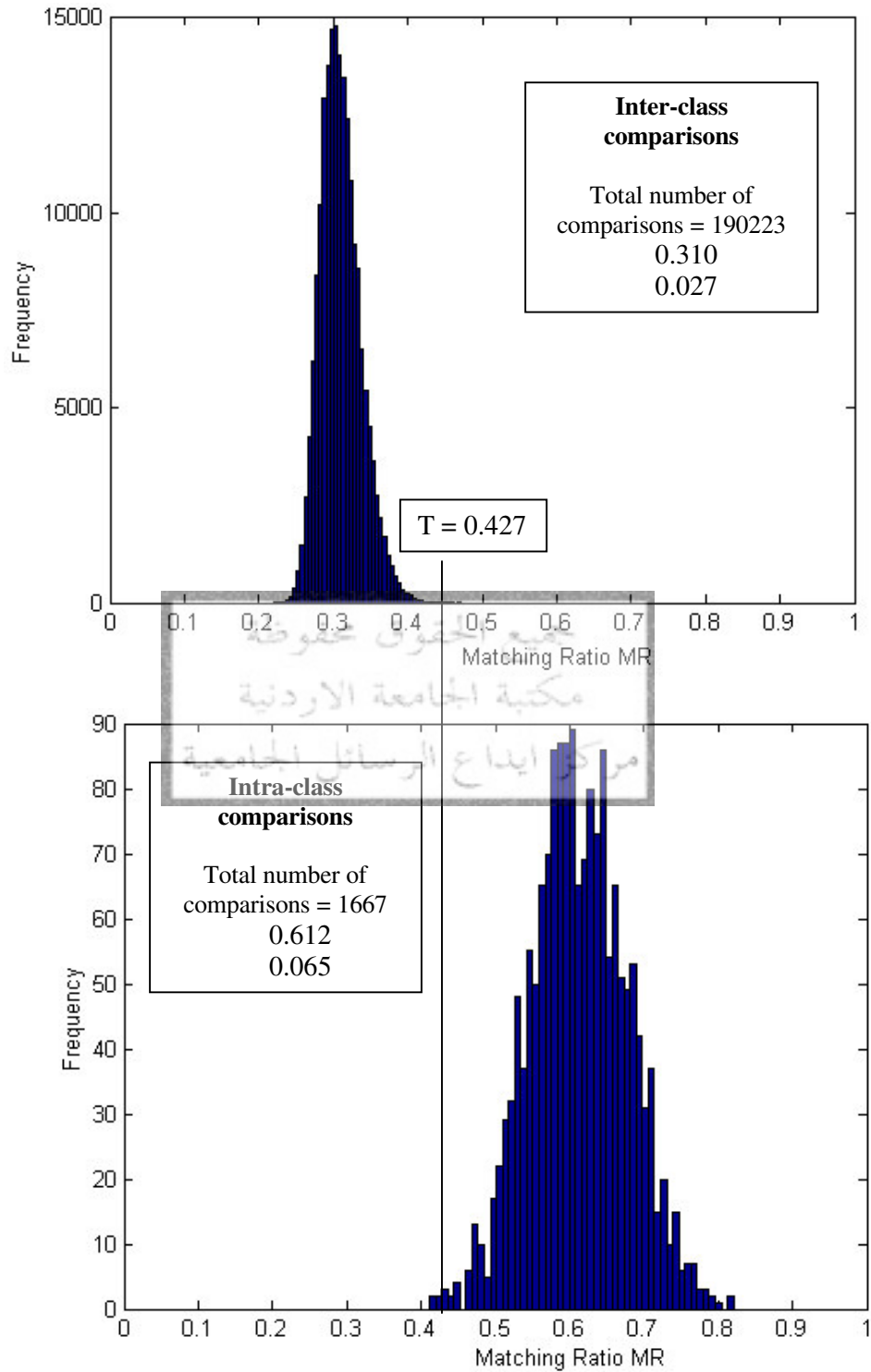
are; the higher decidability, the greater the separation of intra-class and inter-class distributions, which allows for more accurate recognition. This metric defined as

$$d = \frac{|\mu_s - \mu_D|}{\sqrt{\frac{(\sigma_s^2 + \sigma_D^2)}{2}}} \quad (6.3)$$

Where  $\mu_s$  is the mean of the intra-class (same irises) distribution,  $\mu_D$  is the mean of the inter-class (different irises) distribution,  $\sigma_s^2$  is the variance of the intra-class distribution,  $\sigma_D^2$  is the variance of the inter-class distribution.

For the proposed algorithm, the intra-class and inter-class distributions and the threshold between them are shown in Figure 47.





**Figure 47: Distributions of intra-class and interclass comparisons for the proposed algorithm.**

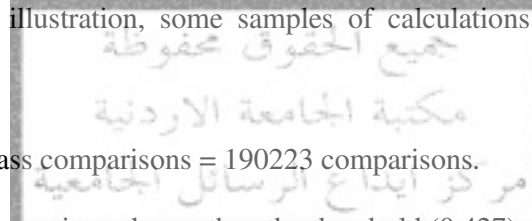


Some samples of FAR and FRR with different thresholds are shown in Table 4. From this table the best threshold is about 0.427, where a small and acceptable FAR and FRR are found.

**Table 4: False Reject Rate (FRR) and False Accept Rate (FAR) for different threshold points**

Threshold	False Accept Rate FAR %	False Reject Rate FRR %
0.425	0.043	0.180
0.426	0.039	0.180
<b>0.427</b>	<b>0.036</b>	<b>0.180</b>
0.428	0.033	0.240
0.429	0.030	0.300

For the sake of illustration, some samples of calculations for FAR%, FRR %, and



- Total inter-class comparisons = 190223 comparisons.

Inter-class comparisons larger than the threshold (0.427) = 69 comparisons.

$$\text{False Accept Rate (FAR) \%} = \frac{69}{190223} \times 100 \% = 0.036 \%$$

- Total intra-class comparisons = 1667 comparisons.

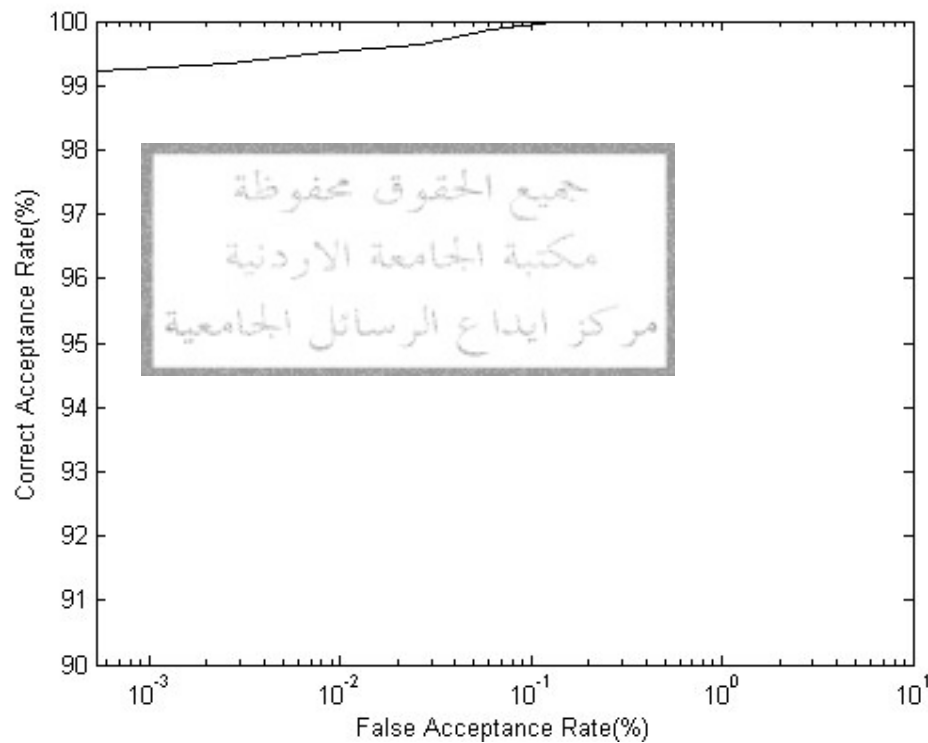
Intra-class comparisons smaller than the threshold (0.427) = 3 comparisons.

$$\text{False Reject Rate FRR} = \frac{3}{1667} \times 100 \% = 0.180 \%$$

- Decidability =  $\frac{|\mu_s - \mu_D|}{\sqrt{\frac{(\sigma_s^2 + \sigma_D^2)}{2}}} = \frac{|0.612 - 0.310|}{\sqrt{\frac{(0.065^2 + 0.027^2)}{2}}} = 6.10$

It is found that the decidability of these distributions is very good if it is compared to that of the well known method for Daugman, which is equal to 7.3 (Daugman, 2002).

- The Receiver Operating Characteristic (ROC) curve is used to report the performance of the algorithm. It is a plot of Correct Acceptance Rate (CAR % = 100-FRR %) against False Acceptance Rate (FAR %) at different thresholds. It shows the overall performance of the system. The ideal ROC curve is a step function at the zero false acceptance rate. Figure 48 shows the ROC curve for the proposed algorithm. This curve proves that the proposed algorithm has a high performance.



**Figure 48: The ROC curve for the proposed algorithm.**

## 8.2 Algorithm Parameters

As illustrated in part 5, the algorithm of feature extraction, encoding, and matching has many parameters which are:

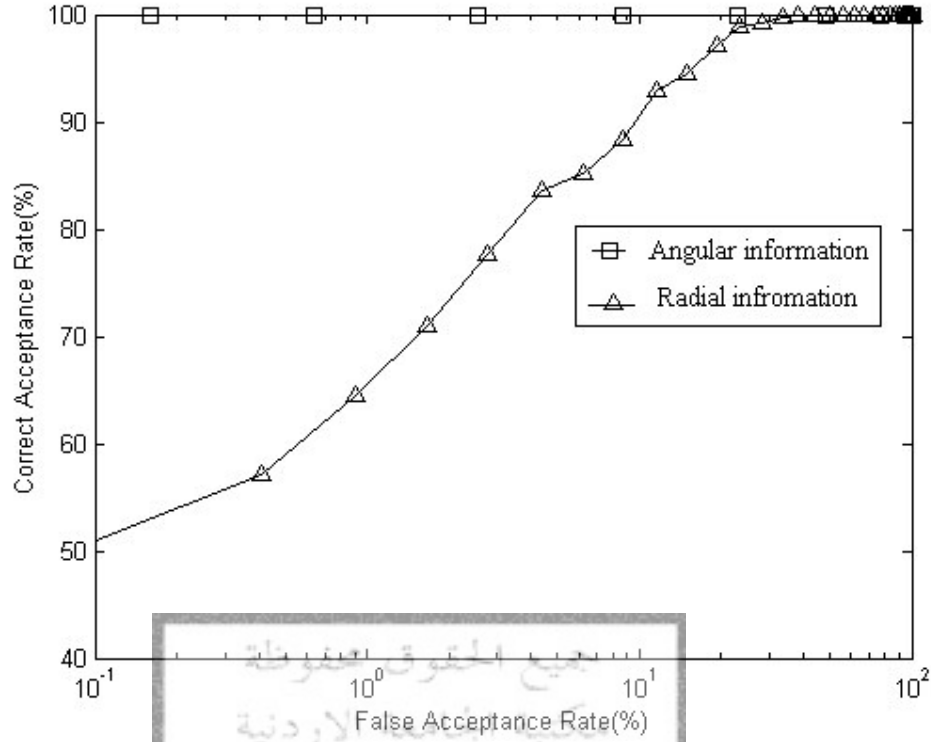
- The size of the normalized iris.
- The parameters of Log-Gabor filters: center wavelength, bandwidth, and number of filters.

- The number of shifts required to solve iris rotation issue.
- The direction of information; where maximum independence occurs (in the angular or radial direction).
- The size of the blocks that we calculate the average over.

As illustrated in part 5, the first four types of parameters are found by Masek (Masek,2003), and they are used in this work, these parameters are: the size of the normalized iris = 20x240 pixels, optimum center wavelength = 18 pixels, optimum bandwidth = 0.5 , number of filters = 1, number of shifts = 8 shifts ( left and right), and the direction of information is the angular direction. In this section the direction of information is verified and the best size of the blocks is found.

