Eye Shape Detection Methods Based on Eye Structure Modeling and Texture Analysis for Interface Systems

January, 2015

DOCTOR OF ENGINEERING

Mohammad Amro Jamal AL-DIBAJA

TOYOHASHI UNIVERSITY OF TECHNOLIOGY

Doctoral Thesis

Eye Shape Detection Methods Based on Eye Structure Modeling and Texture Analysis for Interface Systems

Author:

Mohammad Amro Jamal ALDIBAJA

Supervisor:

Prof. Shinichi SUZUKI

A thesis submitted in fulfillment of the requirements
for the degree of Doctoral of Engineering
in the

Robotics and Mechatronics Laboratory

Department of Mechanical Engineering

Toyohashi University of Technology
Toyohashi, Japan
January 2015

TOYOHASHI UNIVERSITY OF TECHNOLOGY

Abstract

Graduate School of Engineering
Department of Mechanical Engineering

Doctor of Engineering

Eye Shape Detection Methods Based on Eye Structure Modeling and Texture Analysis for Interface Systems

by

Mohammad Amro Jamal AL-DIBAJA

Human-Computer Interaction systems (HCI) are being an attractive field for both researchers and companies in order to increase the human life quality and safety. This is because of the revolution in producing mobile devices that have the capabilities to connect the internet and process information in a very short timeframe.

Eye is considered as a very promising interactive component. It can be employed to achieve many tasks that usually done by hands as well as it has a faster behavior and response. For example, suppose that many products are randomly distributed on a production line and an operator wants to pick one of them by a manipulator robot. The conventional strategy is whether to use a joystick to control the robot's clip or to move the courser on a PC screen and click on the desired object. The robot will then generate the motion matrix to get the location of the object.

More sophisticated approach is to use eyes. The operator should only look at the object whether on the screen or directly in the real world on the production line.

This thesis aims to develop and implement three methods to automatically extract the eye features. The eye features are represented by the term “eye shape”. The eye shape consists of a set of points called eye shape points. The eye shape points are distributed equally to represent the two eye corners and the two eyelids. The eye shape identifies the eye state (open, nearly close and in-between). In addition, it describes the appearance parameters such as scale, rotation angle and translation in eye images.

In order to detect the eye shape, Active Shape Model (ASM) is developed. The developments are done in terms of enhancing initial estimation of the eye contour in a testing eye image using an object recognition technique. The object recognition technique tries to understand the topological eye structure in the testing image. Based on this understanding, the eye state and appearance parameters are estimated and an initial eye shape is suggested.

Searching on eye shape points is also enhanced by describing the local structure around the eye shape points using Principal Component Analysis (PCA). The capabilities of PCA in terms of generalization have increased the robustness of detecting the locations under different lighting conditions.

The improved ASM outperforms the standard/original ASM with increasing accuracy and reducing detection time.

In the second approach, a completely new method in the domain of object shape detection is proposed. Log-polar transform (LPT) is utilized to convert the change of eye scales and rotations in Cartesian coordinate system into translations. Consequently, the task of detecting x,y coordinates of eye shape points is facilitated into detecting only $\log\rho$ coordinates in Log-Polar domain. LPT maps the pixels in equal size and hence the eye geometrical details are preserved regardless the eye size/scale and state in testing images.

In addition, the existence of eye components such as pupil, iris, eyelids and sclera makes the color variation between eye and skin

very sharp. This variation is measured by applying PCA on RGB color channels of the testing image against wide range of changing of lighting conditions. Based on these properties of eye representation by PCA and LPT, the eye shape is extracted regardless the change of scale, rotation, translation, eye states and lighting conditions.

This new method has performed very high accuracy with significantly reducing the detection time compared to ASM and Deformable Template (DT). Therefore, it is very appropriate to be employed in real time systems. Moreover, the design has proven that Log-Polar domain should be the baseline of implementing any eye shape detection technique.

In the third method, as long as LPT has been proven as the initial step of any eye shape detection technique, a recurrent neural network is trained to model the pixel profile in each row of Log-polar eye images. Each pixel profile (row) contains a few pixels that represent the eye contour. The pixel values in a pixel profile vary smoothly and there is no sudden or rapid change exists. Based on this fact, the pixel value of the eye contour in a pixel profile is changed by creating a sudden peak. The recurrent network is trained to predict the location of this peak based on some past outputs and previous inputs of pixel values.

By utilizing the above training approach for eye shape detection, a Log-polar eye image is described by a fixed number of pixel profiles. The locations of the eye contour in these profiles are highlighted by creating peaks. The recurrent network is trained using different Log-polar eye images. This strategy has significantly enhanced the recurrent network prediction ability because the recurrent network has learned the patterns of pixel values as well as it has modeled the eye contour by learning the different frequencies of peaks' locations.

The experimental results have verified a robust performance of the proposed method based RNN compared to ASM and DT

performances. On the other hand, it emphasize that using neural network is very promising for detecting the eye shape and more investigations should be conducted.

TOYOHASHI UNIVERSITY OF TECHNOLOGY

Graduate School of Engineering
Department of Mechanical Engineering

Eye Shape Detection Methods Based on Eye Structure Modeling and Texture Analysis for Interface Systems

Doctor of Engineering Thesis
Of

Mohammad Amro Jamal AL-DIBAJA

Committee Members:

Prof. Dr. Shinichi Suzuki ----------

Prof. Dr. Tetsuo Miyake ----------

Prof. Dr. Zhong Zhang ----------

Prof. Dr. Naoki Uchiyama ----------

Acknowledgment

I would like to express my deep appreciation to Prof. Shinichi Suzuki for his guidance, encouragement and support during my study. I have learned from him the key-points and principles of involving in scientific researches such as writing scientific papers, identifying and addressing problems, investigating solutions, conducting experiments and presenting results scientifically and academically. These principles are the most important issues for any researcher regardless the studying field. On the other hand, I would like to thank him for his kind and friendly dealing with me. This manner has extremely facilitated the university/campus daily life and initialized stable conditions for concentrating on my study.

I am grateful to the committee members, Prof. Tetsuo Miyake, Prof. Naoki Uchiyama and Prof. Zhong Zhang for their valuable supports, guidance, comments and discussions in order to edit and modify this thesis properly as well as for their priceless notices on my semi-defense presentation.

I would like to thank Prof. Naoki Uchiyama for allowing me to participate in his Lab activities that provided me to take a close look on his domain.

Furthermore, I would like to thank the staff of Toyohashi University of

Technology for their kind and friendly dealing with international students. Connected thanks to the financial committee members in the University for approving me to work as a research assistant during my PhD study.

I extend my heartfelt thanks to my parents and brother for their endless support, sacrifice and love. I also thank my relative and close friends for their encouragement. In addition, many thanks return to the staff of Syrian Embassy in Japan and the staff of Tishreen University and Mechatronics Departments in Latakia-Syria for their kind cooperation and understanding.

Last but not least, I want to thank Toyohashi Muslim Community and all friends those I met during my study (especially Egyptian friends).

Mohammad Amro Jamal AL-DIBAJA

Table of Content

List of Figures

List of Definitions

1. **The pixel:** Pixel is the smallest element in an image and it has a square shape. The pixel size determines the image quality. As much as the size is being smaller, the details are being clearer and the image resolution is increased. A pixel has a value, called the pixel value, to represent the intensity in range of [0 to 255].
2. **The image:** From a classical point of view, image is something we see by our vision system. From a technical and deeper point of view, image consists of a set of pixels. The *distribution* of these pixels identifies the image and characterizes the object features such as edges, vertex, corners, surface, depth, distance, size…etc. On the other hand, *pixels' values* represent lighting conditions, luminance, contrast, color, correlation,...etc. Consequently, an image describes thousands of words and expressions that could be recognized differently from one person to another according to the paid attention.
3. **Gaze point:** The point which the user is looking at in the 3D environment [11]. This point is estimated by two steps. The first step is to detect the pupil location based on the eye angle and some eye shape points. The second step is to create a 3D model of the environment (ex. depth map) and correlate the pupil coordinates with the 3D model coordinates [12][13].
4. **Testing image**: is an eye image that contained in a database and used for evaluating the performance of an eye shape detection technique. If the implementation of the detection technique has learning and detection phases, the testing image is usually not used in the training phase.
5. **Exact eye shape:** is the eye contour in a testing image. According to the thesis strategy, eye

contour is represented by a set of *N* points as shown in Fig. 1.1(b). Hence, any point on the eye contour is considered as an exact eye shape point.

6. **Detected eye shape:** is the eye shape of a testing image that detected by an eye shape detection technique. The points of the detected eye shape are expected to be exact points and accurately describe the eye contour.
7. **Static system:** the system that detection time or time consumption is not so important issue such as constructing a panoramic image from different photos. Thus, the quality of results and the high accuracy achievement are most important.
8. **Real-time/on-line system:** the system that detection time is the most important issue because the results must be obtained as fast as possible for allowing the process to run through time. Accuracy is also an important parameter and a tradeoff with detection time must be done and adjusted according to the applications.
9. **Iterative searching strategy**: This strategy is widely used in image processing for shape matching and detection. ASM tries to minimize the error between the fitted eye shape and the exact eye shape in a testing image iteratively. Therefore, a bunch of steps will be repeated until convergence is achieved and the next testing image is then processed.
10. **Covariance Matrix**: it describes the relationship between variables. In other words, it describes how each element changes from the mean point with respects to the others. The sign of elements is very important because it expresses the variable directions. If an element has a large value with positive sign, that means the variables which yield this element change in one direction. Therefore, they are strongly correlated and the relationship is linear. On the other hand, the negative sign with a large value indicates that the variables move in opposite directions. Regardless the sign and when the value is small, that means one variable at least is close to the mean point.

Abbreviations

PCA	Principal Component Analysis
LPT	Log-Polar Transform
FFT	Fast Fourier Transform
RNN	Recurrent Neural Network
FFN	Feed Forward Network
LPEI	Log-Polar Eye Image
ASM	Active Shape Model
DT	Deformable Template
HCI	Human Computer Interface
AAM	Active Appearance Model
ACM	Active Contour Model
RGB	Red Green Blue Color System
SVD	Singular Value Decomposition
KNN	K-Nearest Neighbor

To who has been waiting for this moment for long time,

My mother (to her soul)

To who is sacrificing for his family,

My father

To who has supported me a lot,

My brother

To second mothers,

My aunts

To my motherlands,

Palestine & Syria

Chapter 1

Introduction

1.1. General

Human-Computer Interface (HCI) has become a very interesting topic in image processing recently [1-2]. This is because of the considerable revolution in interaction systems. Therefore, computer vision is considered as the field number one in Google Company and they have established a very big lab in China a few years ago. This promising field has imposed the researchers to redevelop many techniques that got a little attention in past such as object shape detection and feature extraction. In order to deal with such techniques, a clear definition on pixel and image concepts should be given.

The pixel: Pixel is the smallest element in an image and it has a square shape. The pixel size determines the image quality. As much as the size is being smaller, the details are being clearer and the image resolution increases. A pixel has a value, called the pixel value, to represent the intensity in range of [0 to 255].

The image: From a classical point of view, image is something we see by our vision system. From a technical and deeper point of view, image consists of a set of pixels. The *distribution* of these pixels identifies the image and characterizes the object features such as edges, vertex, corners, surface, depth, distance, size…etc. On the other hand, *pixels' values* represent lighting conditions, luminance, contrast, color, correlation,...,etc. Consequently, an image describes thousands of words and expressions that could be recognized differently from one person to

another according to the attention paid.

The essence of HCI systems can be summarized by the ability of the machine (computer) to see, understand the environment and react accordingly. Herein, we give a brief explanation on each term.

To see:

Seeing is the process of creating a digital image of the environment. The image could be obtained from a laser sensor as used for trajectory planning and robot localization [3]. Antennas receive electro-magnetic waves that are reflected by objects' bodies. These waves can be combined and processed to produce an image as in satellite systems [4]. Camera is a common device for capturing images. Several types of cameras are employed in image processing such as thermal cameras [5], infrared cameras [6], Kinect [7], webcams [9]…etc.

To understand:

After the image is being captured, understanding, analyzing and extracting significant information are the next aims. In this process stage, various tasks can be done such as detecting some specific objects, recognizing objects, creating motion map, detecting color…etc. Accordingly, many techniques and algorithms can be applied such as filtering, machine learning, image enhancements, color converting, transformations…etc.

To react:

Reacting is the interaction system's response which describes how the machine understands the environment and the events are being detected. In this stage, a prior knowledge or some databases must be charged such as a table of actions and corresponding reactions. More sophisticated approach is to use some prediction algorithms as a feedback before taking the reaction decision.

In order to implement an interface system or to develop some relevant particular

techniques, an interactive component should be selected such as face, hand, lips, eyes…etc. A proper detector is used to detect the selected component and to determine corresponding location and size. The features of the detected component are then extracted according to the application such as eye corners, eyelids and pupil location if the eye is being the selected interactive component.

The term *feature extraction* is usually used when the process is done on static images whereas *tracking* is the equivalent concept when the process is done on a sequence of images. This step is usually the most important issue in interaction systems because it should be achieved with high accuracy and low time consumption.

Many interactive techniques have been proposed for eye feature extraction with enforcing the user to wear a helmet or glasses. Such techniques are not so preferable because of their use constrains and limitations. For more convenience, we develop/propose three non-contact methods that can be used for eye feature tracking.