To find the best direction of information, where maximum independence occurs, the angular and radial directions are tested separately using 140 high quality images from CASIA iris database. To extract the most of information in the radial and angular

discussed in part 5 is used. For the angular direction test the rows of the normalized iris are considered as 1D signal, and these signals are filtered by 1D Log-Gabor filter with that was discussed in part 5. The final result will be a binary array of size 20x480 pixels. For the radial direction test, the procedure is the same, except that the columns of the normalized iris are considered as 1D signals instead of the rows. The final result will be a binary array of size 40 x 240 pixels. The ROC curves for the two tests are shown in the Figure 49. From this figure it is clear that information along the angular direction is highly discriminating for recognition rather than that along the radial direction. Another angular direction test was 6.78, but for the radial direction was 2.75.



**Figure 49: The Receiver Operating Characteristic (ROC) curves for the angular direction test and the radial direction test. These tests were done using 140 high quality images.**

For choosing the size of the blocks that the averages over are calculated, some important factors must be considered. Some of these factors are related to the horizontal dimension and others are related to the vertical dimension.

To choose the horizontal dimension of the block, the resolution of the angular information, must be kept high as much as possible and this dimension must not affect the rotation process that is done during the matching process to compensate the iris rotation that can occur due to the user head tilts. Due to these factors the test for the horizontal dimension of the block is limited to be 1 or 2 pixels only.

To choose the vertical dimension of the block, the test must be limited to the region of interest, which is the top-most 75 percent section (15x240) of an unwrapped iris image that provides the most useful texture information. The other factor is the size

of the iris template must be small as much as possible. In addition to these factors, the chosen size of the blocks must guarantee a high performance for the system.

According to these factors, experimental tests are done to choose the best size of the block; the results of these tests are shown in Table 5. From this table it is clear that the best size of the block is 5x2. By using this size a small iris template is obtained with a high performance.

**Table 5: Decidability values for various average block sizes**

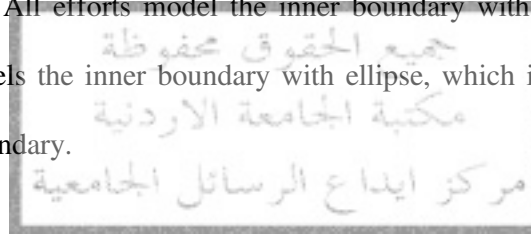
Size of the normalized iris region	Size of the blocks	Size of feature vector	Decidability
16x240	2x2	8x120 = 960	6.17
15x240	3x2	5x120 = 600	6.21
16x240	4x2	4x120 = 480	6.20
16x240	8x1	2x240 = 480	5.77
15x240	5x2	3x120 = 360	6.10
16x240	8x2	2x120 = 240	5.34
15x240	15x1	1x240 = 240	4.25
15x240	15x2	1x120 = 120	3.93

## 9. Comparing with Existing Algorithms

To evaluate the performance of the proposed algorithm, it is compared with some common existing algorithms. This comparing was performed as follows:

- **Iris segmentation process:** as illustrated in part 3, the most common methods for iris segmentation are -differential operator and the Hough transform. -differential operator can fail where there is noise in the eye image, such as from reflections, since it works on a local scale (Masek, 2003) (Tisse *et al.*, 2002). And as each point in an image  $I(x, y)$  is used as a test center, the process time would be very huge. But the proposed method is not very sensitive to specular reflections and it does not require a very high computation time. The Hough transform is used in many efforts, it requires threshold values to be chosen for edge detection, and this is not easy in practical applications and it is computationally intensive (Masek, 2003). And thus may not be suitable for real time

applications. The proposed method is compared with the Masek effort, who used the same database (CASIA database) and implemented the circular Hough Transform outer boundaries of the iris. And he used the linear Hough transform (Radon transform) to locate the upper and lower eyelids. The results of comparing the proposed method with that of Libor Masek are summarized in Table 6. These results are found by using the same database (CASIA database), and MATLAB 6.1 running on 300MHz Pentium II. From these results it can be concluded that the proposed method for iris segmentation has a high performance and it is suitable for practical applications. All efforts model the inner boundary with a circle, but the proposed method models the inner boundary with ellipse, which is more accurate model for the inner boundary.



**Table 6: The results of comparing our method for iris segmentation with that of Libor Masek**

Method	Techniques	Performance	Average speed
Masek	<ul style="list-style-type: none"> <li>- Modified version of Kovesi Canny edge detection to get the edge map.</li> <li>- Circular Hough transform to locate the inner and outer boundaries.</li> <li>- Linear Hough transform (Radon transform) to isolate the upper and lower eyelids.</li> <li>- Simple thresholding technique to isolate the reflections and eyelashes.</li> </ul>	<p><b>83 %</b> of images were segmented correctly.</p>	<p><b>15 minutes</b> To segment only one image.</p>
The proposed method	<ul style="list-style-type: none"> <li>- Finding an approximate location for the pupil region, then the Sobel edge detector is applied to get the edge map for the inner boundary.</li> <li>- Weighted compass kernels with some of morphological operations to get the edge map for the outer boundary and eyelids.</li> <li>- Elliptical and circular Hough transform to find the models of the inner and outer boundaries respectively.</li> <li>- Simple thresholding technique to isolate the reflections and eyelashes.</li> </ul>	<p><b>99 %</b> of images were segmented correctly.</p>	<p><b>55 seconds</b> To segment only one image.</p>

- **Iris image quality assessment:** most of the existing algorithms do not consider this step. But the importance of this step can be proved by Figure 41; from this figure it can be concluded that this process improve the overall performance of the system. It is not easy by hardware to ensure that the user is opening his/her eye suitably, so this step can be considered as a software process rather than a hardware process. But other quality factors such as image resolution can be solved by a good imaging hardware such as in (Daugman,1994) and (Wildes *et al.*1996) as illustrated in part 2, or can be checked by software as in (Ma *et al.*2003) as illustrated in part 3. Also for checking the existence of the iris region in the acquired image can be solved by a good imaging hardware as in (Daugman,1994) and (Wildes *et al.*1996) or can be checked by the software as in the proposed method . So for practical application this step must be considered.
- **Identification mode:** some systems can not be used in identification mode but only in verification mode, such as Wildes *et al.* system; as illustrated in the previous parts, Wildes *et al* used image registration and the normalized correlation techniques

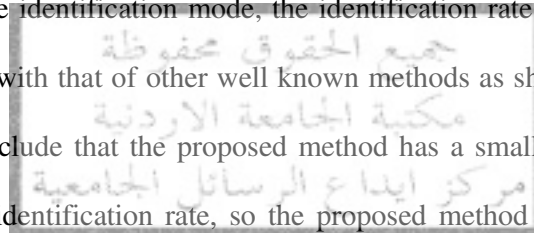
which are computationally very demanding. Also in other systems the limitation comes from the size of the iris template

large iris template (9600 bits for iris texture and 9600 bits for noise mask), so it takes a huge time for matching. And as known in the identification mode the system performs one-to-many comparisons. So, in case of a large database (and this is the usual case in the practical applications), the systems that have a large iris template or computationally very demanding online process can not be used in identification mode. But the proposed method produces a small (360 elements) iris template which is suitable for identification mode in practical applications. To evaluate the proposed

method in the identification mode, the identification rate of the proposed algorithm is compared with that of other well known methods as shown in Table 7. From this table we conclude that the proposed method has a small and suitable iris template with a high identification rate, so the proposed method has high and encouraging

performance in identification mode. Fortunately, the results of Table 7 are found in (Ma *et al*, 2003), and all these methods and the proposed method use the same type of database (CASIA database) but with different sizes; the size of database used by the proposed method equal to 756 images from 108 subjects but the size of that for the other methods in the Table 7 equals to 2255 sequence from 213 subjects.

Another note about Table 7 is the bootstrap method. It is a nonparametric statistical method used to make comparisons between different methods. Using of this method requires an implementation of the compared methods, to get the results from it then perform the comparison according to this method. But as there are no published results and the implementation of the other methods are not easy, this comparison method is not easy to apply. However these results can give acceptable evaluation



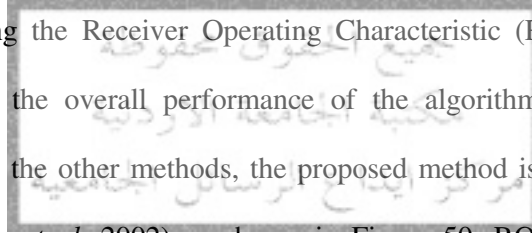


for the proposed method relative to the other methods as all of them use the same type of database (CASIA database).

**Table 7: Comparison of identification rates for different methods**

Method	Feature vector or template size	Recognition Rate %
Daugman(1994)	2048 bits	100.00
L. Ma et al./bootstrap(2003)	160 elements	94.91
L. Ma et al /bootstrap(2003)	384 elements	99.19
L. Ma et al /bootstrap(2003)	200 elements	99.43
L. Ma et al (2002)	160 elements	93.20
L. Ma et al (2002)	384 elements	99.85
Boles/bootstrap(1998)	Not published	92.64
The proposed method	360 elements	99.81

- **Verification mode:** in this mode the proposed method is compared with other methods using the Receiver Operating Characteristic (ROC) curve; it is a better indication to the overall performance of the algorithms. Due to no results are published for the other methods, the proposed method is compared with what it is found in (Ma *et al.* 2002) as shown in Figure 50. ROC curve for the proposed method is shown in Figure 51. From these figures it is clear that the proposed method has a high and encouraging performance.



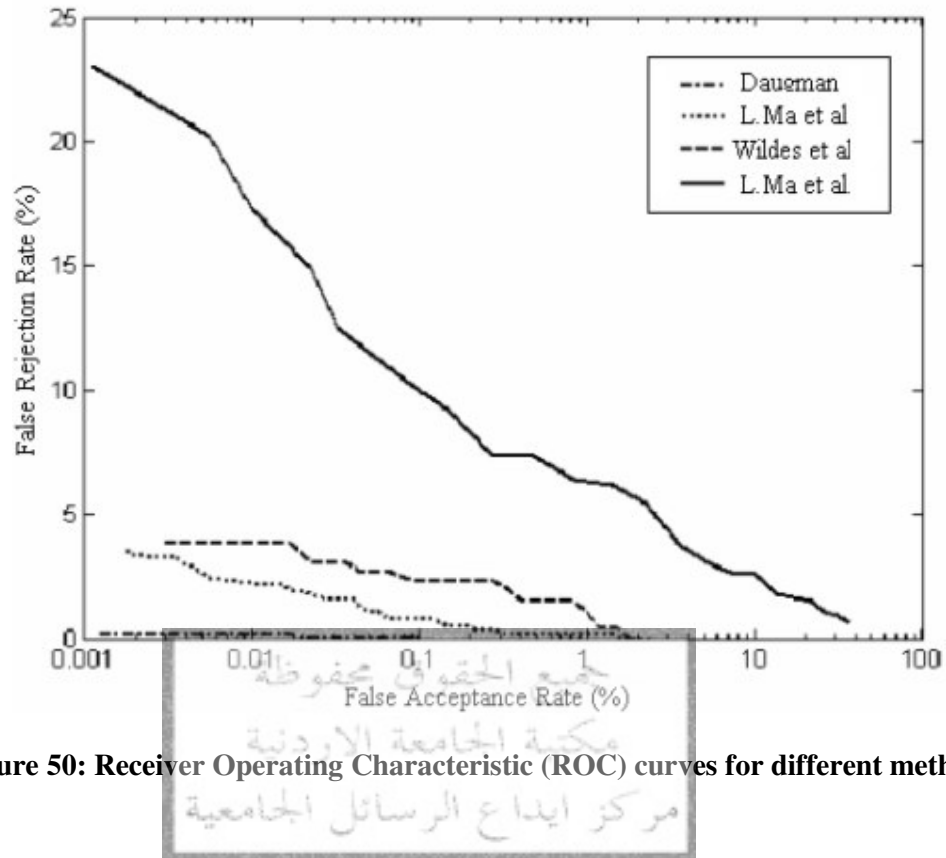


Figure 50: Receiver Operating Characteristic (ROC) curves for different methods.