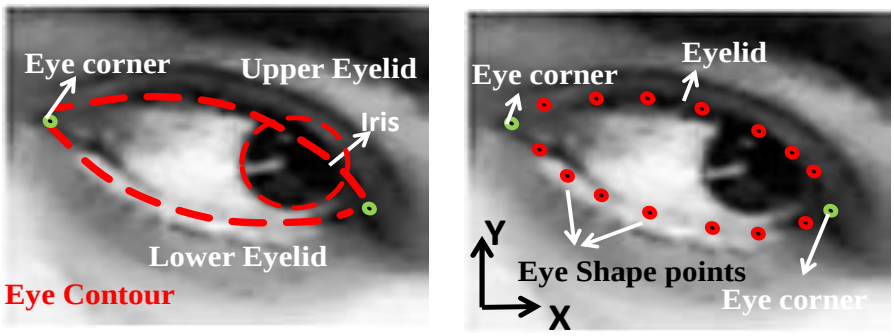
1.2. Motivation

Eye is considered as a very interesting and promising interactive component. By this motivation, designing eye feature extraction techniques is the aim of this thesis.

1.2.1. Why eye shape not eye template?

Eye features are pupil location, two eye corners and upper and lower eyelids. In this thesis, eyelids and two eye corners are referred by the eye contour as shown in Fig. 1.1. There are two strategies in literature to represent the eye contour; eye template and eye shape.

Eye contour can be described by a template as in Deformable Template (DT) [9]. The template consists of two parabolas representing eyelids and a circle representing the size of iris and the pupil location as shown in Fig. 1.1(a). On the other hand, the eye contour can be represented by a set of points as in Active Shape Model (ASM) [10]. These points form the eye shape and called eye shape points as depicted in Fig. 1.1(b). The latter strategy provides more flexibility to allow the user to detect only some particular features individually. For example, if we need to detect the two eye corners or some specific points on the eye contour, the first strategy (DT) enforces us to detect the whole template in order to obtain the points accurately as well as it costs full detection time. In contrast and by using the eye shape, a particular set of points can be detected without influencing the accuracy. Moreover, time consumption is



(a) (b)

Fig. 1.1 Eye contour representation. (a) Eye template. (b) Eye shape.

proportional to the number of points to be detected. In addition, eye template can be constructed using the eye shape as well as the pupil location and the iris size can be easily estimated inside the internal region of the eye shape. Thus, we use the eye shape strategy to represent the eye contour.

1.2.2 Applications

Many applications of eye shape detection techniques can be found, suggested and inspired. Herein, we explain briefly some applications whether they exist (with referring to a reference) or not yet. Before we go ahead, we want to define simply an expression "Gaze point" that will be mentioned many times in the application domain.

Gaze point: The point which a person is looking at in the 3D environment [11]. This point is estimated by two steps. The first step is to detect the pupil location based on the eye angle and some eye shape points. The second step is to create a 3D model of the environment (ex. depth map) and correlate the pupil coordinates with the 3D model coordinates [12][13]. Therefore, the initial step of the gaze point estimation is to detect the eye shape.

- The simplest application of the eye shape detection is to estimate the face angle using the two eye corners. Furthermore, by using the distance between these two corners, a face model can be built to initially indicate the locations of other facial features such as nose and mouth [14]. The line passing the two eye corners intersects iris in one or two points. The intersection points can be utilized to estimate the pupil location.
- Laptops and Smart phones are recently functioned to allow the user to scroll the page on the screen in four directions [15]. The key idea is to compare the pupil location with corresponding locations of two eye corners and two points at the middle of upper and lower eyelids; respectively. Another sophisticated idea can be integrated is to allow the user to

zoom in/out images on the screen based on tracking and analyzing the motion pattern of eye shape points [16][17].

- Nowadays, some TV companies try to add an intelligent touch to their products by tracking and analyzing the change of vertical distance between eyelids. This process is employed to automatically switching-off the TV when the watcher feels sleepy. In addition, the same command can be ordered using gaze point when the watcher turns his eyes away from TV for long time.
- Disabled/handicapped people who have a considerable mobility problem in using their hands to control wheelchairs can be significantly and efficiently helped to reach the desired area using Gaze point [18].
- Google has revealed recently a new application of google glasses that allows the user to take photos by performing certain patterns of blinking. Blinking can be detected by measuring and tracking the vertical distance between eye shape points in a time frame [19]. Besides and more complex, gaze point can be used in museums to provide an immediate information (identifications and historical backgrounds) about the objects that are being focused on by visitors wearing google glasses.
- A very promising and interesting field for eye shape detection techniques is driving systems. The driver fatigue can be detected by tracking the frequency of blinking and the distance between two eyelids [20]. In addition, gaze point can be employed to insure that the road events are being seen by the driver such as traffic signals, cars, pedestrian, navigation signboards…etc [21]. This problem is common in driving system domain and called "looked but NOT SEEN". The system consists of two cameras; one is fixed inside the car on front of the driver. It detects the driver's eye, extracts eye shape and estimates the pupil location. Second camera is fixed on the car's front or hood. It recognizes the objects on

street such as cars, pedestrian, traffic signals…etc. By making calibration between the two cameras and matching the information, gaze point is estimated and the seen objects are determined. These applications can be employed to efficiently avoid a lot of car accidents.

- Eye can be employed to achieve successfully many tasks that usually done by hands. Hence, hands can be turned for other tasks in-parallel. It is proven that eye moves faster than hand [22]. Suppose that a bunch of three randomly distributed products are given. We need to classify these products in three groups using a robot manipulator arm. Generally, the operator will use a joystick to control the robot's arm-tip (clip) to catch each product in 3D world. If the eye is involved in this process, the robot's hand-tip is directly guided to the product locations using gaze point. The same strategy can be significantly employed to enable the fully disabled people to control the mouse cursor on PC screen.

In conclusion, the eye features are pupil location, gaze point, eye state, blinking frequency, eye corners and iris radius/texture. Obviously, the key-point to estimate these features is to detect the eye contour/shape. The eye shape represents eye state and two eye corners. In addition, blinking frequency can be measured by tracking the vertical distance between the two mid points on the eyelids (on the eye shape). Furthermore, the pupil location and iris radius can be easily estimated in the inner region of the eye shape as well as gaze point. Therefore, we propose some methods for detecting the eye shape to be useful and more applicable in general.

1.3. History Review

Numerous methods for object shape detection have been proposed in literature as well as relevant developments. An informative survey on theses technique is given in [23]. Herein, we review briefly some principle methods that can be used to detect the eye shape. Yet, three definitions are given, testing image, exact eye shape and detected eye shape.

Testing image: is an eye image that contained in a database and used for evaluating the performance of an eye shape detection technique. If the implementation of the detection technique has learning and detection phases, the testing image is usually not used in the training phase.

Exact eye shape: is the eye contour in a testing image. According to the thesis strategy, eye contour is represented by a set of *N* points as shown in Fig. 1.1(b). Hence, any point on the eye contour is considered as an exact eye shape point.

Detected eye shape: is the eye shape of a testing image that detected by an eye shape detection technique. The points of the detected eye shape are expected to be exact points and accurately describe the eye contour.

1.3.1. Active Contour Model (snake or ACM)

Active Contour Model (ACM) is a static method that tries to detect the external object edges [24]. Better performance can be obtained when the image is not so complicated, does not contain many objects and the desired object to be detected is sharply highlighted. Therefore, the binary images which represent objects in white and background in black are the most appropriate domain for ACM applications.

From a simple point of view, the key idea is to suppose that a balloon surrounds the object randomly. The balloon is pricked to create a precise hole in the body. The air flows outside and the balloon shrinks to take and fit the exact object shape. From an engineering point of view, the balloon is described by a set of *N* points that randomly and manually labeled around the object. An energy minimization method is then applied iteratively to attract the points to the object edges. After some iterations, the final distribution of the points is considered as the detected shape. From a mathematical point of view, each point has (*x*,*y*) coordinates. An energy function

E_{snake} is used to control the movement step towards the object and the distance between points. The function consists of three terms; internal energy E_i, external energy $E_x$and constrain energy E_c.

$$E_{snake} = E_i + E_x + E_c \quad (1,1)$$

1.3.1.1. Internal energy

This term explains the relationship between the points based on their (*x,y*) coordinates. It is decomposed into elastic energy E_s and bending energyE_b. The particular energies describe the length, curvature and bending forces. The optimal condition of these forces is to preserve the distance between points equally.

Elastic energy: it makes a line that connects two points, P_i and P_{i+1}, on the balloon body behave like a piece of rubber. The length of the line represents the weighted elasticity by a factor K_s. A large value of K_s indicates large elasticity is being allowed.

$$E_s = K_s \sum_{i=1}^{N} |P_i - P_{i+1}|^2 \quad (1.2)$$

Bending energy: it represents the relationship between three points respectively by controlling the curve degree at the middle point. Therefore, the line passing these points behaves as a thin plate or a straight cord according to a bending factor K_b. A large value of K_b creates a curve at the mid-point whereas the small value defines a sharp angle or vertex. When K_b is zero, discontinuity in the second order of the bending function occurs and an angle may occur consequently.

$$E_b = K_b \sum_{i=1}^{N} |(P_{i+1} - P_i) - (P_i - P_{i-1})|^2 \quad (1.3)$$

With the absence of the other energies, the internal energy leads the balloon to take a circular shape.

.

1.3.1.2. External energy

As long as ACM has a term to control the point coordinates and the balloon shape, it is intuitively to include a term to explain the texture (pixel values *I*) of the object. This energy describes the relationship between the balloon shape and image pixel values. In other words, we can consider the image as a surface which consists of valleys, peaks and mountains. A snake (balloon points) moves over this surface and fills in some valleys according to weight and gravity forces. The energy consists of many particular terms; line energy E_l, corner energy E_{co}, edge energy E_e….etc.

$$E_l = K_l \sum_{i=1}^{N} I(P_i) \qquad E_d = K_d \sum_{i=1}^{N} |\nabla I(P_i)|^2 \tag{1.4}$$

These terms are incorporated in the implementation according to the user's desire. Each term is weighted by a factor based on a specific purpose. For example, K_l controls the behavior of points with respect to the pixel values that occupied by these points. Positive sign makes points move toward the dark area in the image whereas the negative sign attracts points to the bright area. A small value regardless the sign makes the sensitivity of change of lighting conditions is low. K_d controls the behavior of points with respect to the object's edges. A large positive value attracts points to the sharp edges in the image whereas the large negative value makes points avoid the sharp edges. A small value regardless the sign leads to low sensitivity of edge recognition.

1.3.1.3. Energy of constrains

This term is assigned by the user to control the motion constrains such as the maximum and minimum allowed distances between the balloon points, the allowed amount of bending degree at the vertexes,…etc.

The balloon tries to minimize the energy terms by moving the points internally towards the

object edges in order to fit the exact object shape.

1.3.2. Deformable Template

Deformable Template (DT) has been proposed originally to detect the eye contour [9]. It describes the eye contour by a template as shown in Fig. 1.1(a). The template consists of two parabolas representing two eyelids and a circle representing the size of iris and pupil location. The template possesses many parameters that control the template shape, size, angle, vertical distance between the eyelids…etc. A testing image is decomposed into many images by using various filters. These images are called energy fields and each of which highlights some locations of the eye features. The general energy fields are valley image E_{vi} which highlights the location of pupil, peak image E_{pi} which highlights the location of sclera, edge image E_{di} which indicates the location of eyelids and iris boundary and gray image E_{ii} which gives significant information on lighting conditions. Other fields can be used according to the user's implementation strategy. The total energy function E_{DT} is represented by summing up the fields and given as follows:

$$E_{DT} = E_{ii} + E_{pi} + E_{vi} + E_{di} \tag{1.5}$$

where E_{vi} is the valley field that should be minimized according to Eq. (1.6). It is the sum of pixel values *I* in the iris region $R_i$divided by the iris area A_i.

$$E_{vi} = -\frac{K_{vi}}{A_i} \int_{R_i} I(x,y)\, dA \tag{1.6}$$

E_{di} is the edge field that should be also minimized and its value is the sum of pixel values that represent the eyelids ∂R_wdivided by their length ∂A_w. This term is summed to the weighted sum of pixel values that represent iris boundary ∂R_i divided by its perimeter ∂A_i.

$$E_{di} = -\frac{K_{di1}}{\partial A_i} \int_{\partial R_i} I(x,y)\, ds - \frac{K_{di2}}{\partial A_w} \int_{\partial R_w} I(x,y)\, ds \quad (1.7)$$

E_{pi} is the pick field. Its value is calculated by summing the pixel values of sclera areas. Notice that the presence of the two terms in Eq. (1.8) depends on iris position.

$$E_{pi} = -K_{pi1} \int_{R_{p1}} I(x,y)\, ds - K_{pi2} \int_{R_{p2}} I(x,y)\, ds \quad (1.8)$$

E_{ii} is the gray field and it represent two areas; iris and sclera. The sum of pixel values inside iris should be minimized whereas the sum of pixel values representing the sclera areas/ eye area should be maximized.

$$E_{ii} = -\frac{K_{ii1}}{A_i} \int_{\partial R_i} I(x,y)\, dA + \frac{K_{ii2}}{A_w} \int_{\partial R_w} I(x,y)\, dA \quad (1.9)$$

Each particular term in the above equation is weighted by a factor that governs the energy behavior according to the pixel values in the studied field.

The template parameters are updated iteratively in order to fit the eye contour in a testing image. First, the testing image is presented and initial values are given to the template parameters and energy factors. The valley image is firstly used to search on the iris center and the template is then fitted. In the next stage, a smooth tuning of the center coordinates is done and the iris radius is updated using gray, edge and valley images. The angle of the template is then estimated using the peak image and the template is rotated around the updated iris center. Finally, the size of the template is modified to match the two eyelids and eye corners using gray, edge and peak images. A smooth tuning can be attained for the whole parameters to adjust the template perfectly.

1.3.3. Active Shape Model

Active Shape Model (ASM) has been introduced as an advanced version of active contour models [10]. ASM has been employed in many applications [25-29]. It can be said that ASM is the simplest and most effective algorithm for object shape detection. Originally, ASM has been applied for analyzing medical images such as heart and brain images [30]. An automatic segmentation of some particular parts in the medical images is very necessary and helps doctors to turn directly into the next process in medical check. As can be inferred from ACM explanation in the previous sections, ACM has no prior knowledge on the object's texture and shape. This weakens the behavior in order to deal with complex object body and to process the objects in grayscale level with complicated background. Moreover, tuning of the constants of the energy terms (K_b, K_d, K_s andK_{bl}) is quiet difficult and hence accurate results are not guaranteed. Therefore, the need of implementing more robust algorithm with knowledge on the object's shape and appearance has arisen and ASM has been invented.

ASM strategy consists of two stages; training and detection. In training stage, database is used to create two models on object's shape and texture. The images in the database must be chosen carefully to represent as much different object shapes and lighting conditions as possible. The object shape is described by a set of points. These points are distributed and labeled manually and carefully over each image in the database to represent the eye contour. In order to represent the general eye state exhibited in the database, the averaged shape over the manually labeled shapes is calculated. Once this step is done, a static method is applied to model the variation of the labeled shapes. Consequently, the model can be utilized to explain the geometrical eye state variation as well as to refine detected shape deformations.

An imaging model of the pixel values around the eye shape points is created. The pixel variation at each eye shape point is modeled. Therefore if the eye shape is described by a set of *N* points, *N* models are created and each model represents the general pixel variation around the

corresponding point.

In detection stage, a testing image is presented and the averaged eye shape is fitted. The point coordinates of the shape are updated using the imaging model. The deformations in the updated coordinates are refined using the eye shape model. The refined shape is then fitted again to the testing image. These procedures are repeated until the convergence is achieved and the final (detected) shape is obtained.

We just gave a brief explanation on this technique because it will be detailed in the third chapter and some development strategies will be introduced as well.

1.3.4. Active Appearance Model

Active Appearance Model (AAM) is a sophisticated method that inspired from ASM and introduced for performing higher accuracy [31]. Originally, AAM has been applied for facial feature extraction and it is rarely used for other applications. The technique creates also two models; shape (same as ASM) and appearance. Unlike the ASM strategy in representing the pixel values around the object shape points, AAM constructs an appearance model to describe the change of full pixel values of the object in an image. The key idea is to sample the area that occupied by the object into triangles. Each triangle is built using three object shape points and these points could be also shared by the other triangles. The number of triangles is optimized using Voronoi diagrams [32].

The two models share a common parameter to describe the correlation between the change of object appearance and shape. Instead of searching on the object shape points locally as ASM, AAM tries to minimize iteratively the difference between the testing image and an image that is synthesized by the appearance model.

Due to the complicated structure of AAM, minimization process is a very difficult

optimization problem. In addition, the implementation of the algorithm is complicated and takes time as well as AAM allocates a very large amount of static and dynamic memories. This makes the use of AAM is not so preferable and the performance very slow with high risk of bugging occurrence. Therefore, AAM is currently not an appropriate technique for real time applications.

In this thesis, we exclude this technique to detect the eye shape and we believe that working on developing and enhancing the ASM performance is more promising and impressive.

1.4. Problem statements and objective

Static system: the system that detection time or time consumption is not so important issue such as constructing a panoramic image from different photos. Thus, the quality of results and the high accuracy achievement are most important.

Real-time/on-line system: the system that the detection time is the most important issue because the results must be obtained as fast as possible for running the process through the time. Accuracy is also an important parameter and a tradeoff with detection time must be done and

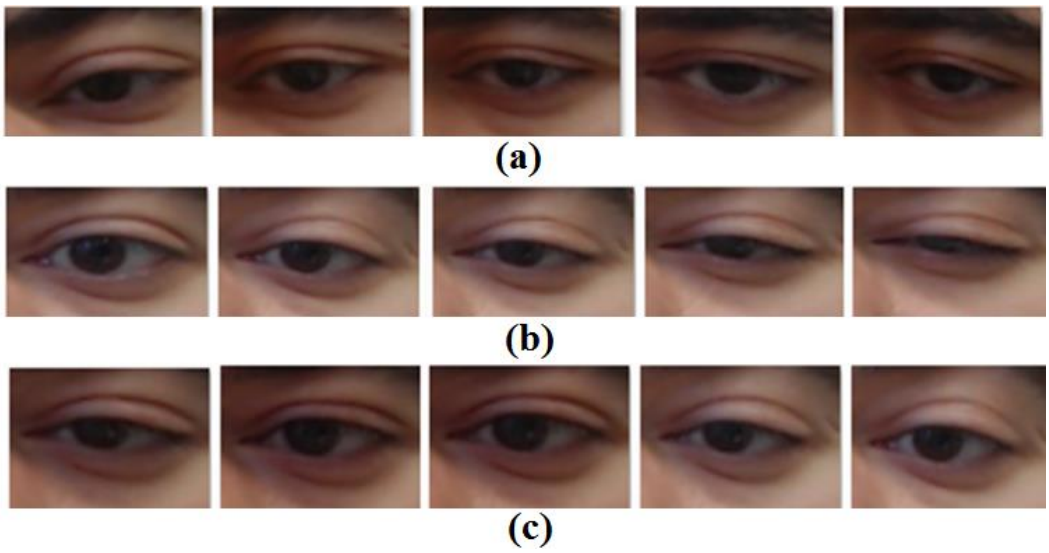


Fig. 1.2 Eye appearance representation. (a) Different rotation angles. (b) Different eye states. (c) Different lighting conditions.

adjusted according to the applications.

There are many parameters and factors that must be taken into account for implementing real time techniques based eye shape detection. Herein, we highlight some general problems:

- From a practical/technical point of view, eye appears in a great degree of verity in shape. Scale and rotation describe the eye profile condition and called the eye shape parameters. The change of rotation can be described as the appearance of eye at different angles as demonstrated in Fig. 1.2(a). The change of scale is described as the change of the distance from the camera or as the change of the eye state; open, nearly closed and in-between as shown in Fig. 1.2(b). These two parameters influence each other and cannot be studied independently. For example, at a certain/fixed scale of the eye, the proposed method must process all available rotation angles of the eye appearance ($\mp 45°$) and vice-versa. The robustness of modeling *pixel patterns* around the eye shape points is the key point to well address this problem.
- From a detection point of view, changing of pixel values influences the eye texture and appearance. This is related to two parameters; lighting conditions and skin color as illustrated in Fig. 1.3(c). A robust *pixel value model* is needed to describe how pixel values change around the eye contour.

Based on addressing the above two problems, a strategy of searching on the eye shape points is chosen such as using an iterative searching technique in ASM and DT.

Iterative searching strategy: This strategy is widely used in image processing for shape matching and detection [33], object detection [34], power iteration [35], affine transformations [36]…etc. For example, ASM tries to minimize the error between the fitted eye shape and the exact eye shape in a testing image iteratively. Therefore, a bunch of steps will be repeated until convergence is achieved and the next testing image is then processed.

There are two common options to describe the convergence condition that terminates an iterative process. In the first option, convergence is achieved when the difference in accuracy between the last two detected eye shapes that obtained by the last two iterations is less than a threshold. The second one is to let the process run for a certain number of iterations regardless the accuracy at the final iteration. In the both options, the eye shape of the last iteration is considered as the detected eye shape and supposed to fit and match the eye contour in the testing image.

In this thesis, an iterative technique, (ASM), has been developed. On the other hand, we will end up with the fact that iterative strategies are not appropriate/trustful for real-time applications. If the convergence is done according to a threshold, this means each testing image is considered as an independent case with consuming different number of iterations depending on the eye structure and state. Thus detection time is not known for each case. Moreover, such technique may stuck and run in a loop without reaching the threshold. On the other hand, if the convergence is attained by a fixed number of iterations, achieving high accuracy is not guaranteed and many deformations in the detected eye shape may occur.

In conclusion, the proposed methods must deal with change of lighting condition and skin color, decrease the number of iterations as possibly or avoiding the use of an iterative searching strategy, have flexibility to process wide range of scales and rotation angles, achieve high accuracy and reduce detection time significantly to be applicable in real time.

Chapter 2

2. Human Visual System

2.1. Eye structure

Eye is considered as the most complex component in the human body. The complexity can be deduced by answering on the common question "*how human sees?*". To get the correct answer and understand the meaning of this question, human visual system should be interpreted and explained from both biological and mathematical points of view. The visual system is a combination of many mathematical transforms. In this chapter, we take into account this truth and briefly explain the human visual system.

2.1.1. Biological structure of human eye

Eyes are connected to the brain by the central optic nerve [37]. The connection point is called

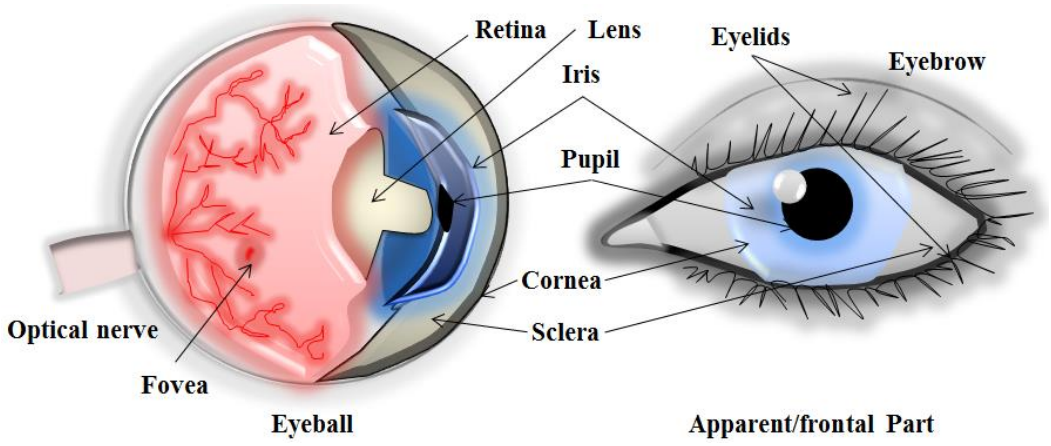


Fig. 2.1 Eye structure.

the blind spot. The nerve is a collector of signals from sub nerves that exist in the eyeball internally as shown in Fig. 2.1. The eye is shaped as a ball and the apparent part has a convex shape covered by *cornea*. *Iris* is the colored part located behind the thin layer of *cornea*. The *iris* controls the *pupil* size to regulate the light flow. The *pupil* is located in the middle of the *iris*. A lens exists behind the *iris* and it has flexibility to change its shape. The lens shape can be stretched and compressed as well as its thickness is changed according to the human's focusing

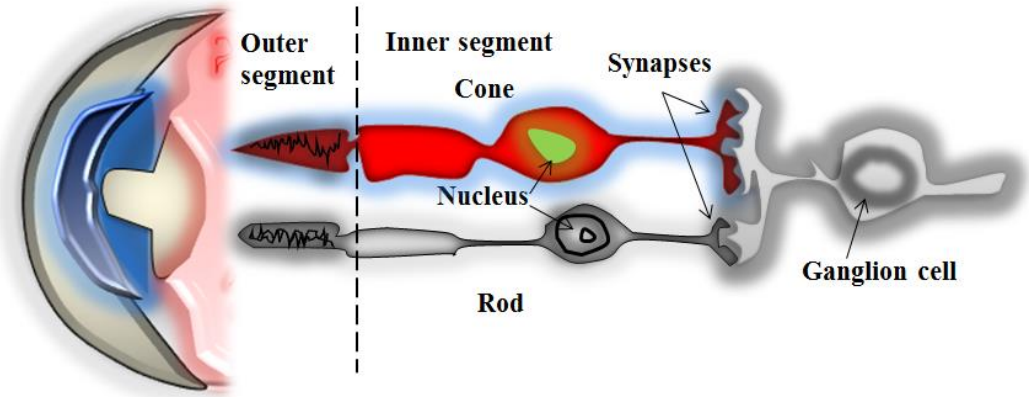


Fig. 2.2 Eye photoreceptors.

attention. If the object is far away from the eye, the lens shape changes to be thin and to be thick if the object is close.

The light is reflected from the objects and goes through the *iris*. The light is then focused and projected through the lens into the *retina*. The *retina* consists of particular elements (photoreceptors) that sense the light. Photoreceptors are around 125 million connected to around 100 million optic nerves. The reason behind non-matching between the two numbers is that each group of photoreceptors is locally assembled and combined into one nerve.

The photoreceptors are classified into two groups; rods and cones as depicted in Fig. 2.2. Rods are very sensitive to light and brightness whereas cones have different types of wavelength color sensation. Cones and rods are arranged in parallel beside each other. Each photoreceptor produces an impulse when it senses light. The impulses are converted into a series and

transmitted to the brain through the optic nerve. The brain assembles the series obtained from different optic nerves and converts them into an image. The image is then understood by the brain by generating different level of image resolution, recognizing lines and edges, creating surfaces, attaching color to the surfaces, adding depth perception and producing movement map [38][39]. The feeling is added to this complicated strategy by discovering, memorizing and understanding what is going on in the environment at the moment.