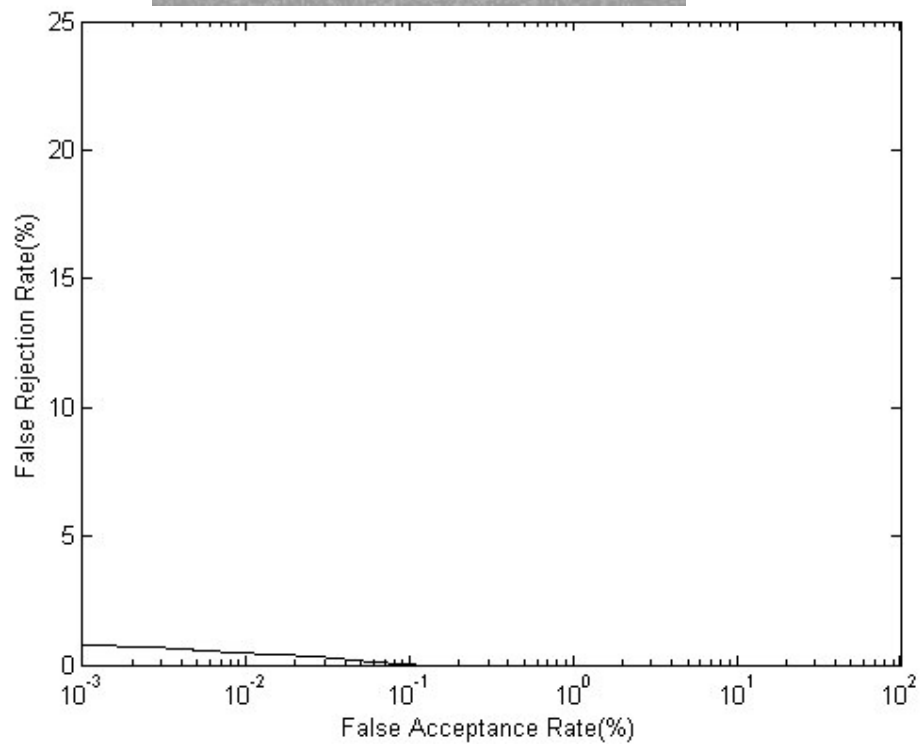


Figure 51: Receiver Operating Characteristic (ROC) curve for the proposed algorithm.

## 10. Summary and Conclusions

This thesis presented an efficient algorithm for iris recognition, this algorithm is composed of iris segmentation, iris image quality assessment, iris normalization, feature extraction, encoding, and matching.

For iris segmentation an approximate center and radius for the inner boundary of the iris are found, then the exact inner and outer boundaries of the iris are found by applying the elliptical and circular Hough transform, respectively. After that the upper and lower eyelids are located by using a weighted compass edge detector with a sequence of image processing that ends by modeling the upper and lower eyelids with a polynomials of power two or four according to the locations of these edges. A simple thresholding technique is used to isolate the eyelashes and reflections.

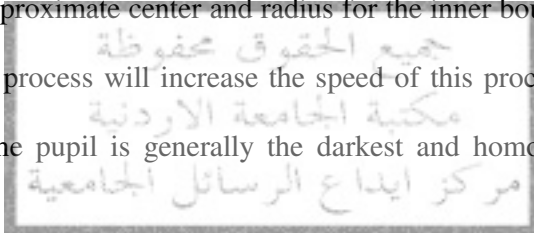
In iris image quality assessment process, the noise region percent in the iris region is evaluated and according to a predetermined threshold the images will be accepted or rejected. When the administrator determines this threshold he/she must clearly understand the value of the information or systems to be protected; as decreasing this threshold will increase the performance of the system but also will decrease the number of accepted images.

using this technique the segmented iris region is unwrapped into a rectangular block with a constant dimensions. The normalization process is useful to eliminate dimensional inconsistencies between iris regions.

After that, the rows of the normalized iris are filtered by using 1D-Log-Gabor filter, then the averages of a small blocks of the filtered images are calculated, then the phase information of the averaging process results are quantized to four decimal levels (1,2,3,4). The final result is a small iris template with size of 360 elements. The

matching between these templates can be found by a simple matching ratio, which is similar to the Hamming distance. During the matching process the rotation in the iris region is considered by rotating the iris template by a certain angles.

From a practical point of view, any human recognition system must be fast, reliable, and automatic identification or verification system. The experimental results prove that the proposed algorithm achieves in the software part the previous characteristics; it is fast, reliable, and automatic human identification and verification algorithm. This can be verified by discussing the main keys of the proposed algorithm which are:

- Finding an approximate center and radius for the inner boundary as a first step in the segmentation process will increase the speed of this process. The main idea in this step is that the pupil is generally the darkest and homogenous region in the eye image. 
- Modeling the inner boundary with elliptical model will give more accurate segmentation process. Also this may be useful when the changes in the pupil size is exploited for iris liveness detection.
- Iris image quality assessment automatically rejects the defected images and accepts only the high quality images. So this will improve the performance of the system and make it more reliable and automatic system.
- that the pupil can be non-concentric to the iris must be considered. In the proposed algorithm this issue is considered and solved by a clever technique. So this will give accurate normalization process.

- Using the information along the angular direction, 1D Log-Gabor filter, averaging technique, and phase quantization give a small, suitable, and accurate iris template. So this will give a high speed and accurate algorithm.

The proposed algorithm is tested in identification mode and verification mode, and the experimental tests show that the proposed algorithm can be used in both modes with a high performance; it achieves a high identification rate which is equal to 99.81 %, also it proves a high verification rate when comparing it with the well known algorithms.

## 11. Suggestions for Future Work

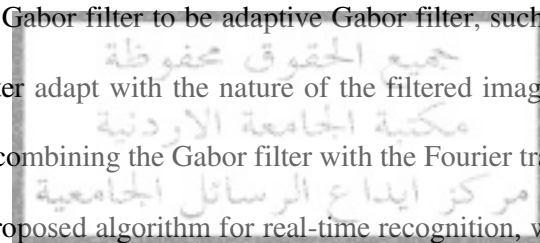
Based on this study of iris recognition, there are a number of suggestions for future work, which need to be addressed. These suggestions are:

- Iris liveness detection is very important; it aims to ensure that an input image is from a live subject instead of an iris photograph, a video playback, a glass eye, or other artifacts. However, efforts on iris liveness detection are still limited. So iris liveness detection remains to be an important research topic.
- Iris image quality assessment also is a very important topic. Both software and hardware must be shared to improve this topic. However, iris image quality assessment remains to be an important research topic.

- normalization is the best normalization technique, but there is an issue related to this technique, which is the size of the normalized iris. In other words, how we choose the radial and angular resolutions. Also this issue is related to the feature extraction techniques; for example, if we use Gabor filter for feature extraction, its parameters will depend on the size of the normalized iris. Most of efforts on iris recognition depend on the experimental tests to determine these parameters, but this is a

complex and hard process. So this issue must be considered more to get the optimum solution in a more regular way.

- As illustrated before in this thesis that using Gabor filters is the most used technique for feature extraction, but there is an issue related to this technique, which is determining the filter parameters, these parameters are optimum center frequency, optimum bandwidth, number of filters, and the direction of these filters. Most of efforts on iris recognition depend on the experimental tests to determine these parameters, but this is a complex and hard process. So this issue must be considered more to get the optimum solution in a more regular way. I suggest improving the definition of Gabor filter to be adaptive Gabor filter, such that the parameters of the improved filter adapt with the nature of the filtered image. This suggestion may be achieved by combining the Gabor filter with the Fourier transform.
- To use the proposed algorithm for real-time recognition, we must consider the speed factor so the algorithm must be implemented in a proper programming language as C or C++ instead of MATLAB and run on a high speed computer system.



## References:

Jafar M. Ali and Aboul Ella Hassanien, (2003). **An Iris Recognition System to Enhance E-security Environment Based on Wavelet Theory**. AMO - Advanced Modeling and Optimization, Vol.5, No.(2).

W.W. Boles and B. Boashash, (1998). **A Human Identification Technique Using Images of the Iris and Wavelet Transform**. IEEE transaction on signal processing , Vol.46, No.(4).

Kenneth R.Castleman, (1979). **Digital Image Processing**. Englewood Cliffs, N.J: Prentice Hall.

David A.Clausi, M.Ed Jernigan, (2000). **Designing Gabor Filters for Optimal Texture Separability** . The Journal of the pattern recognition society,1835-1849.

John G. Daugman, (1994). **Biometric Personal Identification System Based on Iris Analysis**. Patent no. 5 291 560.

John G. Daugman, (2002). **How Iris Recognition Works**. Proceedings of 2002 International Conference on Image Processing, Vol.1.

D. Field, (1987). **Relations Between the Statistics of Natural Images and the Response Properties of Cortical Cells**. Journal of the Optical Society of America.

Leonard Flom, (1987). **Iris Recognition System**. Patent no. 4,641,349.

Rafel C.Gonzalez, and Richard E.Woods, (2002). **Digital Image Processing**, (2<sup>nd</sup> ed). Upper Saddle River, N.J : Prentice Hall .

Anil K.Jain, (1989). **Fundamentals of Digital Image Processing**. Englewood Cliffs, N.J: Prentice Hall.

Mikio Kamada, (2004). **Iris Authentication Apparatus**. Patent no.6,785,406 B1.

P. Kovesi. MATLAB Functions for Computer Vision and Image Analysis. Available at: <http://www.cs.uwa.edu.au/~pk/Research/MatlabFns/index.html>

Tai Sing Lee, (1996). **Image Representation Using 2D Gabor Wavelets**. IEEE transaction on pattern analysis and machine intelligence, Vol.18, No.(10).

Rhys Lewis, (1990). **Practical Digital Image Processing**. England: Ellis Horwood Limited.

Shinyoung Lim, Kwanyong Lee, Okhwan Byeon, and Taiyun Kim, (2001). **Efficient Iris Recognition through Improvement of Feature Vector and Classifier**. ETRI Journal, Vol.23, No.(2).

Gareth Loy, (2002). **Fast Computation of the Gabor Wavelet Transform**. DICTA2002: Digital image computing techniques and applications, 21-22 January, Melbourne, Australia.

L. Ma, Y. Wang, T. Tan, (2002). **Iris Recognition Based on Multichannel Gabor Filtering**. Proceedings of ACCV' 2002: the 5<sup>th</sup> Asian Conference on Computer Vision, 23-25 January, Melbourne, Australia.

Li Ma, Yunhong Wang, and Tieniu Tan, (2002). **Iris Recognition Using Circular Symmetric Filters**. -417.

Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang, (2003). **Personal Identification Based on Iris Texture Analysis**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.(12).

Libor Masek, (2003). **Recognition of Human Iris Patterns for Biometric Identification**. School of Computer Science and Software Engineering. The University of Western Australia. <http://www.csse.uwa.edu.au/~mask101/>.

Mitsuji Matsushita, (1999). **Iris Identification System and Iris Identification Method**. Patent no. 5, 901, 238.

James L. Melsa, and David L. Cohn, (1978). **Decision and Estimation Theory**. Tokyo: McGraw-Hill.

Ales Muron, Jaroslav Pospisil, (2000). **The Human Iris Structure and its Usages**. Univ. Palacki. Olomuc, Fac. Rer. Nat, Physica (39), 87-95.

Alan V.Oppenheim, and Ronald W.Schafer with John R.Buck, (1999). **Discrete- Time Signal Processing**, (2<sup>nd</sup> ed). Upper Saddle River, N.J: Prentice Hall.

Athanasios Papoulis, and S.Unnikrishna Pillai, (2002). **Probability, Random Variables, and Stochastic Processes**, (4<sup>th</sup> ed). New York : McGraw-Hill.

R. Szewczyk, P. Jablonski, D. Makowski, and A. Napieralski, (2002). **Automatic Person Identification on the Basis of Iris Pattern - Extraction Methods Separability Tests**.

Computer Science, Al. Politechniki (11), 93-590.

Christel-

**Person Identification Technique Using Human Iris Recognition**. Proceeding of Vision Interface, 294-299.

Anne Waugh ,and Allison Grant, (2001). **Anatomy and Physiology in Health and Illness**,(9<sup>th</sup> ed). New York: Churchill Livingstone.