Colors provide more details to the human perception. There are three types of cones to recognize the basic colors; S cones for blue color sensation which has wavelength of 437nm, M cones for green color sensation at wavelength 533nm and L for red color at 564 nm wavelength as graphed in Fig. 2.3. The sharpest vision occurs at 2 degrees from the center of the visual axis in *fovea* which contains only cones.

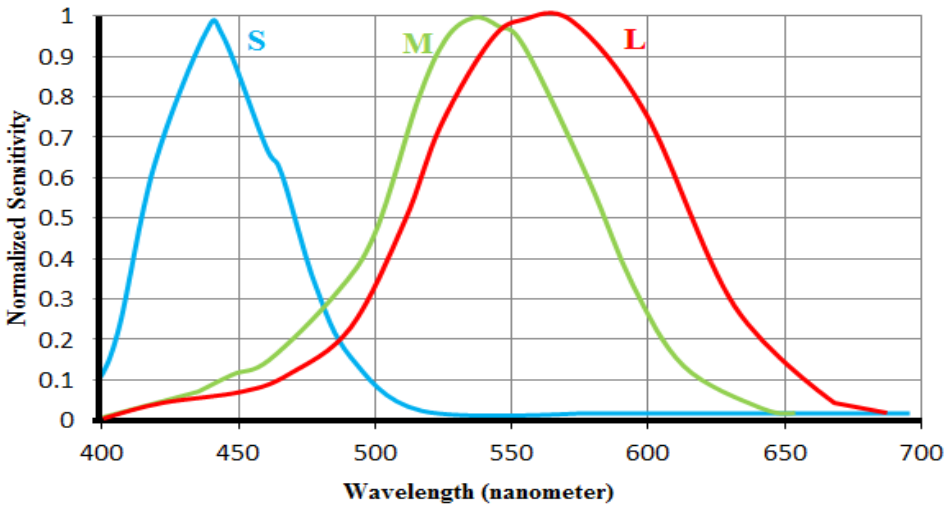


Fig. 2.3 Color response of S, M and L cones.

The cones and rods response is nonlinear and can be approximated by a logarithmic function of energy. There is a very known and common mistake in understanding how brain recognizes and processes colors. The general understanding is that the cones give red, blue and green signals to the brain. The visual system is more complicated and right interpretation is that the generated signals by S, L and M cones are not transmitted directly to the brain. They are

converted into three components; achromatic which is the summation of three signals, red-green signal and blue-yellow signal. This low level process of color signals is done by a complicated network of cells called *Ganglions*.

A *ganglion* cell collects the output signals from few cones according to their positions in the *retina*. For example, in the *fovea* area, where the most details are concentrated, a few cones

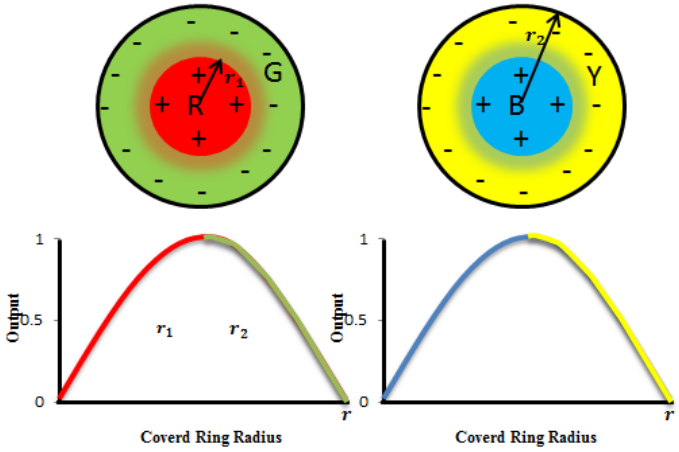


Fig. 2.4 On-off center-surround respective field.

are connected to one cell. The number of the combined cones in the *retina* increases radially as the distance increases far from *fovea*. A *ganglion* cell has the style of on-off center-surround response which is called the respective field as illustrated in Fig. 2.4. Each receptive field can be represented as a central disc surrounded by a ring. When the light covers a part of the central disc, the corresponding *ganglion* output increases. A full fill of the central disc means that the *ganglion* output is at maximum. When the central disc and a circular part of the ring region are covered by light, the output is reduced. A fully covered of the receptive field leads to zero output. For this reason, we mentioned that cones has a non-linearly response. As we get far from the fovea, the receptive field region becomes wider.

The central disk may sense red color while the ring region senses green color. According to this strategy, colour-opponent receptive field is built. The receptive fields are arranged in rows

(a) **(b)**

Fig. 2.5 Log-polar transform done by human visual system.(a) Raw image. (b) Foveated image.

and this distribution is used to recognize edges and lines.

2.1.2. Mathematical human eye structure

From a mathematical point of view, some mathematical functions used by eyes can be deduced from the previous explanation. In this section, we try to highlight simply these functions.

2.1.2.1. Foviated system

The eye focuses on some particular points in the scene. This is because of the receptive field distribution and the high visual details concentrated in *fovea*. This truth can be proven by imaging that a student asks a question to a teacher. The student's eyes may look at some particular features on the teacher's face. Suppose that the teacher starts to answer the question and lays his left hand horizontally. The student recognizes this action while he keeps focusing on the teacher's face. The teacher starts to move randomly his left hand fingers. In this case, the student may only recognize the motion but he is not able to determine which finger moves at the moment because the visual details are proportionally distorted as the distance increases from the gaze point.

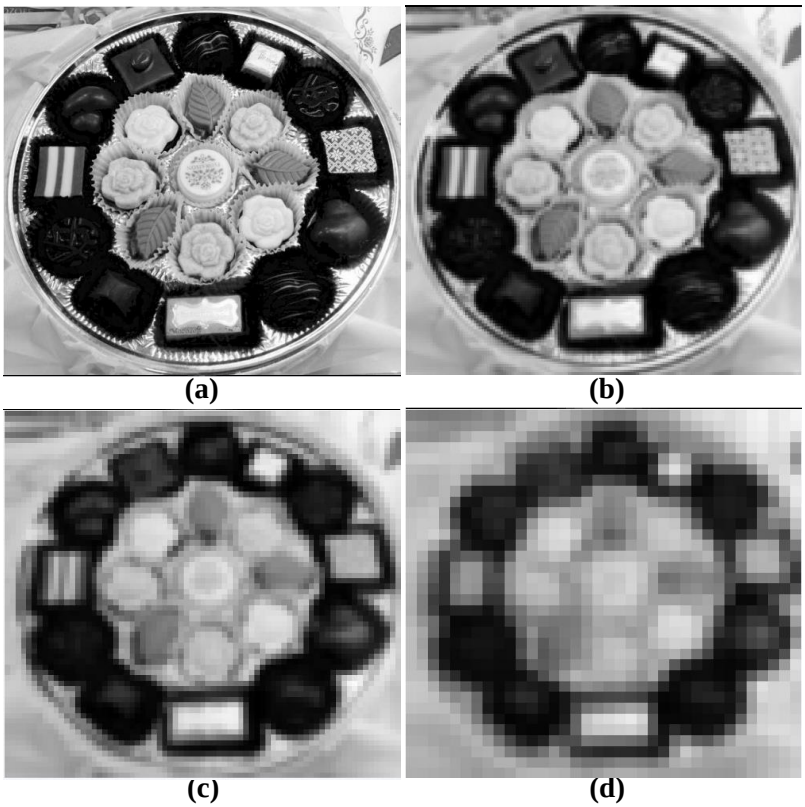
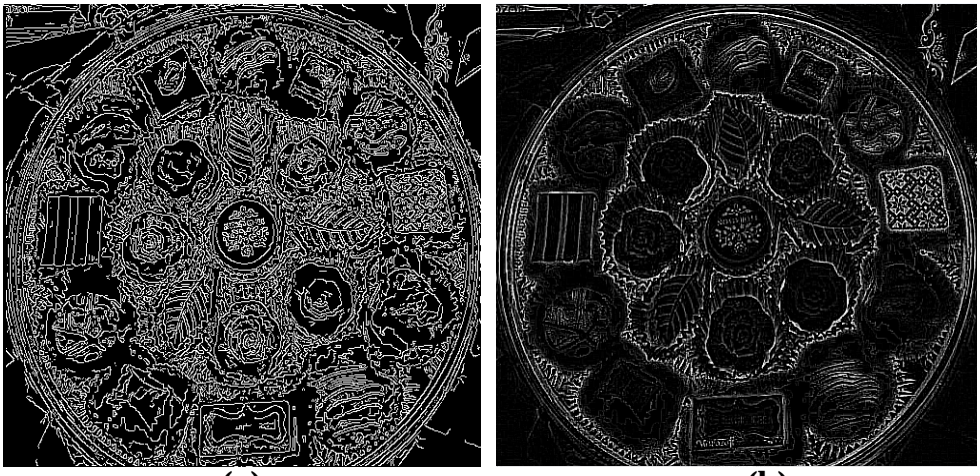


Fig. 2.6 Four gray level pyramid. (a) Top image. (d) Base image.

This processing strategy can be mathematically interpreted as the eye uses Logarithmic-polar transform [40]. The mathematical equation and more details on this transform will be provided in chapter 4. Figure 2.5 demonstrates the process by showing in part (a) a chocolate box. Suppose that a person looks at the center of the image (gaze point is at the image center). The corresponding created image in the brain resembles the image in the part (b) that generated using log-polar transform. It can be noticed that the details are proportionally distorted far from the center.

2.2. Image construction

The first stage in human visual system is to build a gray level pyramid of the scene as shown in Fig. 2.6. The base of the pyramid represents less and poor details whereas the top represents the



(a) (b)

Fig. 2.7 Object surface construction. (a) Canny filter. (b) Laplace filter.

high resolution image with high details. The reason of creating a pyramid is to understand and differentiate the objects in the image easily, processing faster and reducing energy consumption by the brain. For example, the less detail image is used to differentiate the masses of the big objects and to measure the distance between them. Going up for the next level, the edges of the big objects may be detected and small objects' masses can be observed. This operation runs until the high resolution image is reached up and all objects are being detected and recognized in their textures, sizes, depths, shapes…etc. The operation can be mathematically interpreted as the eye uses Gaussian filter to create the pyramid [41]. Figure 2.6 illustrates a pyramid of four levels that created on the image in Fig. 2.5 using Gaussian function.

2.3. Object construction

The second stage is to detect lines and edges in the scene. This detection operation is applied on the pyramid images. The process can be simply represented as the eye scans the scene using some filters such as Canny [42], Hough [43], Laplace [44]…etc. Figure 2.7 demonstrates this process by showing the results of applying Canny and Laplace filters on the images in Fig. 2.6(a). Different features are highlighted and detected in each result as well as there are some

common ones. Due to complicated structure of neural networks, human brain is capable to apply various filters on the signals that are delivered by the central optic nerve.

2.4. Color detection

Based on the detected lines and edges, the brain tries to generate a deeper perception on the scene by constructing objects' surfaces. Colors are then given to these surfaces. The colors are recognized by cones by generating three channels; chromatic, red-green and blue-yellow. This transform is similar to HSV color system in image processing where H is hue channel, S is saturation channel and V is value channel. The human brain transforms the three components internally into RGB color space. From a mathematical point of view, the visual system uses an equation to transform the image/scene form HSV color space to RGB. In the next two sections, a brief explanation on these two color spaces is provided.

2.4.1. RGB Color Space

RGB color space consists of three color channels, red, green and blue. Each channel consists of pixels that vary from 0 pixel value (intensity value) representing black color to 255 representing white color. In the color wavelength spectrum, red is 645.61 nm, green is 526.2 and blue is 444.4. The correlation between these channels is very high. This means if the intensity changes, all three channels will change accordingly.

Mixing the three components RGB with various intensity proportions produces hundreds of different colors. The color space can be modeled using a cube in 3D Cartesian coordinate system as illustrated in Fig 2.8. The axes of the coordinate system represent the three main colors RGB whereas the non-contact vertexes represent the colors obtained by the intersection areas between the main three colors.

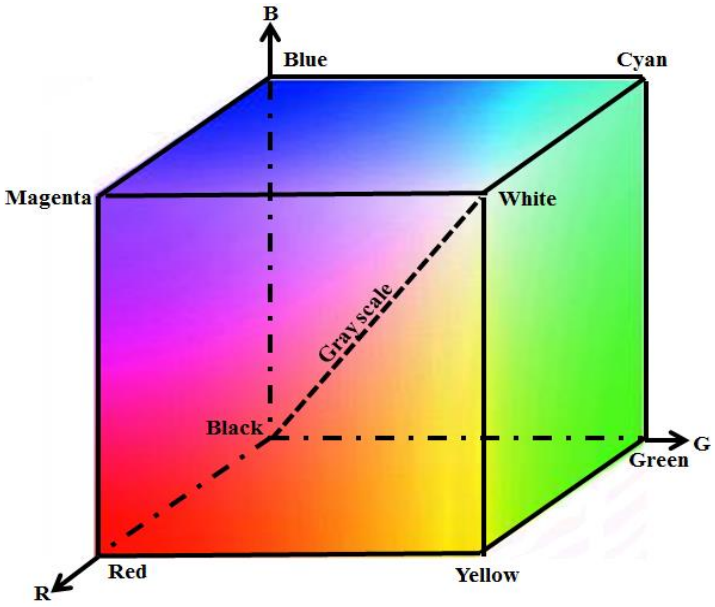


Fig. 2.8 RGB color system.

The diagonal axis of the cube that originates from the origin represents gray color. Images can be represented in grayscale in order to provide less information in each pixel. Therefore, the pixel value changes between 0 and 255.

2.4.2. HSV Color Space

HSV color space is represented by a cone and there are three parameters to determine a particular color as shown in Fig 2.9. Vertical position determines brightness, radial position determines the color saturation and the angular position determines the hue. Brightness varies from 0 which represents darkness to 1 representing the full brightness. Saturation varies from 0 corresponding to shed of grey until the fully saturated color. Hue ranges from 0 degree which represents red color through yellow, green, cyan, blue, magenta and back to red at 360 degree. Dealing with this color domain is closer to the human perception of colors. For example, we like to change the color of an object in the image with preserving its other features such as brightness and shadows. This task is very hard to be achieved in RGB color space because of high correlation between the color channels. On the other hand, it is very easy in HSV space

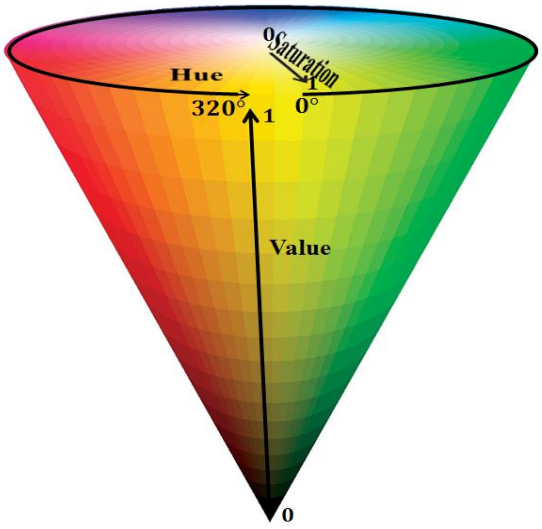


Fig. 2.9 HSV color system.

because the object color has a specific hue value. This value can be treated independently regardless the values of intensity and saturation. In Fig. 2.10(b), the color of a chocolate piece has been changed from brown, as in Fig. 2.10(a), to dark blue with preserving shadows and lighting characteristics.

The process mentioned above describes an extreme small degree of complexity of the human visual system because following this strategy the brain can process and recognize the objects under different colors and lighting conditions.

(a) **(b)**

Fig. 2.10 Object color change in HSV color space. (a) Raw image(all chairs in red color). (b) Modified image (one chair in yellow).

2.5. Object recognition

After detecting colors, the brain tries to recognize the objects and the environment in the scene. The concept of recognition is vast and there are no rules to describe exactly how the brain identifies objects. The key tool is using neural networks to understand, analyze, memorize, recognize, perceive, store, and process the environment. Each term of these concepts is a scientific research domain. There is no certain number of objects that can be recognized by the human brain but some studies refer to around one million. Neural network is a very active topic in computer vision. In chapter 5, we try to explain as simple as possible the principles of neural network and relevant learning techniques.

2.6. 3D perception

The last step in the visual system is to provide the depth information to each particular element the scene. The depth map is calculated using two eyes [45]. Simply, each eye delivers to the brain an image of the scene/view. Brain tries to detect correspondence and similarities between the two images.

Suppose that the right image is the reference. Brain starts to scan the image pixel by pixel and row by row. During the scanning process, brain tries to find the same/corresponding pixel in the left image in order to measure the difference in positions. The difference in positions is then used to calculate the location (depth) in the real world. Consequently, depth information is incorporated to provide an ultimate feeling and sensation of the scene.

2.7. Eye motion

Three pairs of muscles control the eye movements, Left/right, up/down and rotate it around the

visual axis [46]. The field of vision is determined by the eye position and rotation angle which is the same angle of human face. Each eye covers around 120 degree of field vision view whereas both of them guarantee 160 to 210 view degree.

Many laws have been proposed to model and address the eye motion [47][48]. From a technical point of view, eye moves with five degree of freedom. This means that the gaze point changes rapidly as well as the field of view. Face rotation angle also affects the direction of the field of view and should be taken into account in the gaze point calculation. Therefore, pupil location must be detected accurately as well as some reference points on the eye contour (eye corners and two middle eyelid points). Thus, we consider eye shape as the key and general point for implementing an interaction system-based eye movement. Moreover, it paves the way for adding numerous extensions and incorporating many tasks with the implemented system.

Finally, there are many techniques and concepts that not mentioned in this chapter and we want to emphasize that the discussed operations are simply explained to be compatible with the thesis's context from the author's point of view.

Chapter 3

3. Eye Shape Detection Using Active Shape Model

3.1. Introduction

ASM is a static method used to solve many problems in image processing [49-54]. Unlike the other methods, ASM strategy takes into account the need of prior knowledge on object's shape and texture. ASM simulates a very simple and logical mechanism to detect shapes. Suppose that the aim is to detect the eye contour which is described by N points in a testing image. It is intuitive to start by a general shape of the eye and try to match its points to the eye contour. As long as the general shape is fitted randomly without considering the actual eye state in the testing image, each eye shape point must be attracted independently to the closest location of the eye contour. Hence, this point becomes an exact eye shape point.

An issue that can be highlighted is *how to attract the points to the eye contour*? Obviously, a searching technique based prior knowledge on pixel values at the eye contour should be used. Therefore, ASM models the change of pixel values at the eye contour. After applying the searching operation on the common eye shape points, their locations are modified. As long as the searching operation is done independently, many deformations may occur and appear in the eye shape such as two modified points are being very close or one point is at wrong location while the others are at right ones. Therefore, an eye shape model is needed to refine the

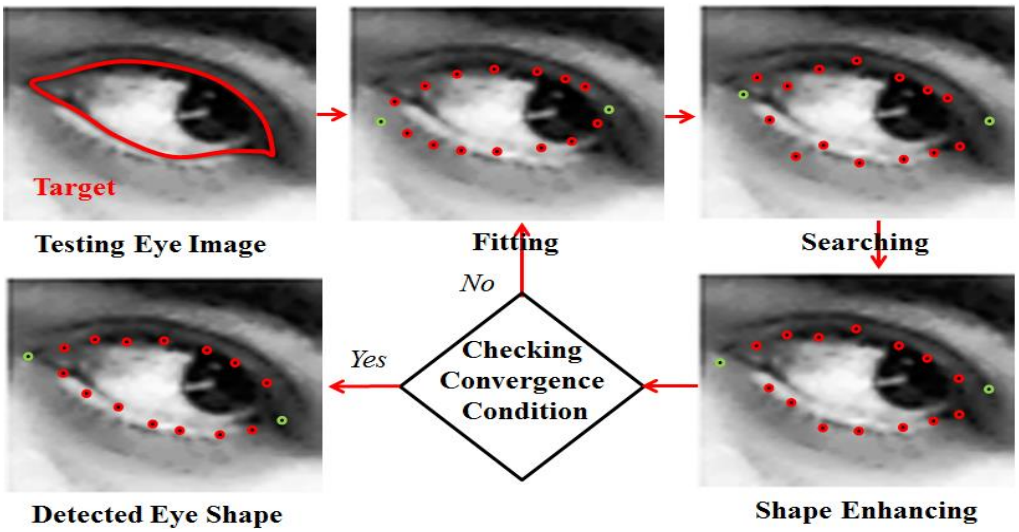


Fig. 3.1 Active Shape Model structure.

deformations with respect to all modified locations of eye shape points.

The second issue arises after refining the deformations is that *does the refined eye shape match the actual eye contour perfectly*? Intuitively, no!, because the eye model just refines the partial geometrical deformations regardless the point positions in the image domain. Therefore, it is not guaranteed that the searching operation will attract the shape points to the eye contour perfectly. On the other hand, the modified eye shape is expected to approach the eye contour. This eye shape can be utilized to search again and getting closer and closer to the eye contour. Thus, ASM uses an iterative searching technique to search on the eye contour until satisfying the convergence condition and getting the final detected eye shape.

Figure 3.1 illustrates the structure of ASM. First, a testing eye image is presented and an initial eye shape is suggested to be closer in the eye contour characteristics. This procedure is called fitting operation. A searching operation based pixel value model is then applied to modify the locations of the eye shape points. The deformations of the modified eye shape is then refined using the eye shape model. Finally, the convergence condition is checked whether to run the process again for the next iteration using the refined eye shape or to terminate the process and consider the refined shape as the detected eye shape in the testing image. In the next sections,

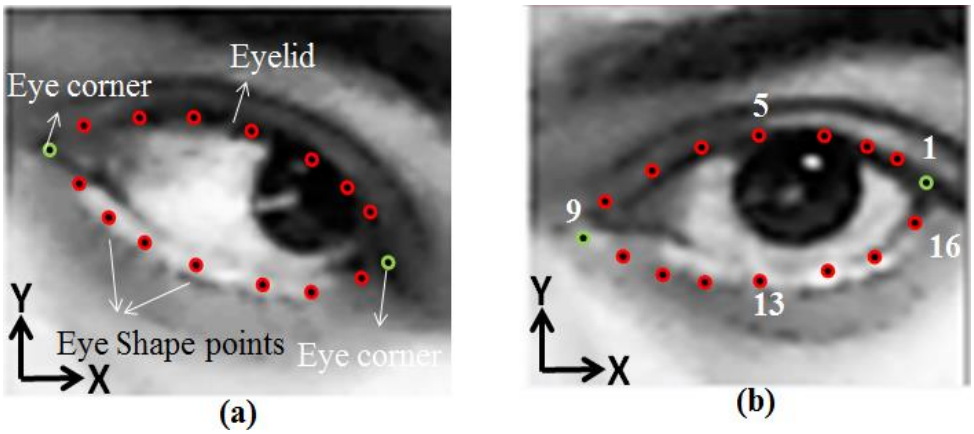


Fig. 3.2 Two eye images with different appearance.

each block is detailed theoretically and mathematically.

3.2. Eye shape modeling strategy

ASM uses a prior knowledge on the eye's shape and texture. An eye image database is used to create these two models. In order to model the eye shape, the eye image database must contain as various eye profiles as possible. On the other hand, the eye shape model must have capability of generalization. Generalization means that the model can recognize new eye shapes that have been not presented in the charged database.

3.2.1. Eye shape description

The eye shape is described by a set of N points and each point has (x,y) coordinates. Each eye shape point must be approximately located at a specific area on the eye contour. For example, we use N=16 points to describe the eye shape; 1^{st} and 9^{th} points represent two eye corners, 5^{th} and 13^{th} represent the middle points at upper and lower eyelids respectively. The other points are distributed equally between these remarkable points as clarified in Fig. 3.2(a). According to this distribution strategy, eye shapes of eye images in database are manually labeled. Thus, the programmer should click on the points over each image with taking into account two issues; the

points should be located at the sharpest edges of the eyelids as well as the local correspondence of point locations among the eye images must be highly preserved. Figure 3.2 shows two eye images and corresponding eye shapes that are manually labelled. The correspondence between points can be observed as the first point represents the right eye corner regardless the eye angle and state. The eye shape points are labeled in Cartesian coordinate system and their (*x,y*) coordinates are arranged in a vector as in Eq. (3.1). This mathematical representation is needed for the next steps of modeling the eye shape.

$$X_i = \left(x_{i1},\ x_{i2}, \dots x_{ij}, .., x_{iN}, y_{i1}, y_{i2}, \dots , y_{ij}, \dots , y_{iN}\right)^T \quad (3.1)$$

where X_i represents the exact eye shape vector of i^{th} eye image in database, $x_{ij}\ and\ y_{ij}$ are the j^{th}point coordinates in *x*-*y* plane and *T* refers to the vector transpose operation.

By the end of this step and if the database contains *M* eye images, *M* eye shape vectors are obtained and each of which consists of 2x16 elements (*x*,*y* coordinates).

3.2.2. Eye shape alignment

The aim of the eye shape model is to represent as different eye states as possible. Therefore, the Euclidian similarity between the eye shape vectors should be removed. In order to achieve this process, an alignment operation must be applied on the eye shapes. In the next section, we use Procrustes analysis method to align two eye shapes then this procedure is applied on the *M* eye shape vectors [55].

3.2.2.1. Two eye shapes alignment

Figures 3.3(a-c) show two eye shapes that have different shape parameters; scale, rotation and translation. Mathematically, the two eye shapes are represented by two vectors as follows:

$$X_1 = \left(x_{11},\ x_{12}, \dots x_{1j}, .., x_{1N}, y_{11}, y_{12}, \dots , y_{1j}, \dots , y_{1N}\right)^T \quad (3.2)$$

$$X_2 = \left(x_{21},\ x_{22}, \dots x_{2j}, .., x_{2N}, y_{21}, y_{22}, \dots , y_{2j}, \dots , y_{2N}\right)^T \quad (3.3)$$

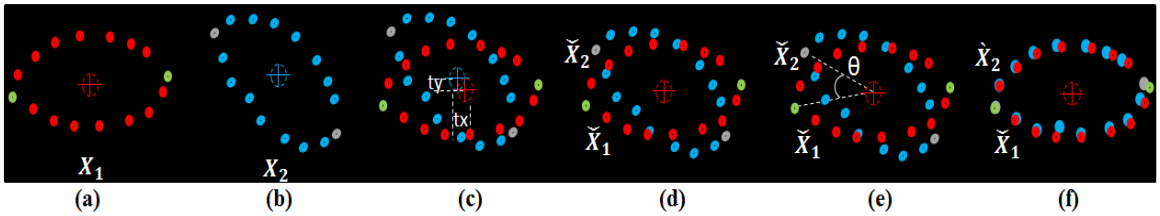


Fig. 3.3 Two eye shapes alignment.