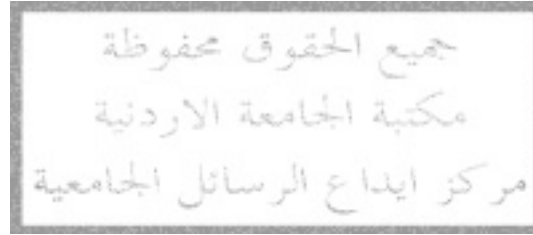
Richard P.Wildes, Jane C.Asmuth, Keith J.Hanna, Stephen C.Hsu, Raymond J.Kolezynski, James R.Matey, and Sterling E.McBride, (1996). **Automated, non-Invasive Iris Recognition System and Method**. Patent no. 5,5 72, 596 .



Richard P. Wildes, (1997). Iris Recognition: **An Emerging Biometric Technology**. Proceedings of the IEEE, Vol.85, 1348-1363.

Y. Zhu, T. Tan, and Y. Wang, (2000). **Biometric Personal Identification Based on Iris Patterns** -808.

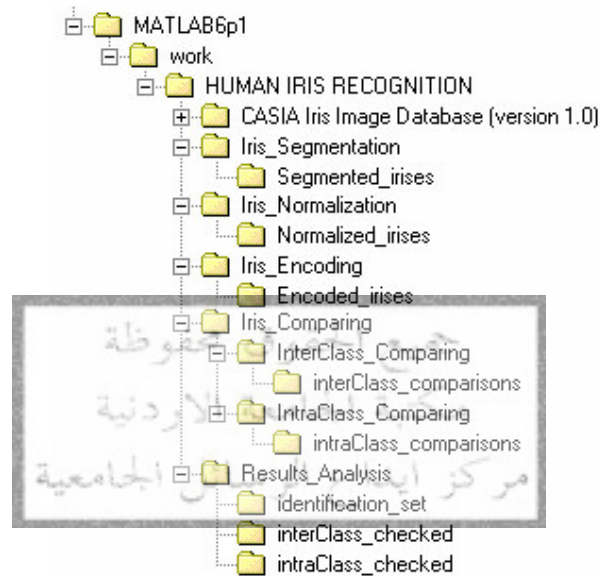
Chinese Academy of Sciences-Institute of Automation. Database of 756 Grayscale Eye Images. [http:// www.sinobiometrics.com](http://www.sinobiometrics.com) .Version (1.0), 2003.



**Appendix A**  
**Software Architecture**

جميع الحقوق محفوظة  
مكتبة الجامعة الاردنية  
مركز ايداع الرسائل الجامعية

The software of the proposed algorithm is arranged as shown in Figure 52 .

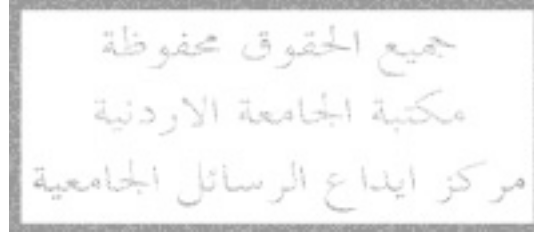


**Figure 52: Software architecture**

By using this architecture, we get the results of each stage separately, and this will facilitate analyzing the results and evaluation the system performance in each stage. The final sub-folders contain the final results of each stage. The program source code for each stage will be presented in the next Appendices.

## Appendix B

**This appendix contains the program source code for iris segmentation**

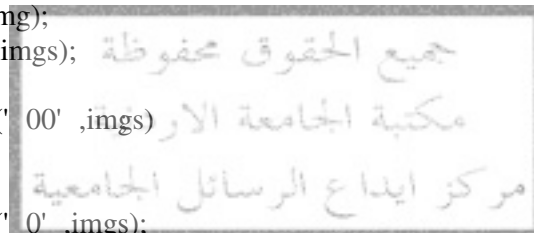


```

%////////////////////////////////////////////////////////////////////
% segment_iris
% This program: locates the iris boundaries and isolates noise areas such as
% eyelids, eyelashes and reflections
% This program was designed for CASIA iris database (ver 1.0)
%////////////////////////////////////////////////////////////////////
clc
close all
clear all
disp(' Start... ' )
%----- determine input images from CASIA iris database-----
from_img=1;
to_img=1;
up_threshold = 50; % upper noise threshold
low_threshold= 50; % lower noise threshold
%-----find the path of the inputs and read it -----
for img=from_img:to_img
    imgs=int2str(img);
    Limgs=length(imgs);
    if Limgs==1
        imgs=strcat('00' ,imgs)
    end
    if Limgs==2
        imgs=strcat('0' ,imgs);
    end
    for img_parts=1:2
        img_part=int2str(img_parts);
        if img_parts==1
            img_end=3;
        else
            img_end=4;
        end
        for img_indexs=1:img_end
            img_index=int2str(img_indexs);

            read_img= strcat(imgs,'\ ' ,img_part, ' ,imgs,' _' ,img_part,' _' ,img_index,' .bmp' );
            Segmentation_dir=cd;
            Recognition_dir=Segmentation_dir(1,1:length(cd)-length(' Iris_Segmentation' ));
            inputimg=imread([Recognition_dir,'CASIA Iris Image Database (version
            1.0)\' ,read_img]);
            %----- define masks matrices and outputimg for later uses-----
            maskmatrix1=zeros(size(inputimg));
            maskmatrix2=zeros(size(inputimg));
            outputimg=inputimg; % note: 255 grey level is reserved for noise regions
            %-----
% Note: for the next steps, imgIS - is a variable for image of index I in step S

```



```

%===== Step 1 : Find an approximate center and radiuses for the pupil =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear y;clear x;clear xnew;clear ynew;clear img11;clear img21;clear row;clear img31;
clear img41;clear r;clear c;clear tabler;clear maxr;clear rr1;clear cc1;clear rcenter;clear
rcenterapp; clear tablec;clear maxc;clear cradius;clear rr2;clear cc2;clear ccenter;clear
ccenterapp;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
[y,x]=size(inputimg);
xnew=x-10;
ynew=y-10;
img11=medfilt2(inputimg,[20 20]);
img21=img11([10:ynew],[10:xnew]);
for row=1:ynew-10
    img31(row,:)=img21(row,:)==min(img21(row,:));% image contains only the
                                                % minimum values in each row
end
img41=medfilt2(img31,[35 35]);
[r,c]=find(img41==1);
%-----
tabler=tabulate(r);
maxr=max(tabler(:,2));
rradius=maxr/2;% approximate pupil radius in the row direction(vertical direction)
[rr1,cc1]=find(tabler(:,2)==maxr);
rcenter=tabler(rr1,1);
rcenter=sum(rcenter)/length(rcenter);
rcenterapp=rcenter+10;% approximate row(or y) coordinate for the pupil center
%-----
tablec=tabulate(c);
maxc=max(tablec(:,2));
cradius=maxc/2;% approximate pupil radius in the column direction(horizontal
                                                direction )

[rr2,cc2]=find(tablec(:,2)==maxc);
ccenter=tablec(rr2,1);
ccenter=sum(ccenter)/length(ccenter);
ccenterapp=ccenter+10;% approximate column(or x) coordinate for the pupil center

%===== Step 2 : Find the pupil boundary exactly =====
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear img12;clear img22;clear i;clear j;clear a;clear b;clear r;clear c;clear EE;clear
rn;clear cn; clear an;clear bn;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
margin=10;% margin contains the pupil boundary
rminlimt= ceil(rcenterapp -rradius-margin);
rmaxlimt= ceil(rcenterapp +rradius+margin);
cminlimt= ceil(ccenterapp -cradius-margin);
cmaxlimt= ceil(ccenterapp +cradius+margin);
img12=img11([rminlimt:rmaxlimt],[cminlimt:cmaxlimt]);
img22=edge(img12,' sobel' );

```



```

img63=imfilter(img13,10*hw);
img73=im2bw(img63);
img43 = bwmorph(img73,'skel',Inf);
for i= 1:ynew
    for j= 1:centerapp-1
        if img43(ceil(i),ceil(j))==1
            rvalue= abs(i-rcenterapp);
            cvalue= abs(j-ccenterapp);
            theta=atan(rvalue/cvalue);
            if theta < 1.0472
                img53(i,j)=img43(i,j);
                % image contains only the edge points between -60 deg and +60 deg in the
                % left side of the edge map ,where the pupil center is the reference point.
            end
        end
    end
end
end

%-----search for the iris outer boundary -----
[iL,jL]=find(img53 ==1);
maxfreqL=1;
for r=ceil(rcenter+10-maxr/10):ceil(rcenter+10+maxr/10)
    for c=ceil(ccenter+10-maxc/10):ceil(ccenter+10+maxc/10)
        dist=ceil(((iL-r).^2+(jL-c).^2).^0.5);
        table=tabulate(dist+1);
        freq=max(table(:,2));
        [maxrL,maxcL]=find(table(:,2)==freq);
        radial=table(maxrL,1);
        if (freq >= maxfreqL)&(radial(1,1) > (maxc/2+10))&(radial(1,1)>(maxr/2+10))
            radialL=radial(1,1); % radius of the outer iris boundary
            maxfreqL=freq;
            rL=r; %row(or y) coordinate of the outer boundary center
            cL=c; %column(or x) coordinate of the outer boundary center
        end
    end
end

%===== Step 4 : Check the results of step1 and step2 =====
% Some useful limits are found to be used in the next steps
check=1; % for checking step
maxuplimit1=rL-radialL;
if maxuplimit1 <=0
    check=0;
end
maxuplimit2= re-an;
if maxuplimit2 <=0
    check=0;
end
rmid1=ceil((maxuplimit2+re)/2);
maxdownlimit2= re + an;
if maxdownlimit2 >= y

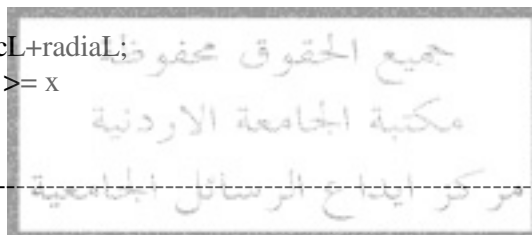
```



```

        check=0;
    end
    maxdowlimit1=rL+ radiaL;
    if maxdowlimit1 >= y
        check=0;
    end
    rmid2=ceil((re+maxdowlimit2)/2);
    maxleftlimit2=ce-bn;
    if maxleftlimit2 <=0
        check=0;
    end
    maxleftlimit1=cL-radialL;
    if maxleftlimit1 <=0
        check=0;
    end
    maxrightlimit2= ce+bn;
    if maxrightlimit2 >= x
        check=0;
    end
    maxrightlimit1= cL+radialL;
    if maxrightlimit1 >= x
        check=0;
    end
    %-----
    if check == 0
        figure, imshow(inputimg);
        title(' Please, center your eye to the imaging device, and recenter the eye image
              again ' );
    else
        % continue to the next step ...

```



### **%Step 5 : Mark the upper,lower eyelids,eyelashes,reflections,iris and pupil regions**

```

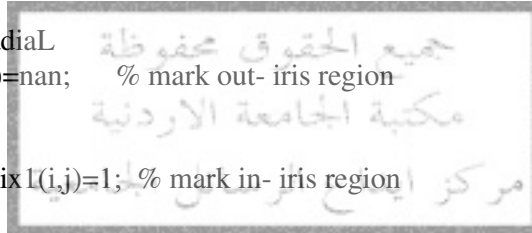
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear img14;clear img24;clear img34;clear img44;clear img54;clear img64;clear
img74;clear i;clear j;clear distp;clear img84;clear 94;clear i2;clear j2; clear i21;clear
j21;clear xpoi1;clear xpoi;clear ypoi1;clear ypoi;clear po;clear po1;clear ypoi2 ;clear
ypoi21;clear r;clear c;clear EE;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hn=[1 1 1;1 -2 1;-1 -1 -1]; % North compass gradient
hs=[-1 -1 -1;1 -2 1;1 1 1]; % South compass gradient
img14=img13;
img14(1:re,:)=imfilter(img13(1:re,:),30*hn);
img14(re:y,:)=imfilter(img13(re:y,:),30*hs);
img14(re,:)=0;
img24=im2bw(img14,0.8);
img34=medfilt2(img24,[1 5]);
img44=medfilt2(img34,[5 1]);
img54=imcomplement(img44);
img54(maxuplimit2-5:maxdowlimit2+5,maxleftlimit2-5:maxrightlimit2+5)=...
ones((maxdowlimit2+5-maxuplimit2+5)+1,(maxrightlimit2+5-maxleftlimit2+5)+1);

```

```

img54(rmid1:rmid2,maxleftlimit1:maxrightlimit1)=ones((rmid2-rmid1)+1,...
(maxrightlimit1-maxleftlimit1)+1);
img64=img54;
img64(1:maxuplimit1,:)=nan;
img64(maxdowlimit1:y,:)=nan;
img64(maxuplimit1:maxdowlimit1,1:maxleftlimit1)=nan;
img64(maxuplimit1:maxdowlimit1,maxrightlimit1:x)=nan;
for i=maxuplimit1:maxdowlimit1
    for j=maxleftlimit1:maxrightlimit1
        distp=ceil(((i-rL)^2+(j-cL)^2)^0.5);
        EE=ceil(((i-re).^2)/(an*an)+(j-ce).^2)/(bn*bn)); % Ellipse Equation
        if inputimg(i,j)<=100
            outputimg(i,j)=255; % mark the eyelashes
            maskmatrix2(i,j)=1;
        end
        if inputimg(i,j)>=240
            outputimg(i,j)=255; % mark the reflections
            maskmatrix2(i,j)=1;
        end
        if distp >=radialL
            img64(i,j)=nan; % mark out- iris region
        else
            maskmatrix1(i,j)=1; % mark in- iris region
        end
        if EE <= 1
            outputimg(i,j)=255; % mark pupil region
            maskmatrix1(i,j)=0;% mask the pupil region ( will be used in the next steps)
            maskmatrix2(i,j)=1;
        end
    end
end
img74 =(imfill((img64),8,' holes' ));
%----- mark the upper and lower eyelids -----
cou=0;
for j=maxleftlimit1:maxrightlimit1
    cou=cou+1;
    [i2u,j2u]=find(img74(maxuplimit1:re,j)== 0);
    icor2u=maxuplimit1+ max(i2u); % corrected i2u , i(rows) index of the upper edge
    xpoi_up(cou)=j; % x points for the upper edge
    ypoi_up(cou)=icor2u; % y points for the upper edge
    %-----
    [i2d,j2d]=find(img74(re:y,j)== 0);
    icor2d=re + min(i2d); % corrected i2d , i(rows) index of the lower(down) edge
    xpoi_do(cou)=j; % x points for the lower edge
    ypoi_do(cou)=icor2d; % y points for the lower edge
    %-----
end
xpoi_up = xpoi_up/max(xpoi_up); % scale x points ;
%-----

```



```

% Generate Z , H , to estimate the upper eyelid curve.
Z=ypoi_up' ;
H=[(sxpoy_up' ).^4 (sxpoy_up' ).^3 (sxpoy_up' ) ones(size(sxpoy_up,2),1)];
% result
theta=((H' *H)\(H' )'*Z;
po_up=theta' ;
%-----
ypoi_upr = round(polyval(po_up,sxpoy_up));
midpo_up=ypoi_upr(1,round(length(ypoi_upr)/2));
if midpo_up > (maxuplimit2 -(maxuplimit2-maxuplimit1)/4)
%-----
% Generate Z , H , to estimate the upper eyelid curve.
Z=ypoi_up' ;
H=[(sxpoy_up' ).^2 (sxpoy_up' ) ones(size(sxpoy_up,2),1)];
% result
theta=((H' *H)\(H' )'*Z;
po_up=theta' ;
%-----
ypoi_upr = round(polyval(po_up,sxpoy_up));
end
%-----
sxpoy_do = sxpoy_up ;
%-----
% Generate Z , H , to estimate the lower eyelid curve.
Z=ypoi_do' ;
H=[(sxpoy_do' ).^4 (sxpoy_do' ).^3 (sxpoy_do' ).^2 (sxpoy_do' ) ones(size(sxpoy_do,2),1)];
% result
theta=((H' *H)\(H' )'*Z;
po_do=theta' ;
%-----
ypoi_don = round(polyval(po_do,sxpoy_do));
midpo_do=ypoi_don(1,round(length(ypoi_don)/2));
if midpo_do < (maxdowlimit2 +(maxdowlimit1-maxdowlimit2)/4)
%-----
% Generate Z , H , to estimate the lower eyelid curve.
Z=ypoi_do' ;
H=[ (sxpoy_do' ).^2 (sxpoy_do' ) ones(size(sxpoy_do,2),1)];
% result
theta=((H' *H)\(H' )'*Z;
po_do=theta' ;
%-----
ypoi_don = round(polyval(po_do,sxpoy_do));
end
%-----
for j=maxleftlimit1:maxrightlimit1
    outputmg(1:ypoi_upr(j+1-maxleftlimit1),j)=255;% mark upper eyelid region
    outputmg(ypoi_don(j+1-maxleftlimit1):y,j)=255;% mark lower eyelid region
    maskmatrix2(1:ypoi_upr(j+1-maxleftlimit1),j)=1;
    maskmatrix2(ypoi_don(j+1-maxleftlimit1):y,j)=1;
end

```

**%== Step 6 : Check the segmentation results and give a message as feedback ==**

```

noise_img=and(maskmatrix1,maskmatrix2);
[iup,jup]=find(noise_img(1:rL,:)==1);
[ilow,jlow]=find(noise_img(rL:size(outputimg,1),:)==1);
noisarea_up=length(iup);
noisarea_low=length(ilow);
comparison_area=(pi*(radiaL^2)-pi*(an^2))/2;
noispercent_up=(noisarea_up/comparison_area)*100;
noispercent_low=(noisarea_low/comparison_area)*100;
%-----
if (noispercent_up > up_threshold) | (noispercent_low > low_threshold)
figure;imshow(outputimg);
%----- draw the outer boundary-----
c=[cL rL];r=radiaL;
a = [0:pi/16:2*pi];
H=line(r*cos(a)+c(1), r*sin(a)+c(2));
set(H,' color' , ' black' )
%----- draw the inner boundary-----
c=[ce re];rmax=an;rmin=bn;
a = [0:pi/16:2*pi];
H=line(rmax*cos(a)+c(1), rmin*sin(a)+c(2));
set(H,' color' , ' black' )
%-----
title(' Please, open your eye more, and re-enter the eye image again' );
text(1,5, strcat(' upper noise=' ,int2str(noispercent_up),' %' ))
text(1,20, strcat(' down noise=' ,int2str(noispercent_low),' %' ))
text(1,40,[' time = ' ,num2str(toc)]);
else
figure;imshow(outputimg);
%----- draw the outer boundary-----
c=[cL rL];r=radiaL;
a = [0:pi/16:2*pi];
H=line(r*cos(a)+c(1), r*sin(a)+c(2));
set(H,' color' ' black' )
%----- draw the inner boundary-----
c=[ce re];rmax=an;rmin=bn;
a = [0:pi/16:2*pi];
H=line(rmax*cos(a)+c(1), rmin*sin(a)+c(2));
set(H,' color' , ' black' )
%-----
title([read_img(7:9),'-' ,read_img(11),' ,read_img(13)]);
text(1,5, strcat(' upper noise=' ,int2str(noispercent_up),' %' ))
text(1,20, strcat(' down noise=' ,int2str(noispercent_low),' %' ))
text(1,40,[' time = ' ,num2str(toc)]);

```

**%=== Step 7 : Save the results, to be used in the next stages=====**

```

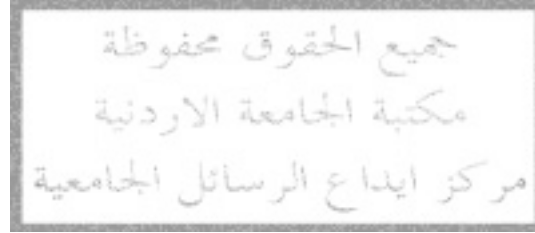
Segmentation_file = [cd,'\Segmented_irises\' ,imgs,' _' ,img_part,' _' ,img_index,'
segmentation.mat' ];
%save(Segmentation_file,' an' , ' bn' , ' rL' , ' cL' , ' radiaL' , ' maxuplimit1' , ' maxuplimit2' , ...

```

```

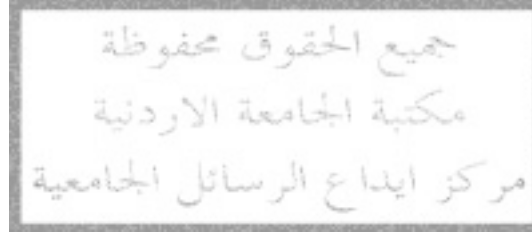
%' maxrightlimit1' , maxrightlimit2' , maxleftlimit1' , maxdownlimit1' , maxl
eftlimit2' , outputimg' , maskmatrix1' , maskmatrix2' );
end
end
end
end
end
disp (' End' )
%////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```



## Appendix C

**This appendix contains the program source code for iris normalization**

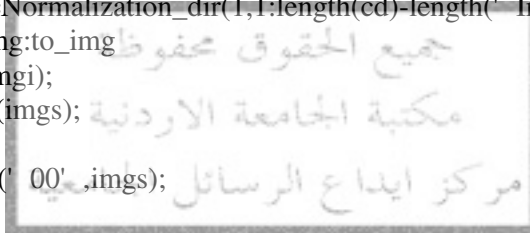


```

%////////////////////
% normalize_iris :
% This program: normalizes the iris region to a fixed dimensions array=20x240.
% This program is designed for CASIA iris database (ver 1.0)
%////////////////////
clc
close all
clear all
disp(' Start... ')
%--determine the input segmented irises that were resulted from iris segmentation-----
from_img=1;
to_img=1;
thta_dim=240; % angular length (240 pixles)
r_dim=20; % radial length (20 pixles)
thta_dim=thta_dim-1;
r_dim=r_dim-1;
%-----find the path of the inputs and read it -----
Normalization_dir=cd;
Recognition_dir=Normalization_dir(1,1:length(cd)-length(' Iris_Normalization' ));
for imgi=from_img:to_img
    imgs=int2str(imgi);
    Limgis=length(imgs);
    if Limgis==1
        imgis=strcat(' 00',imgs);
    end
    if Limgis==2
        imgis=strcat(' 0' ,imgs);
    end
    for img_parts=1:2
        img_part=int2str(img_parts);
        if img_parts==1
            img_end=3;
        else
            img_end=4;
        end
        for img_indexs=1:img_end
            img_index=int2str(img_indexs);
            segmentation_file = [Recognition_dir,'Iris_Segmentation' '\Segmented_irises' ',...
            imgis,'_' ,img_part,'_' ,img_index,'segmentation.mat' ];
            load(segmentation_file);

%=====Normalization process =====
%cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
clear ip;clear jp;clear angrad;clear ia;clear ja;clear r;clear rp;clear x;clear y;
%cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
edg_img=edge(maskmatrix1,' canny' );
[ip,jp]=find(edg_img ==1);%the iris edges