The eye shape X_2 can be aligned to X_1 (reference shape) by estimating translation parameters T(t_x, t_y) which measure how much the center of X_2 is shifted from that of X_1, the scale factor *S* which measures how much X_2 is bigger/smallar than X_1 and the rotation angle θ which best rotates X_2 to the reference shape.

First, the center coordinates $\bar{x}_i$, $\bar{y}_i$ of each eye shape is computed by calculating the average *x*,*y* coordinates over the eye shape points as follows:

$$\bar{x}_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij} \qquad \bar{y}_i = \frac{1}{N}\sum_{j=1}^{N} y_{ij}\,; \quad i = 1,2 \tag{3.4}$$

In order to make the shapes zero mean $\tilde{X}_1, \tilde{X}_2$, the center is subtracted from the shape points using Eq. (3.5). Figure 3.3(d) illustrates this step and it can be observed that the centers of the two eye shapes are exactly matched.

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_i \qquad \tilde{y}_{ij} = y_{ij} - \bar{y}_i \tag{3.5}$$

The shapes are then normalized to make the scale equals to 1using Eq. (3.6).

$$\check{X}_1 = \frac{\tilde{X}_1}{|\tilde{X}_1|} \qquad \check{X}_2 = \frac{\tilde{X}_2}{|\tilde{X}_2|} \tag{3.6}$$

Yet, the shapes have the same scale and center coordinates. Rotating the second eye shape is the last action to be taken for completing the alignment procedure as illustrated in Fig. 3.3(e). The rotation matrix is given in Cartesian coordinate system by Eq. (3.7).

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \tag{3.7}$$

In order to calculate the best alignment angleθ, Singular Value Decomposition (SVD) is used [55]. The 2x2 covariance matrix of these two shapes is calculated as follows:

$$C = \check{X}_1 \check{X}_2^T \tag{3.8}$$

The rotation angle θ is obtained by maximizing the trace of *RC* matrix. This means that the correlation between the two shapes is maximized. The trace is calculated by summation of the diagonal elements of *RC* matrix. The covariance matrix is broken down into three matrices USV^T where *U* and *V* are two orthogonal matrices and *S* is a diagonal matrix. The term of $E = RUV^T$ is also an orthogonal matrix. Therefore, maximizing the trace of *RC*=*SE* can be achieved by enforcing the diagonal elements of *E* matrix to be 1. Hence, $R = VU^T$ and the second eye shape $\check{X}_2$ can be aligned to the first shape $\check{X}_1$ using Eq. (3.9) as demonstrated in Fig. 3.3(f).

$$\grave{X}_2 = R\check{X}_2 \tag{3.9}$$

3.2.2.2. *M* Eye shapes alignment

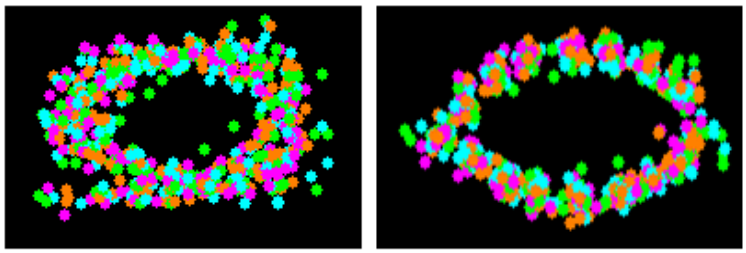
Aligning a set of eye shapes is similar to aligning two shapes with some additional steps. Figure 3.4(a) shows the normalized *M* eye shapes that have zero mean and unity scale factor. Figure 3.4(b) shows the corresponding aligned shapes after removing the differences in rotation. The alignment algorithm can be summarized as follows:

1) Initial stage.
    a) Consider the first eye shape as a reference shape;
    b) Align the remaining *M*-1 eye shape to the reference shape.

   The initially aligned *M* eye shapes are the input data to the next stage.

2) Iterative stage.
    a) Calculate the mean shape $\bar{X}$ of *M* eye shapes;

$$\bar{X} = \frac{1}{M}\sum_{i=1}^{M} \grave{X}_i \tag{3.10}$$



(a) (b)

Fig. 3.4 Eye shapes alignment.

b) Align the *M* shapes to the mean shape $\overline{X}$;

c) Repeat steps (a) and (b) until convergence. The convergence is achieved when the difference between the previous and current calculated mean eye shapes is smaller than a threshold.

By the end of this step, we obtain *M* aligned eye shapes as well as the mean eye shape $\bar{X}$which represents the general and common eye state in the database.

3.2.3. Eye shape representation in the model space

The *M* aligned eye shapes provide significant information on how the eye shape points behave among database. For simplicity, Fig. 3.5 highlights this issue particularly for each eye shape point using red circles. Each eye shape is a vector consists of *2N* =2*16 elements that represent the *x*,*y* points' coordinates. Figure 3.6(a) shows an eye shape in *x*, *y* Cartesian coordinate system using *2N* coordinates of *N* eye shape points.

On the other hand, each eye shape can be represented as a point in *2N* dimensions space as demonstrated in Fig. 3.6(b). Figure 3.6(c) shows the *M* aligned eye shapes represented in x_j and y_j plane. As long as we deal with only eye shapes, the *M* points are gathered in a very narrow/tight cluster namely, eye shape cluster. The factor which governs the distribution of these points is only the change of the eye state. Furthermore, the cluster region can be considered as the allowable area to generate new shapes that not exist in the set of *M* aligned

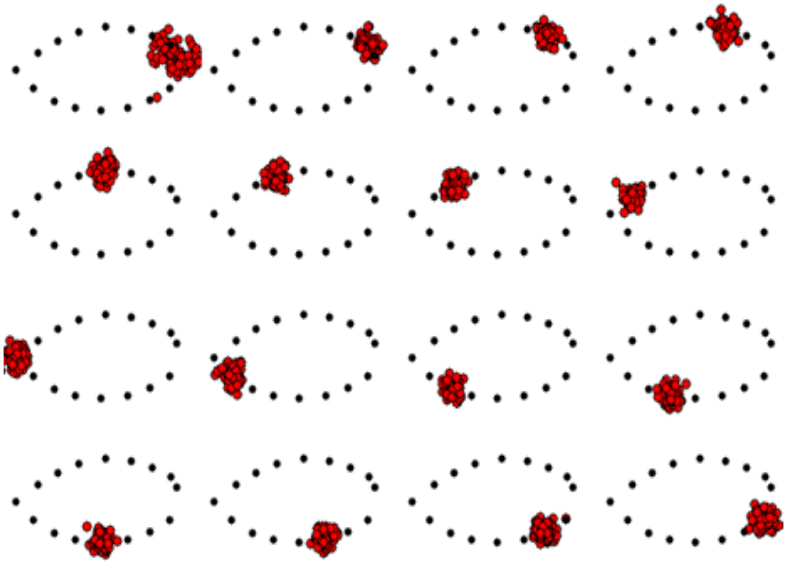


Fig. 3.5 Particular distributions of eye shape points.

eye shapes.

Hence if a point is pinpointed within the region, the corresponding eye shape can be obtained by figuring out the point's projections on the *2N* axes. In order to estimate the cluster

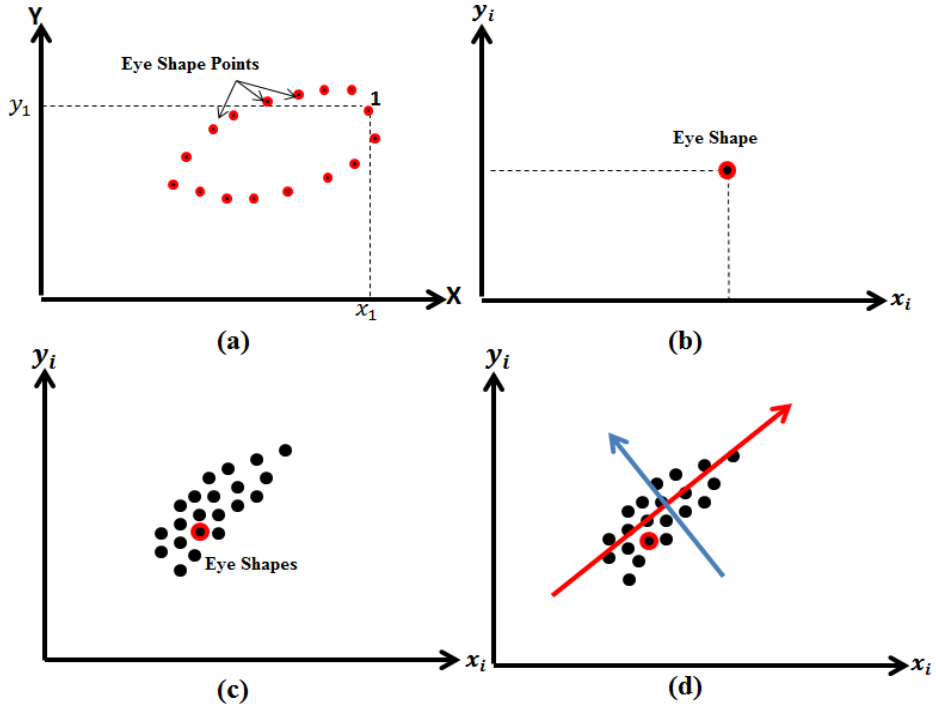


Fig. 3.6 Eye shape distribution. (a) Eye shape points in *x,y* Cartesian coordinate system. (b) An eye shape represented in point coordinate system. (c) Eye shape cluster. (d) Fitting eye shape cluster by a new coordinate system.

boundary and analyze the eye shape variation, the eye shape distributions in Fig. 3.6(c) can be better described by a new coordinate system that best maximizes the representation of variation along its axes as shown in Fig. 3.6(d).

In the next section, we will explain how to obtain this new coordinate system by using Principal Component Analysis (PCA) and the relevant advantages.

3.2.4. Principal Component Analysis

PCA is a statistical technique that describes the eye shape variation by creating a new coordinate system [56-58]. The bases of this coordinate system are orthogonal and called eigenvectors/principal components. Each eigenvector has an eigenvalue which describes the represented variance along the eigenvector direction. The eigenvector which possesses the largest eigenvalue is called the first eigenvector and indicates the direction of the most variation in database. The least variation is described by the second eigenvector, which has the second largest eigenvalue, and so forth.

The PCA strategy can be listed by the following steps:

1) Subtract the mean eye shape $\bar{X}$ (obtained by the alignment operation) from each aligned eye shape.

$$\grave{X}_i = \acute{X}_i - \bar{X} = \{(\acute{x}_{i1} - \bar{x}_1), \dots, (\acute{x}_{ij} - \bar{x}_j) \dots (\acute{x}_{iN} - \bar{x}_N), (\acute{y}_{i1} - \bar{y}_1), (\acute{y}_{i2} - \bar{y}_2) \dots (\acute{y}_{iN} - \bar{y}_N)\} \quad (3.11)$$

Where $\acute{X}_i$represents the i^{th}aligned eye shape, $\bar{X}$ is the mean eye shape and $\grave{X}_i$ represents the distance vector (deviation) between $\acute{X}_i$ and the mean eye shape. After this step is applied for *M* shapes, the mean eye shape is located at the origin and the *M* eye shapes are distributed around it.

2) Mathematically, the distance between $\acute{X}_i$ and the mean eye shape is expressed by Eq. 3.12.

$$\boldsymbol{dis} = \sqrt{(\acute{x}_{i1} - \bar{x}_1)^2 + (\acute{x}_{i2} - \bar{x}_2)^2, \ldots\ldots + (\acute{x}_{iN} - \bar{x}_N)^2 + (\acute{y}_{i1} - \bar{y}_1)^2 + (\acute{y}_{i2} - \bar{y}_2)^2 + \cdots \ldots (\acute{y}_{iN} - \bar{y}_N)^2}$$

Or

$$dis^2 = \acute{X}_i^T \bar{X} \tag{3.12}$$

We aim to analysis and model the variations of *M* points in *2N* dimensional space. Therefore, deviation notion cannot be used because it measures the variance in one dimension. Based on this fact, covariance matrix is used as an equivalent notion in *2N* dimensional space. Thus, a matrix Φ whose columns contain $\grave{X}_1, \grave{X}_2, \ldots\ldots, \grave{X}_M$ is constructed as follows:

$$\Phi = \left[\grave{X}_1, \grave{X}_2, \ldots\ldots, \grave{X}_M\right] = \begin{bmatrix} \grave{x}_{11} & \grave{x}_{21} \ldots\ldots\ldots\ldots \grave{x}_{M1} \\ \grave{x}_{12} & \grave{x}_{22} \ldots\ldots\ldots\ldots \grave{x}_{M2} \\ & \vdots & \\ \grave{x}_{1N} & \grave{x}_{2N} \ldots \grave{x}_{ij} . \grave{x}_{MN} \\ \grave{y}_{11} & \grave{y}_{21} \ldots\ldots\ldots \grave{y}_{M1} \\ & \vdots & \\ \grave{y}_{1N} & \grave{y}_{2N} \ldots\ldots\ldots \grave{y}_{MN} \end{bmatrix} =$$

$$\begin{bmatrix} (x_{11} - \bar{x}_1)(x_{21} - \bar{x}_1) \ldots\ldots\ldots\ldots (x_{M1} - \bar{x}_1) \\ (x_{12} - \bar{x}_2)(x_{22} - \bar{x}_2) \ldots\ldots\ldots\ldots (x_{M2} - \bar{x}_2) \\ \vdots \\ (x_{1N} - \bar{x}_N)(x_{2N} - \bar{x}_N) \ldots\ldots\ldots (x_{MN} - \bar{x}_N) \\ (y_{11} - \bar{y}_1)(y_{21} - \bar{y}_1) \ldots\ldots\ldots\ldots (y_{M1} - \bar{y}_1) \\ \vdots \\ (y_{1N} - \bar{y}_N)(y_{2N} - \bar{y}_N) \ldots\ldots\ldots (y_{MN} - \bar{y}_N) \end{bmatrix} \tag{3.13}$$

3) Calculate the covariance matrix *C* of Φ.

$$C = \frac{1}{M} \Phi\ \Phi^T \tag{3.14}$$

Covariance Matrix: it describes the relationship between variables. In other words, it describes how each element changes from the mean point with respects to the others. The sign of elements is very important because it expresses the variable directions. If an element

has a large value with a positive sign, that means the variables which generate this element change in one direction. Therefore, they are strongly correlated and the relationship is linear. On the other hand, the negative sign with a large value indicates that the variables move in opposite directions. Regardless the sign and when the value is small, one variable at least is close to the mean point.