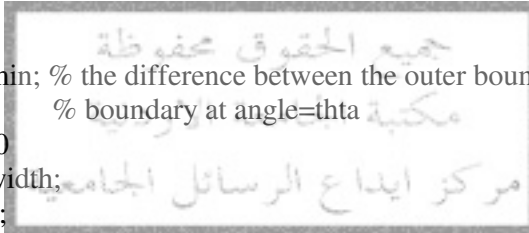
```



```

%-----Find the angles of all iris points with pupil center as the origin point-----
xp=jp-ce;
yp=re-ip;
angrad=atan2(yp,xp);
for i=1:length(angrad)
    if angrad(i) < 0
        angrad(i)=angrad(i)+2*pi;
    end
end
angdeg=round(angrad*180/pi);
%-----
rpupil=abs(bn);% radius of the pupil at angle = 0
iris_width=abs(radial-rpupil);% the width of iris region at angle = 0
cont=0;
for thta=0:360/thta_dim:360;
    cont=cont+1;
    [ia,ja]=find(angdeg==round(thta));
    p=((ip(ia)-re).^2+(jp(ia)-ce).^2).^0.5;
    pmax=max(p);
    pmin=min(p);
    diffp=pmax-pmin;% the difference between the outer boundary and the inner
                    % boundary at angle=thta
    if length(ia)==0
        diffp =iris_width;
        pmin=rpupil;
    else
        if diffp < 10
            diffp =iris_width;
            pmin=rpupil;
        end
    end
    iris_width=diffp;
    rpupil=pmin;
    iw(cont)=iris_width;%the width of iris region at angle=thta
    rp(cont)=rpupil;%the radius of the pupil at angle =thta
end
rp_array= ones(1,r_dim+3)' *rp;
iw_array= ones(1,r_dim+3)' *iw;
iw_array= iw_array.* (ones(thta_dim+1,1)*[0:1/(r_dim+2):1])' ;
iw_array= iw_array+ rp_array;
iw_array = iw_array(2:(r_dim+2), :);
theta = 0:2*pi/thta_dim:2*pi;
theta = double(theta);
xcos_array = ones(r_dim+1,1)*cos(theta);
xsin_array = ones(r_dim+1,1)*sin(theta);
xc = iw_array.*xcos_array;
yc = iw_array.*xsin_array;
xc = ce+xc;
yc = re-yc;
imagewithnoise=[];

```

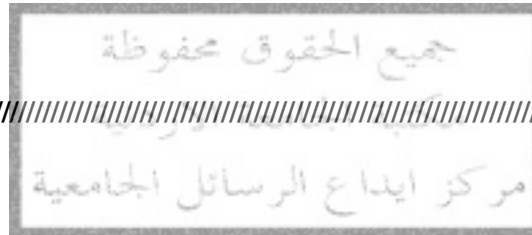




```

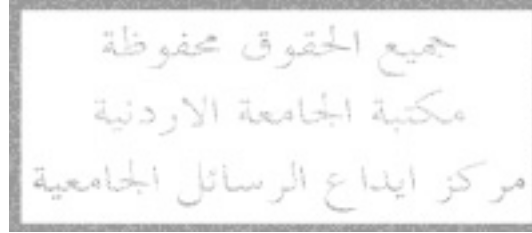
imagewithnoise(:,:)=outputimg(:,:);
imagewithnoise(imagewithnoise==255)=nan;
image=imagewithnoise;
[x,y] = meshgrid(1:size(image,2),1:size(image,1));
polar_array1 = interp2(x,y,image,xc,yc);
polar_array = double(polar_array1)./255;
xy_coord = find(isnan(polar_array));
polar_array(xy_coord) = 0;
polar_array(xy_coord) =
sum(sum(polar_array))/(size(polar_array,1)*size(polar_array,2)-length(xy_coord));
noise_array=zeros(size(polar_array));
noise_array(xy_coord)=1;
%----- save the results-----
Normalization_file = [cd,'\Normalized_irises\' ,imgis,' _' ,img_part,' _' ,img_index,'
normalization.mat' ];
save(Normalization_file,' polar_array' ,' noise_array' );
figure,imshow(polar_array)
end
end
end
disp(' End' )
%////////////////////////////////////

```



## Appendix D

**This appendix contains the program source codes for iris feature extraction and encoding**

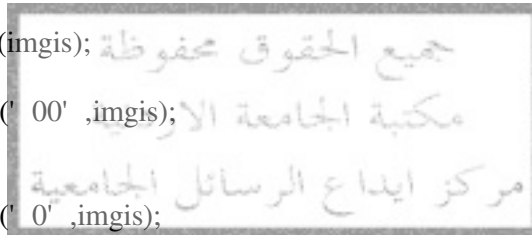


```

%////////////////////////////////////////////////////////////////////////////////////////////////////////////////
% Encode_iris
% This program: extracts iris feature from the normalized iris and encodes it to
% generate an iris template of size = 3x120.
% This program is designed for CASIA iris database (ver 1.0)
%////////////////////////////////////////////////////////////////////////////////////////////////////////////////
clc
close all
clear all
disp(' Start... ' )
%determine the input normalized irises that were resulted from normalization process
from_img=1;
to_img=108;
%-----find the path of the inputs and read it -----
Encoding_dir=cd;
Recognition_dir=Encoding_dir(1,1:length(cd)-length(' Iris_Encoding' ));
for imgi=from_img:to_img
    imgis=int2str(imgi);
    disp(imgis)
    Limgis=length(imgis);
    if Limgis==1
        imgis=strcat(' 00' ,imgis);
    end
    if Limgis==2
        imgis=strcat(' 0' ,imgis);
    end
end
for img_parts=1:2
    img_part=int2str(img_parts);
    if img_parts==1
        img_end=3;
    else
        img_end=4;
    end
end
for img_indexs=1:img_end
    img_index=int2str(img_indexs);
    Normalization_file = [Recognition_dir ' Iris_NormalizationNormalized_irises\' ,
    imgis,' _' ,img_part,' _' ,img_index,' normalization.mat' ];
    load(Normalization_file);
    noise_array(noise_array==1)=nan;
    polar_array=polar_array(1:15,:);
    noise_array=noise_array(1:15,:);

%=====Feature extraction process =====
[E0 filtersum] = gaborconvolve(polar_array, 1, 18,1,0.5);
%===== Encoding process=====
k=1;
E1 = E0{k}; % The output of the log-Gabor filter.
fun = inline(' sum(sum(x))10' );
E2=blkproc(E1,[5 2],fun); % The average of each block(5x2)
ang=angle(E2); % The angle of each complex number .

```



```

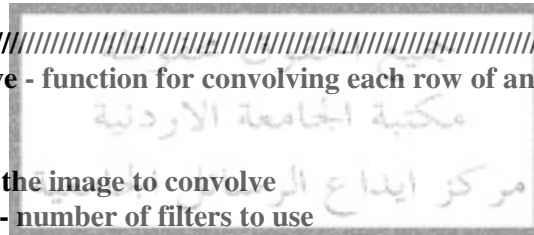
ang1=((ang>= 0)&(ang<pi/2)).*1; % angles in the first quarter quantized to 1
ang2=((ang>=pi/2)&(ang<pi)).*2; % angles in the second quarter quantized to 2
ang3=((ang<0)&(ang >= -pi/2)).*3; % angles in the third quarter quantized to 3
ang4=((ang <-pi/2)&(ang>=-pi)).*4; % angles in the forth quarter quantized to 4
angnet1=ang1+ang2+ang3+ang4;
angnet2=angnet1+imresize(noise_array,[3 120]);
iris_template=angnet2;
%-----save the results -----
Encoding_file = [cd,'\Encoded_irises\' ,imgis,' _' ,img_part,' _' ,img_index,'
encoding.mat' ];
%save(Encoding_file,' iris_template' );
end
end
end
disp(' End' )
%/////////////////////////////////////////////////////////////////

*****

%/////////////////////////////////////////////////////////////////
% gaborconvolve - function for convolving each row of an image with 1D log-
% Gabor filters
% Arguments:
% im - the image to convolve
% nscale - number of filters to use
% minWaveLength - wavelength of the basis filter
% mult - multiplicative factor between each filter
% sigmaOnf - Ratio of the standard deviation of the Gaussian describing
% the log Gabor filter's transfer function in the frequency
% domain to the filter center frequency.
%
% Output:
% E0 - a 1D cell array of complex valued convolution results
%
% Author:
% Original 'gaborconvolve' by Peter Kovesi, 2001
% Heavily modified by Libor Masek, 2003
% masekl01@csse.uwa.edu.au
% School of Computer Science & Software Engineering
% The University of Western Australia
% November 2003
%/////////////////////////////////////////////////////////////////

function [EO, filtersum] = gaborconvolve(im, nscale, minWaveLength, mult, ...
    sigmaOnf)
[rows cols] = size(im);
filtersum = zeros(1,size(im,2));
EO = cell(1, nscale); % Pre-allocate cell array
ndata = cols;
if mod(ndata,2) == 1 % If there is an odd No of data points

```



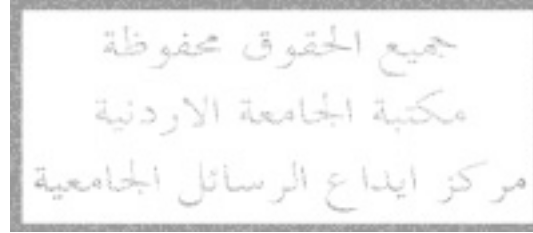
```

    ndata = ndata-1;           % throw away the last one.
end
logGabor = zeros(1,ndata);
result = zeros(rows,ndata);
radius = [0:fix(ndata/2)]/fix(ndata/2)/2; % Frequency values 0 - 0.5
radius(1) = 1;
wavelength = minWaveLength; % Initialize filter wavelength.
for s = 1:nscale,           % For each scale.
    % Construct the filter - first calculate the radial filter component.
    fo = 1.0/wavelength;      % Centre frequency of filter.
    rfo = fo/0.5;             % Normalised radius from centre of frequency plane
    % corresponding to fo.
    logGabor(1:ndata/2+1) = exp((-log(radius/fo)).^2) / (2 * log(sigmaOnf)^2));
    logGabor(1) = 0;
    filter = logGabor;
    filtersum = filtersum+filter;
    % for each row of the input image, do the convolution, back transform
    for r = 1:rows           % For each row
        signal = im(r,1:ndata);
        imagefft = fft( signal);
        result(r,:) = ifft(imagefft .* filter);
    end
    % save the ouput for each scale
    EO{s} = result;
    wavelength = wavelength * mult; % Finally calculate Wavelength of next filter
end % ... and process the next scale
filtersum = fftshift(filtersum);
%////////////////////////////////////////////////////////////////////////////////////////////////////

```

## Appendix E

**This appendix contains the program source codes for iris matching or comparing( intra-class comparisons and inter-class comparisons)**



```

%////////////////////////////////////
% compare_interclass
% This program:compares the templates of different irises by calculating the
% Matching Ratio (MR) between them.
% This program is designed for CASIA iris database (ver 1.0)
%////////////////////////////////////
close all
clear all
clc
disp(' Start... ' )
%---- determine the input iris templates that were resulted from encoding process-----
from_Ref=1; % from the reference image
to_Ref=107; % to the reference image
to_img=108; % to image
%-----find the path of the inputs and read it -----
interClass_dir=cd;
intercomparings_dir=[cd,'\interClass_comparisons' ];
Recognition_dir=interClass_dir(1,1:length(cd)-
length(' Iris_ComparingInterClass_Comparing' ));
for imgi=from_Ref:to_Ref
img_counter=0;
clear comparing_result;
ref_img=int2str(imgi);
Limgis=length(ref_img);
    if Limgis==1
        ref_img=strcat(' 00' ,ref_img);
    end
    if Limgis==2
        ref_img=strcat(' 0' ,ref_img);
    end
    disp(ref_img);
%-----
compar_from=imgi+1;
compar_to=to_img;
%-----
Encoding_file1 = [Recognition_dir,' Iris_EncodingEncoded_irises\' ,ref_img,' _1-1
encoding.mat' ];
load(Encoding_file1);
template1=iris_template;
Encoding_file2 = [Recognition_dir,' Iris_EncodingEncoded_irises\' ,ref_img,' _1-2
encoding.mat' ];
load(Encoding_file2);
template2=iris_template;
Encoding_file3 = [Recognition_dir,' Iris_EncodingEncoded_irises\' ,ref_img,' _1-3
encoding.mat' ];
load(Encoding_file3);
template3=iris_template;
Encoding_file4 = [Recognition_dir,' Iris_EncodingEncoded_irises\' ,ref_img,' _2-1
encoding.mat' ];
load(Encoding_file4);

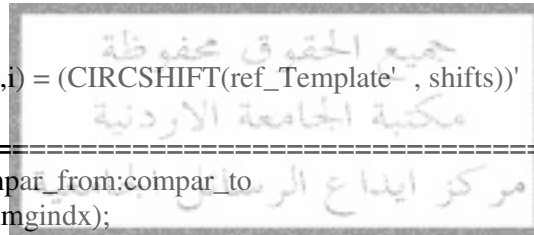
```

```

template4=iris_template;
Encoding_file5 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,ref_img,' _2_2
encoding.mat' ];
load(Encoding_file5);
template5=iris_template;
Encoding_file6 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,ref_img,'_2_3-
encoding.mat' ];
load(Encoding_file6);
template6=iris_template;
Encoding_file7 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,ref_img,' _2_4
encoding.mat' ];
load(Encoding_file7);
template7=iris_template;
ref_Template=[template1;template2;template3;template4;template5;template6;template
7];
w=size(template1,1);
%===== shift the ref_Template=====
i=0;
for shifts=-4:4
i=i+1;
ref_Templates(:,i) = (CIRCSHIFT(ref_Template' , shifts))' ;
end
%=====
for imgindx=compar_from:compar_to
imgis=int2str(imgindx);
Limgis=length(imgis);
if Limgis==1
imgis=strcat(' 00' ,imgis);
end
if Limgis==2
imgis=strcat(' 0' ,imgis);
end

Encoding_file1 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _1-1
encoding.mat' ];
load(Encoding_file1);
template1=iris_template;
Encoding_file2 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _1-2
encoding.mat' ];
load(Encoding_file2);
template2=iris_template;
Encoding_file3 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _1-3
encoding.mat' ];
load(Encoding_file3);
template3=iris_template;
Encoding_file4 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _2-1
encoding.mat' ];
load(Encoding_file4);
template4=iris_template;

```





```

Encoding_file5 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _2_2
encoding.mat' ];
load(Encoding_file5);
template5=iris_template;
Encoding_file6 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _2_3
encoding.mat' ];
load(Encoding_file6);
template6=iris_template;
Encoding_file7 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _2_4
encoding.mat' ];
load(Encoding_file7);
template7=iris_template;
Templates=[template1;template2;template3;template4;template5;template6;template7];
%-----
compare_counter=0;
for i1=1:w:size(ref_Template,1)
    T1=ref_Templates(i1:i1+w-1,,:);
    for i2 =1:w:size(Templates,1)
        T2=Templates(i2:i2+w-1,:);
        compare_counter=compare_counter+1;
%-----Matching process-----
        MRmax=0;
        for i3=1:9
            d=abs(T1(:,i3)-T2);
            MR=length(d(d==0))/length(d(d>=0));
            if MR>MRmax
                MRmax=MR;
            end
        end
        Matching_result(compare_counter)=MRmax;
%-----
    end
end
%-----save the results -----
img_counter=img_counter+1;
comparing_result(img_counter,:)=Matching_result;
inter_file=[intercomparings_dir,' REF' ,int2str(imgi),' To' ,int2str(to_img),'
intercomparing.mat' ];
save(inter_file,' comparing_result' );
end
end
disp(' End' )
%////////////////////////////////////

```

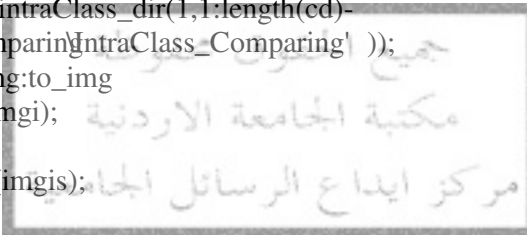
\*\*\*\*\*

```

%//////////////////////////////////////////////////////////////////
% compare_intraclass
% This program:compare the templates of the same irises by calculating the
% Matching Ratio(MR) between them.
% This program is designed for CASIA iris database (ver 1.0)
%//////////////////////////////////////////////////////////////////

close all
clear all
clc
img_counter=0;
disp(' Start..' )
%--- determine the input iris templates that were resulted from encoding process----
from_img=1;
to_img=108;
%----- find the path of these inputs and read it-----
intraClass_dir=cd;
intracomparings_dir=[cd,'\intraClass_comparisons\ '];
Recognition_dir=intraClass_dir(1,1:length(cd)-
length(' Iris_Comparing\intraClass_Comparing' ));
for imgi=from_img:to_img
    imgis=int2str(imgi);
    disp(imgis)
    Limgis=length(imgis);
    if Limgis==1
        imgis=strcat(' 00' ,imgis);
    end
    if Limgis==2
        imgis=strcat(' 0' ,imgis);
    end
    Encoding_file1 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _1-1
encoding.mat' ];
    load(Encoding_file1);
    template1=iris_template;
    Encoding_file2 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _1-2
encoding.mat' ];
    load(Encoding_file2);
    template2=iris_template;
    Encoding_file3 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _1-3
encoding.mat' ];
    load(Encoding_file3);
    template3=iris_template;
    Encoding_file4 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _2-1
encoding.mat' ];
    load(Encoding_file4);
    template4=iris_template;
    Encoding_file5 = [Recognition_dir,' Iris_Encoding\Encoded_irises\' ,imgis,' _2-2
encoding.mat' ];
    load(Encoding_file5);
    template5=iris_template;

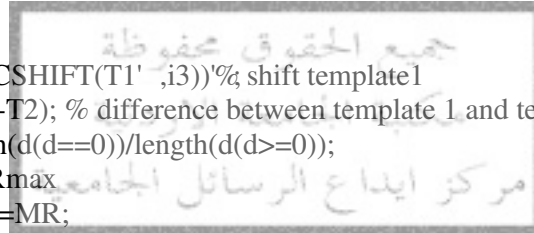
```



```

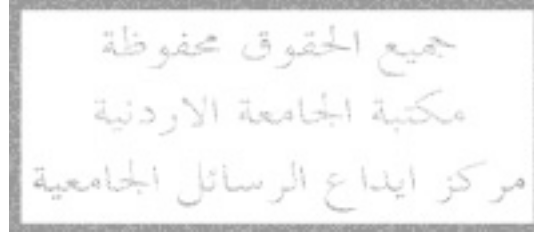
Encoding_file6 = [Recognition_dir, ' Isi_Encoding\Encoded_irises\' ,imgis,' _2_3
encoding.mat' ];
load(Encoding_file6);
template6=iris_template;
Encoding_file7 = [Recognition_dir, ' Iris_Encoding\Encoded_irises\' ,imgis,' _2_4
encoding.mat' ];
load(Encoding_file7);
template7=iris_template;
Templates=[template1;template2;template3;template4;template5;template6;template7];
w=size(template1,1);
%-----
compare_counter=0;
for i1=1:w:size(Templates,1)
    T1=(Templates(i1:i1+w-1,:));
    for i2 =i1+w:w:size(Templates,1)
        T2=(Templates(i2:i2+w-1,:));
        compare_counter=compare_counter+1;
%-----Matching process-----
MRmax=0;
for i3=-4:4
    T1s=(CIRCSHIFT(T1' ,i3))'% shift template1
    d=abs(T1s-T2); % difference between template 1 and template2
    MR=length(d(d==0))/length(d(d>=0));
    if MR>MRmax
        MRmax=MR;
    end
end
Matching_result(compare_counter)=MRmax;
%-----
end
end
%----- save the results -----
img_counter=img_counter+1;
comparing_result(img_counter,:)=Matching_result;
intra_file=[intracomparings_dir,' From' ,int2str(from_img),' To' ,int2str(to_img),'
intracomparing.mat' ];
%save(intra_file,' comparing_result' );
end
disp(' End' )
%//////////

```



## Appendix F

**This appendix contains the program source codes for results analysis**



```

%////////////////////////////////////
% check_interclass
% This program:rejects the defected images according to the predetermined
% noise threshold, this rejection is done by changing the related values in the
% interClass_comparisons to NaNs, so when we analyze the results, these NaN
% values do not effect on the analyzing process.
% This program is designed for CASIA iris database (ver 1.0)
%////////////////////////////////////
clc
close all
clear all
disp(' Start...' )
%----- determine the input images(1-107) -----
from_img=1;
to_img=107;
%-----determine the upper and lower noise thresholds -----
up_threshold=50;
low_threshold=50;
%----- find the paths of the inputs and read it -----
Analysis_dir=cd;
Recognition_dir=Analysis_dir(1,1:length(cd)-length(' Results_Analysis' ));
interClass_dir1=[Recognition_dir,' Iris_Comparing\interClass_Comparing\interClass_co
mparisons' ];
for k=1:107
load([interClass_dir1,' REF' ,int2str(k),' To108rcomparing' ]); % load the result
interchecked_dir=[cd,'\interClass_checked\ '];
inter_file=[interchecked_dir,' REF' ,int2str(k),' To'-int2str(k),'\interClass_Comparing.mat' ];
save(inter_file,' comparing_result' );
end
%-----
interClass_dir=cd;
interchecked_dir=[cd,'\interClass_checked\ '];
%-----
for imgi=from_img:to_img
    subimg_idx=0;
    imgis=int2str(imgi);
    disp(imgis)
    Limgis=length(imgis);
    if Limgis==1
        imgis=strcat(' 00' ,imgis);
    end
    if Limgis==2
        imgis=strcat(' 0' ,imgis);
    end
end
for img_parts=1:2
    img_part=int2str(img_parts);
    if img_parts==1
        img_end=3;
    else
        img_end=4;
    end
end

```

```

end
for img_index=1:img_end
    img_index=int2str(img_index);
    read_img= strcat(imgis,'\ ' ,img_part' ,imgis,' _' ,img_part,' _' ,img_index,' .bmp' );
%-----
    img_indx=imgi;
    subimg_indx=subimg_indx+1;
%-----
segmentation_file =
[Recognition_dir,'\Iris_Segmentation' \Segmented_irises\ ' ,imgis,' _' ,...
    img_part,' _img_index,'-segmentation.mat' ];
load(segmentation_file);
% ----- check the noise percent -----
img_noise=and(maskmatrix1,maskmatrix2);
[iup,jup]=find(img_noise(1:rL,:)==1);
[ilow,jlow]=find(img_noise(rL:size(outputimg,1),:)==1);
noisarea_up=length(iup);
noisarea_low=length(ilow);
comparisonarea=(pi*(radiaL^2)-pi*(an^2))/2;
noispercent_up=(noisarea_up/comparisonarea)*100;
noispercent_low=(noisarea_low/comparisonarea)*100;
noise_chek=0;
if (noispercent_up > up_threshold)|(noispercent_low > low_threshold )
    noise_chek=1;
end
%----- keep only the chosen result according to noise percents-----
if noise_chek == 1
load([interchecked_dir,' REF' ,int2str(imgi),' To-108rcomparing' ]); % load the result
comparing_result(:,(subimg_indx-1)*7+1:subimg_indx*7)=nan;
inter_file=[interchecked_dir,' REF' ,int2str(imgi),' To-108rcomparing.mat' ];
save(inter_file,' comparing_result' );
for i=1:imgi-1
load([interchecked_dir,' REF' ,int2str(i),' To-108rcomparing' ]); % load the result
for j=subimg_indx:7:49
comparing_result(imgi-i,j)=nan;
end
inter_file=[interchecked_dir,' REF' ,int2str(i),' To-108rcomparing.mat' ];
save(inter_file,' comparing_result' );
end
end
end
end
disp(' End' )
%////////////////////////////////////
*****

```

```

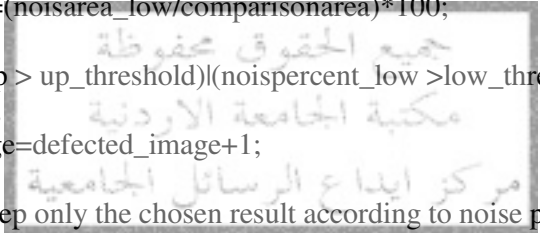
%/////////////////////////////////////////////////////////////////
% check_intraClass
% This program:rejects the defected images according to the predetermined
% noise threshold, this rejection is done by changing the related values in the
% intraClass_comparisons to NaNs, so when we analyze the results, these NaN
% values do not effect on the analyzing process.
% This program is designed for CASIA iris database (ver 1.0)
%/////////////////////////////////////////////////////////////////
clc
close all
clear all
disp(' Start...' )
%----- determine the input images(1-107) -----
from_img=1;
to_img=1;
%-----determine the upper and lower thresholds-----
up_threshold=50;
low_threshold=50;
%-----find the paths of the inputs and read it -----
Analysis_dir=cd;
Recognition_dir=Analysis_dir(1,1:length(cd)-length(' Results_Analysis' ));
intraClass_dir=[Recognition_dir,' Iris_Comparing\intraClass_Comparing\intraClass_co
mparisons' ];
load([intraClass_dir,' From1To108intraComparing' ]); % load the result
intrachecked_dir=[cd,'\intraClass_checked' ];
intra_file=[intrachecked_dir,' From1To108intraComparing.mat' ];
save(intra_file,' comparing_result' );
%-----
intraClass_dir=cd;
intrachecked_dir=[cd,'\intraClass_checked' ];
defected_image=0;
%-----
for imgi=from_img:to_img
    subimg_idx=0;
    imgis=int2str(imgi);
    disp(imgis)
    Limgis=length(imgis);
    if Limgis==1
        imgis=strcat(' 00' ,imgis);
    end
    if Limgis==2
        imgis=strcat(' 0' ,imgis);
    end
for img_parts=1:2
    img_part=int2str(img_parts);
    if img_parts==1
        img_end=3;
    else
        img_end=4;
    end
end

```