For example,

$$C(2,1) = \frac{1}{M}(\dot{x}_{12}\dot{x}_{11} + \dot{x}_{22}\dot{x}_{21} + \dot{x}_{32}\dot{x}_{31} + \cdots + \dot{x}_{M2}\dot{x}_{M1})$$

where $\acute{x}_{ij}$ is the *x* coordinate of the j^{th} point of i^{th} aligned eye shape. It is very nicely to observe that *C*(2,1) explains the relationship between x_1 and x_2 coordinates or it represents the relationship between the first and second points in *x* direction over the whole aligned eye shapes. Obviously, this element representation can be enriched up as much as the number of eye shapes increases.

4) Covariance matrix 2*N*x2*N* describes the distribution of the *M* eye shapes in the cluster. Therefore, this matrix can be replaced by *2N* eigenvectors and *2N* eigenvalues λ.

$$CA = \lambda A \tag{3.15}$$

where λ is a diagonal matrix and *A* is a matrix contains the eigenvectors in its columns. These eigenvectors are the bases of a new coordinate system that mentioned earlier. Eigenvectors classify the variance of the eye state by fitting the eye shape cluster in *2N* directions. Each eigenvector has an eigenvalue which describes how much the eigenvector contributes in representing the total variance. The eigenvectors are arranged in matrix *A* according to their corresponding eigenvalues in descending order. Therefore, the first eigenvector in this matrix has the largest eigenvalue and expresses the direction of the most variation in the eye shape cluster. The least variation is described by the second eigenvector, which has the second largest eigenvalue, and so forth.

By the end of this step, the eye shapes are being gathered in the eye shape cluster. The point distribution in the cluster is described by covariance matrix. The cluster is fitted by *2N* axes called principal components by calculating the eigenvectors of the covariance matrix. The eigenvectors create a coordinate system called *eigenspace*. The eigenvalues are also obtained to represent the degree of exhibited variation by each principal component.

3.2.5. Eye shape model

An eye shape is represented by a point in the eigenspace. The point possesses projections on the bases of the eigenspace $B = b_1, b_2, b_3, \ldots\ldots b_{2N}$. Thus, the eye shape can be **perfectly** represented as a linear combination of the principal components using these projections as follows:

$$\grave{X}_i = AB_i \Rightarrow X_i = \bar{X} + \acute{X}_i = \bar{X} + AB_i \tag{3.16}$$

It can be observed that the mean eye shape is represented by a point located at the origin of eigenspace because its projections have zero values. On the other hand, the projections of a point in the eigenspace can be obtained by Eq. (3.17).

$$B_i = A^{-1}(X_i - \bar{X}) \tag{3.17}$$

As the principal components fit the shape cluster in order to maximize the variation, the most significant information is carried out by the first few components. This fact is proven by descending order of the corresponding eigenvalues. An eigenvalue is related to variance σ by Eq. (3.18).

$$\lambda_i = \sigma_i^2 \tag{3.18}$$

Therefore, an eye shape can be **approximately** reconstructed by the first few components and the matrix *A* is reduced to $\acute{A}$ accordingly as well as Eq. (3.16) becomes:

$$X_i \approx \bar{X} + \acute{A}B_i \tag{3.19}$$

The first *k* eigenvectors are usually kept to explain 95% of variance that exhibited in the aligned shapes. The number *K* of eigenvectors to satisfy this percent is estimated using Eq. (3.20).

$$V = \sum_{i=1}^{2N} \lambda_i \quad \Rightarrow \quad \sum_{i=1}^{k} \lambda_i > PV \tag{3.20}$$

where *P* is the proportion of the total variation to be kept (95% in our case).

3.2.5.1. Testing of the eye shape model

Let us take a moment and think about the eye from the motion perspective. We can deduce that the eye shape moves vertically to identify the eye state. By returning to Fig. 3.5, it can be observed that the change of distribution at each eye shape point is approximately same. This issue is very important to prevent the change of any point to dominate the variance representation by the principal components. Thus and by using the *M* aligned shapes, points' distributions, it is guaranteed that each principle component nearly represents different pattern

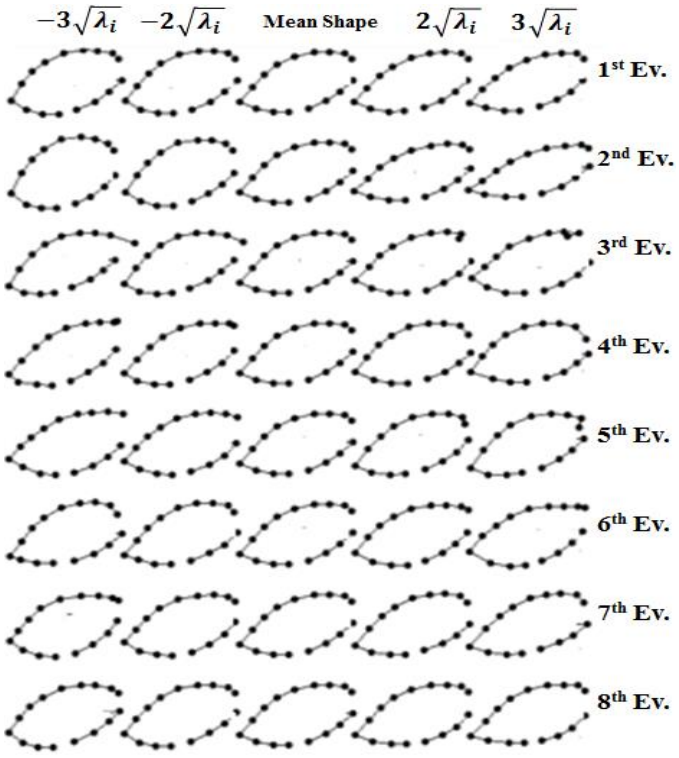


Fig. 3.7 Eigenvector behaviors.

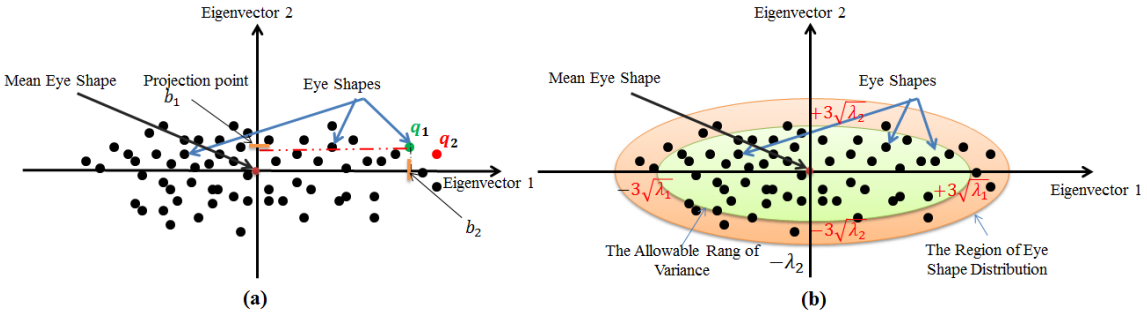


Fig. 3.8 Eye shape in eigenspace. (a) Two eye shapes with different positions. (b) Allowable range of eye shape generalization in eigenspace.

of changing of the eye shape points.

The mean eye shape is located at the origin of the eigenspace and it has a zero projection vector *B*. If each element of this vector is changed individually and independently with preserving the others at zero, only one eigenvector will contribute in constructing the eye shape using Eq. 3.19. Hence, the changing pattern of the eye shape that represented by this eigenvector can be discerned. Figure 3.7 demonstrates the first eight eigenvector behaviors in the range of $\pm 3\sqrt{\lambda}$. One can deduce that the middle shape is the mean shape as well as the exhibited variance is reduced as the eigenvector index decreases.

3.2.5.2. Generalization of the eye shape model

As mentioned before, the eye shape model must have capability of generalization; it should generate new shapes that not exist in the aligned shapes. The distribution of the eye shapes creates a volume/cluster that fitted by the eigenspace. This volume is the region of generating new shapes. In other words, suppose that the eigenspace is created by the first and second eigenvectors, and we have an aligned eye shape that represented by point q_1 in this eigenspace. If a new point q_2 is dropped beside q_1 with a slight shift in the positive direction of the first eigenvector, this means that the new point represents a legal eye shape with some slight difference in the eye state of q_1 as depicted in Fig. 3.8(a).

The issue arises here is " how much we can go far to change the position of the dropping

point inside the eigenspace?" which is equivalent to " what are the boundaries/limits of the volume on each eigenvector?". As long as the variance σ exhibited by each eigenvector is described by the corresponding eigenvalue, Eq. (3.18), the allowable range can be estimated based on Gaussian distribution that "the best representation of data distribution is housed in the range of $\pm 3\sigma$" [59]. Accordingly, a green region has been created in Fig. 3.8(b) to highlight the allowable range of generalization.

3.2.6. Refining eye shape deformations

In the previous section, the generalization capability of the eye shape model has been proven. This model can be utilized to refine eye shape deformations. Deformations are produced by searching on new locations of eye shape points. Suppose that we have an eye shape with some deformations as shown in Fig. 3.9. These deformations indicate that the eye shape is not located in the allowable range of generalization. Consequently, some of its projections are out of some

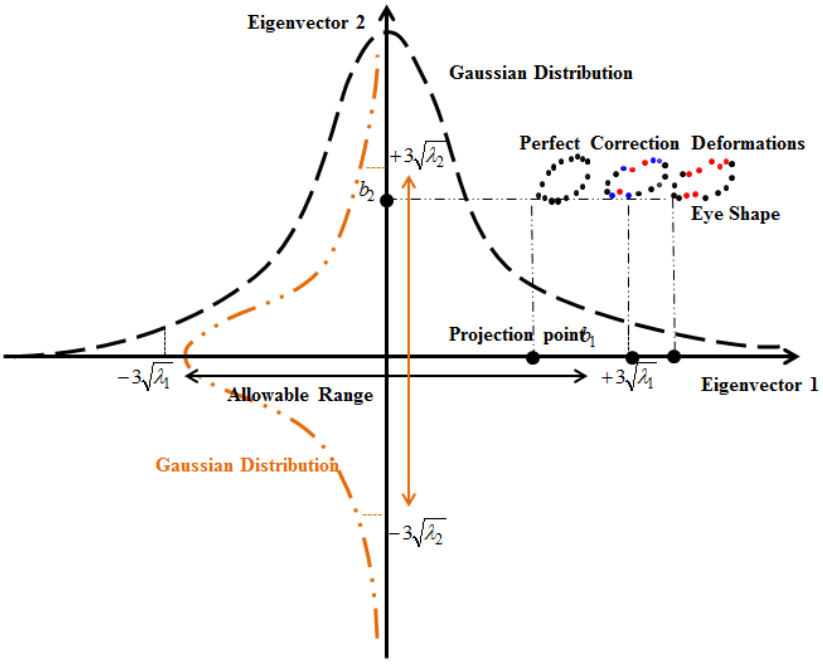


Fig. 3.9 Refining operation (correction deformations and enhancing the eye shape).

eigenvectors' boundaries. The simplest approach to correct the wrong projections is to enforce their values to be at the eigenvectors' boundaries. This can be done as follows:

- The eye shape is projected to the eigenspace using Eq. (3.17) and the corresponding projection vector *B* is obtained.
- Each element b_{th} in the projection vector is checked individually according to the corresponding eigenvalue in the range of $\pm 3\sqrt{\lambda_{th}}$.
- Based on the element's sign, each outrange element is shifted to the limit of the allowable range. Mathematically,

  If $b_i > +3\sqrt{\lambda_i}$ then $b_i = +3\sqrt{\lambda_i}$,

  else, if $b_i < -3\sqrt{\lambda_i}$ then $b_i = -3\sqrt{\lambda_i}$.
- By using Eq. (3.18) and the modified projection vector, the eye shape is reconstructed with much less deformations as illustrated in Fig. 3.9.

Obviously, the perfect shape can be obtained by shifting the element inside the allowable range. This procedure is achieved by the searching operation iteratively as detailed in the next section.

At the end of this step, the eye shape model has been created using the aligned eye shapes, mean eye shape which represents the general eye state has been obtained and the capability of the model's in refining deformations and generalizing new eye shape have been checked.