```

for img_index=1:img_end
    img_index=int2str(img_index);
    read_img= strcat(img_dir,'_img_part',img_index,'_',img_index,'.bmp' );
%-----
    img_index=img_index;
    subimg_index=subimg_index+1;
%-----
segmentation_file = [Recognition_dir,'Iris_Segmentation' \Segmented_irises' ,...
    img_index,'_',img_index,'_',img_index,'segmentation.mat' ];
load(segmentation_file);
% ----- check the noise percent-----
img_noise=and(maskmatrix1,maskmatrix2);
[iup,jup]=find(img_noise(1:rL,:)==1);
[ilow,jlow]=find(img_noise(rL:size(outputimg,1),:)==1);
noisarea_up=length(iup);
noisarea_low=length(ilow);
comparisonarea=(pi*(radiaL^2)-pi*(an^2))/2;
noispercent_up=(noisarea_up/comparisonarea)*100;
noispercent_low=(noisarea_low/comparisonarea)*100;
noise_chek=0;
if (noispercent_up > up_threshold)|(noispercent_low > low_threshold )
    noise_chek=1;
    defected_image=defected_image+1;
end
%----- keep only the chosen result according to noise percents-----
if noise_chek==1
load([intrachecked_dir,' From1To108Intracomparing' ]); % load the result
if subimg_index==1
    comparing_result(img_index,1:6)=nan;
end
if subimg_index==2
    comparing_result(img_index,7:11)=nan;
    comparing_result(img_index,1)=nan;
end
if subimg_index==3
    comparing_result(img_index,12:15)=nan;
    comparing_result(img_index,2)=nan;
    comparing_result(img_index,7)=nan;
end
if subimg_index==4
    comparing_result(img_index,16:18)=nan;
    comparing_result(img_index,3)=nan;
    comparing_result(img_index,8)=nan;
    comparing_result(img_index,12)=nan;
end
if subimg_index==5
    comparing_result(img_index,19:20)=nan;
    comparing_result(img_index,4)=nan;
    comparing_result(img_index,9)=nan;
    comparing_result(img_index,13)=nan;

```





```

    comparing_result(imgi,16)=nan;
end
if subimg_indx==6
    comparing_result(imgi,21)=nan;
    comparing_result(imgi,5)=nan;
    comparing_result(imgi,10)=nan;
    comparing_result(imgi,14)=nan;
    comparing_result(imgi,17)=nan;
    comparing_result(imgi,19)=nan;
end
if subimg_indx==7
    comparing_result(imgi,6)=nan;
    comparing_result(imgi,11)=nan;
    comparing_result(imgi,15)=nan;
    comparing_result(imgi,18)=nan;
    comparing_result(imgi,20)=nan;
    comparing_result(imgi,21)=nan;
end

intra_file=[intrachecked_dir,' From1To108intracomparing.mat' ];
save(intra_file,' comparing_result' );

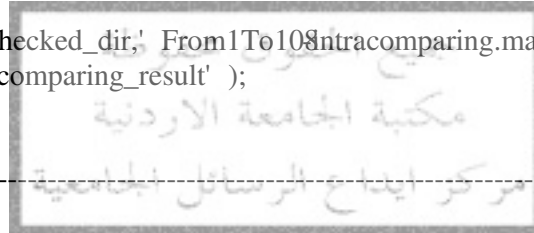
end
%-----
end
end
end
disp([' Number of defected images = ' ,num2str(defected_image)])
disp(' End' )
%////////////////////////////////////
%*****

%////////////////////////////////////
% analyze_results
% This program: plots the distributions of inter and intra classes, and the
% Receiver Operating Characteristic(ROC)curve, also it calculates ' decidability' .
% This program is designed for CASIA iris database (ver 1.0)
%////////////////////////////////////

clc
close all
clear all
disp(' Start...' )

%-----
interClass_dir=cd;
interchecked_dir=[cd,'\interClass_checked' ];
intrachecked_dir=[cd,'\intraClass_checked' ];
Analysis_dir=cd;
Recognition_dir=Analysis_dir(1,1:length(cd)-length(' Results_Analysis' ));

```



```

%-----intraClass_comparisons -----
intraClass_file = [intrachecked_dir,' From1To108intracomparing.mat' ];
load(intraClass_file);
intra_results=comparing_result;
%-----interClass_comparisons-----
a=1;
for i=1:107
interClass_file = [interchecked_dir,' REF' ,int2str(i),' To108rcomparing.mat' ];
load(interClass_file);
interclass=comparing_result;
b=a+size(interclass,1)-1;
intra_results(a:b,:)=interclass;
a=b+1;
end
%----- Display the results-----
disp('----- Results -----' )
A(find(intra_results>=0))=intra_results(find(intra_results>=0));
A(A==0)=nan;
B(find(inter_results>=0))=inter_results(find(inter_results>=0));
B(B==0)=nan;
hist(B,50);title(' InteClass' );
V = AXIS;
AXIS([0 1 V(3) V(4)])
xlabel(' Matching Ratio MR' );
ylabel(' Frequency' );
figure;hist(A,50);title(' IntraClass' );
V = AXIS;
AXIS([0 1 V(3) V(4)])
xlabel(' Matching Ratio MR' );
ylabel(' Frequency' );
%-----ROC curve -----
i=0;
for threshold=0:0.01:1
i=i+1;
[n1,m1]=find(inter_results > threshold);
FAR(i)=(length(n1)/length(inter_results(inter_results>=0)))*100;
[n2,m2]=find(intra_results < threshold);
FRR(i)=(length(n2)/length(intra_results(intra_results>=0)))*100;
thresh(i)=threshold;
end
figure;
semilogx(FAR,FRR,' black' );
AXIS([0.001 100 0 25])
xlabel(' False Acceptance Rate(%)' );
ylabel(' False Rejection Rate(%)' );
%CAR=100-FRR;
%semilogx(FAR,CAR,' black' );
%AXIS([0 10 90 100])
%ylabel(' Correct Acceptance Rate(%)' );
%save(cd,' FAR' , ' CAR' )

```



```

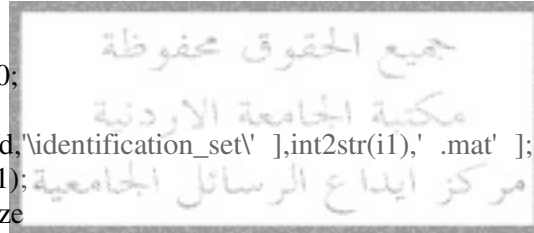
    imgis=strcat(' 00' ,imgis);
end
if Limgis==2
    imgis=strcat(' 0' ,imgis);
end
for img_parts=1:2
    img_part=int2str(img_parts);
    if img_parts==1
        img_end=3;
    else
        img_end=4;
    end
    for img_indexs=1:img_end
        img_index=int2str(img_indexs);
        read_img= strcat(imgis,'\ ' ,img_part" ,imgis,' _' ,img_part,' _' ,img_index,' .bmp' );
        Encoding_file =
        [Recognition_dir,' Iris_Encoding\Encoded_irises_test1\' ,imgis,' _' ,img_part,' _' ,img_index
        ,'-encoding.mat' ];
        load(Encoding_file);
        % ----- check the noise percent -----
        segmentation_file =
        [Recognition_dir,'Iris_Segmentation' \Segmented_irises\' ,imgis,' img_i part,' _' ,img_ind
        ex,'-segmentation.mat' ];
        load(segmentation_file);
        img_noise=and(maskmatrix1,maskmatrix2);
        [iup,jup]=find(img_noise(1:rL,:)==1);
        [ilow,jlow]=find(img_noise(rL:size(outputimg,1),:)==1);
        noisarea_up=length(iup);
        noisarea_low=length(ilow);
        comparisonarea=(pi*(radiaL^2)-pi*(an^2))/2;
        noispercent_up=(noisarea_up/comparisonarea)*100;
        noispercent_low=(noisarea_low/comparisonarea)*100;
        if (noispercent_up > up_threshold)|(noispercent_low >low_threshold )
            % reject the image.
        else
            set_size=set_size+1;
            inter_file1=[[cd,'identification_set\' ],int2str(imgnum),' _' , int2str(set_size),' .mat' ];
            save(inter_file1,'iris_template' );
        end
    end
end
end
%----- save only the accepted images-----
if (set_size ==0)
    imgnum=imgnum-1;
else
    if (set_size==1)
        delete(inter_file1);% to ensure that at least one image for training and one for
        % test
        imgnum=imgnum-1;
    else

```

```

inter_file2=[[cd,'\identification_set' ],int2str(imgnum),' .mat' ];
save(inter_file2,' set_size' );
end
end
%-----
end
disp(' End' )
%////////////////////////////////////
                        *****
%////////////////////////////////////
% identification_result
% This program: finds the Correct Identification Rate (CIR), by specifying
% one image of each class as training samples, and the rest of images as test
% samples.
%////////////////////////////////////
clc
close all
clear all
disp(' Start' )
compare_count=0;
for i1=1:101
inter_file1=[[cd,'\identification_set' ],int2str(i1),' .mat' ];
load(inter_file1);
for i2=2:set_size
compare_count=compare_count+1;
MRmax=0;
inter_file2=[[cd,'\identification_set' ],int2str(i1),' _' ,int2str(i2),' .mat' ];
load(inter_file2);
%-----shift iris template code by -4 to +4 -----
j=0;
for shifts=-4:4
j=j+1;
iris_template1(:,j) = (CIRCSHIFT(iris_template' , shifts))' ;
end
%-----
for i3=1:101
inter_file3=[[cd,'\identification_set' ],int2str(i3),' _1' , ' .mat' ];
load(inter_file3);
iris_template2=iris_template;
%-----compare and find MR -----
for k=1:9
d=abs(iris_template1(:,k)-iris_template2);
MR=length(d(d==0))/length(d(d>=0));
if MR>MRmax
MRmax=MR;
img=i3;
end
end
end
%-----

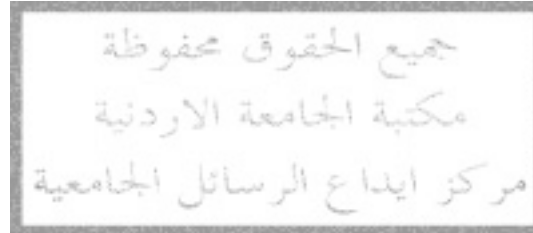
```

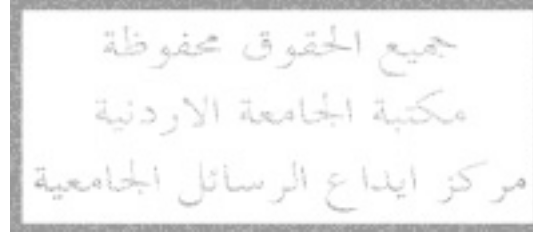


```

end
compare_count          % as counter for waiting
MRresult(compare_count)= MRmax;  % to be used later
if (img==i1)&(MRmax > 0.427)      % put threshold here
    Result(compare_count)=1;
else
    Result(compare_count)=0;
end
end
end
disp(' Identification Rate % = ' )
Ide_Rate =(length(Result(Result==1))/length(Result))*100
save(' Ide_Rate');
%-----
disp(' End' )
%////////////////////////////////////

```





The Hough Transform and the Compass gradients

Daugmans rubber sheet model

1D log-Gabor filter

Hamming distance

Matching ratio

( MATLAB6.1

(CASIA

(Receiver Operating Characteristic  
(Daugman, Wildes *et al.* ,L. Ma *et al.*

جميع الحقوق محفوظة  
مكتبة الجامعة الاردنية  
مركز ايداع الرسائل الجامعية