3.3. Modeling pixel values

After modeling eye shape variations, the change of pixel values around the eye contour must be modeled. Any region can be considered and studied. Let us take a moment and focus on an eye image such as the one shown in Fig. 3.10(a). The eye contour is the boundary between flesh and eyeball. Any point on this boundary can be considered as an exact eye shape point.

The existence of eyelids makes a sharp feature of the eye contour to characterize the eye shape points. As long as the eyelids move vertically, any point on the upper part of the eye contour has some pixels to represent flesh, some pixels to represent upper eyelid and some pixel to represent eyeball and vice-versa for the lower part of the eye contour. The two eye corners can be detected horizontally and it has no pixels to represent eyelids. Based on this characterization and from a simple point of view, four patters of pixel values can be taken into account; two patterns for left and right eye corners; respectively and two patters for upper and lower eyelids; respectively.

On the other hand, two factors can affect the above characterization in order to represent eyeball pixels; the thickness of eyelids and the iris/pupil location. Therefore, it is better and less risky to create a pixel value model for each eye shape point instead of using only four models. Furthermore, the eye images in database must be selected carefully to contain as various states

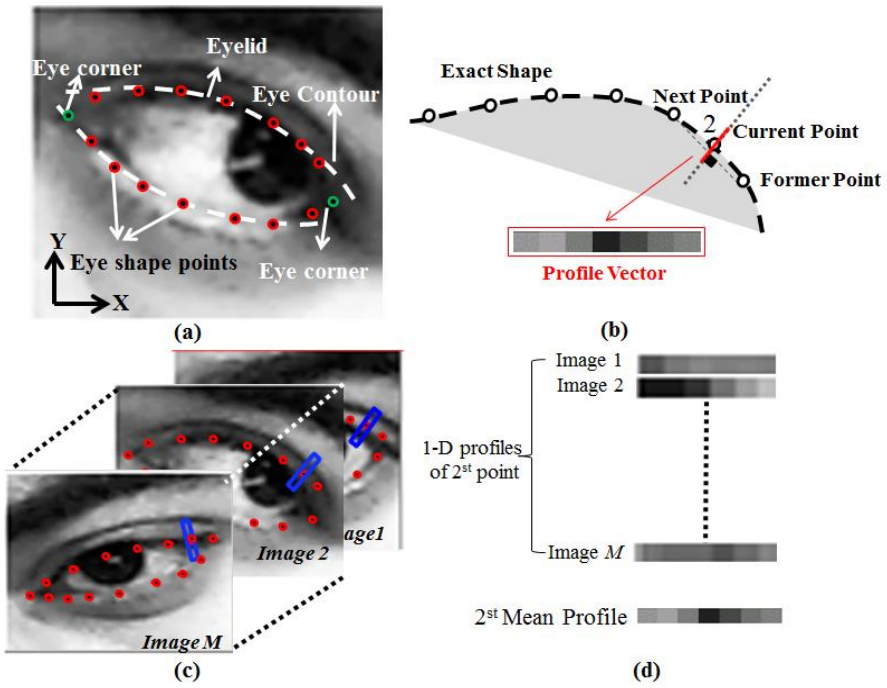


Fig. 3.10 Pixel value modeling. (a) Eye image. (b) Pixel profile extracting. (c) Collecting pixel profiles of the second point over *M* images. (d) Collected profiles and corresponding mean profile.

of these two factors as possible.

Based on the previous discussion, ASM was designed to model the pixel values along a line passing through an eye shape point in the vertical direction and perpendicular to a line that connects the following and former points as demonstrated in Fig. 3.10(b). This procedure is called *mapping strategy*. Figure 3.10(c) summarizes this strategy in order to model the pixel values at the second labeled eye shape point using eye images in database. This operation can be described generally for any point as follows:

For i^{th} eye shape point;

1) For j^{th}eye image in database;

   a) Move the pixel values that are taken according to the mapping strategy into a vector g, namely profile vector as shown in Fig. 3.10(d).

      $g_{ij} = (I_1, I_2, \dots I_l, \dots I_z)$

      where g_{ij} is a *1-D* profile of i^{th}point in j^{th}eye image, $I_l$is the intensity value of l^{th}pixel and *z* is a fixed number of pixels.

2) The *M* profile vectors of i^{th} eye shape point are gathered in a matrix G_i and averaged to obtain the mean profile vector $\bar{g}_i$ as shown in Fig. 3.10(d).

$$\bar{g}_i = \frac{1}{M}\sum_{j=1}^{M} g_j \tag{3.21}$$

   The mean profile vector $\bar{g}_i$ is considered as a common descriptor of the pixel values around i^{th} eye shape point.

3) A covariance matrix CG_i of the *M* profile vectors is calculated using the corresponding mean profile vector.

$$CG_i = \frac{1}{M} G_i G_i^T \tag{3.22}$$

These procedures are applied for *N* eye shape points. By the end of this step, each eye shape has

a mean profile vector to describe the common pixel values and a covariance matrix to express the change of pixel values in the corresponding profile vectors.

By the end of the training stage, mean eye shape is obtained to describe the common/ general eye state, an eye shape model is created to represent the change of the eye shape, *N* mean profile vectors are constructed to describe the common/general pixel values at the eye shape points according to the mapping strategy and *N* covariance matrices are calculated to represent the change of pixel values at the eye shape points.

3.4. Detection stage

ASM can be used either for static or real time applications. The aim of this stage is to detect the eye contour in a testing eye image based on the models that created in the training stage. Before we go deeper, we want to refer that the next section explains the detection operation ideally

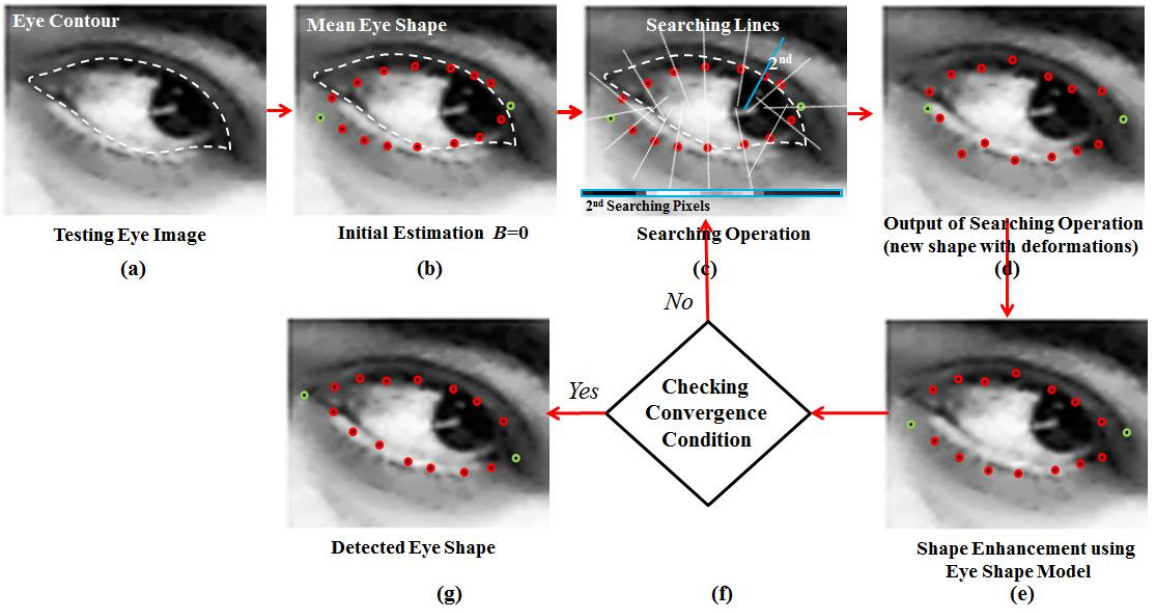


Fig. 3.11 ASM detection stage. (a) Testing eye image. (b) Fitting initial estimation. (c) Searching operation. (d) Searched eye shape. (e) Enhancing deformations. (f) Convergence condition checking. (g) Detected eye shape.

without mentioning on problems that could be faced.

3.4.1. Eye shape detection

ASM uses an iterative strategy to search on the eye contour. Figure 3.11 illustrates the detection stage by a simple flowchart. The flowchart consists of blocks that highlight the particular operations. These operations are detailed in the following steps:

First, a testing image is presented to the algorithm as shown in Fig. 3.11(a). The eye contour is the target that the eye shape points must be located at.

- ASM uses the mean eye shape as an initial estimation of the eye contour in the testing image. The mean eye shape is located at the origin of the eigenspace and has fixed shape parameters. The mean shape can be generated using the eye shape model by substituting a zero projection vector *B* in Eq. (3.16). Figure 3.11(b) shows the testing eye image is fitted by the mean shape.
- In the next step, ASM tries to update the locations of mean eye shape points by searching on the eye contour. ASM searches on the eye contour based on the mapping strategy. Figure 3.11(c) shows a searching line created at 2nd shape point. The covered pixels by this line are contained in a vector g which is *v* dimensions vector, *v*>>*z*. Using the 2nd mean profile $\bar{g}_2$ and the second covariance matrix CG_2, ASM searches in this vector on a segment g_u ($g_u$is 1x*z* vector) that its pixel values minimizes Eq. (3.23). The location u of this segment is expected to be at the eye contour (exact eye shape point) or close to it.

$$\mathcal{F}_2(g_u) = (g_u - \bar{g}_2)^T G_2^{-1}(g_u - \bar{g}_2) \quad (3.23)$$

By applying this procedures on each eye shape point with taking into account the use of corresponding mean profile and covariance matrix, the locations of eye shape points are updated and a modified eye shape is obtained as shown in Fig. 3.11(d).

- The modified eye shape is expected to have deformations because of many factors that influence the searching operation such as different lighting conditions and eye states. Intuitively, the eye shape model will be used to refine these deformations by checking the elements of the projection vector of the modified eye shape as explained in *section* 3.2.6.
- The last step in ASM strategy is to check the convergence condition. If it is not satisfied, the corrected projection vector is used to generate the corresponding eye shape using Eq. (3.19). The generated shape is fitted to the testing eye image and a new cycle of searching operation is conducted as shown in Fig. 3.11(f), flowchart.

The above steps are repeated until the convergence condition is satisfied and the last modified eye shape is considered as the detected eye shape in the testing eye image. Therefore, it is considered as the exact eye shape as illustrated in Fig. 3.11(g).

3.5. History of developments

At the first implementation, ASM was incapable to process the shapes of grayscale objects. The object is transformed into a binary image by passing a filter over the gray image and then ASM is run to search on the object boundary. Due to the strong interest in detecting various object shapes, the dealing with grayscale objects has been imposed. Therefore, many approaches have been proposed to improve the ASM performance in gray image domain [60-66].

Cootes has used a local gray-level pyramid in order to increase the robustness of the searching operation against illumination changes [60]. The idea is inspired by the human visual system. The brain creates a primed of the scene being seen in order to facilitate the object interpretation, recognition and other operations as detailed in chapter 2. Therefore, Cootes tried to process the objects level by level starting from the image which has less details and resolution. This image gives significant information on the object edge. Thus, it is the most

appropriate domain for initiating the searching operation. The same procedures of calculating mean profile vector and covariance matrix are applied for each eye shape point in the gray level pyramid. More simply, if the pyramid is created with four levels, an eye shape point at the end of modeling operation will have four mean vector profiles and four covariance matrices.

In detection stage, searching operation is modified by creating a gray level pyramid on the testing image. The searching operation is applied on the testing image in the first level of pyramid that shows less information using the first level of covariance matrices and mean vector profiles of the *N* eye shape points. The searching operation is raised up and applied on the second level using the results of first level and so on.

Sukno and *et al*. [61] has proposed an approach in the same direction of enhancing the searching operation and representing the pixel values at the eye shape points. They used a non-linear classifier to label the shape points near the object boundary. The advantage is that the classifier makes the description of pixel values around the shape points invariant to rigid transformation. Baram and *et al*. [62] have employed *K*-Nearest Neighbor (KNN) to search on the segment in the searching line that best matches the mean profile vector. Mahoor and *et al.* [63] have incorporated RGB color information for localizing eyes and mouth in face images. Moreover, the color information is also used in pixel modeling at shape points. Log-Gabor features have been used to describe the local 1D texture at the eye shape points [64]. Gabor wavelet transformation can be used to extract the 2D features of shape points because its coefficients contain rich local-texture information [65].

3.6. Problem statements

The improvement approaches in the previous section can be classified according to three strategies; enhancing the description of texture around eye shape points like using Log-Gabor

features, enhancing the matching measure process in order to determine the new location of shape points like using KNN and changing the mechanism of searching operation like building gray level pyramid. An extra strategy that rarely discussed in literature is enhancing the initial estimation of eye shape parameters [66].

Based on the detection stage explanation, it can be deduced that initial estimation and searching operation are the most important issues to be considered in enhancing ASM performance. Initial estimation of the eye shape affects the behavior of searching operation and number of iterations. If the initial shape was very close to the eye contour, searching operation would just become a tuning process for shifting points smoothly. Consequently, the number of iterations consumed in searching operation can be significantly decreased.

On the other hand, the searching operation is used iteratively to detect the exact eye shape under different lighting conditions and eye states. It must be robust to prop the decision of the initial estimation in the first iteration and to smoothly process the modifications of the eye shape in next iterations. Moreover, the initial estimation may not be appropriate for some complex eye states. Thus, the aim is to determine the path to the eye contour when the eye shape is badly initialized. Otherwise, detection operation ends up with shrinking the eye shape into a point or expanding it out of the image or running in an infinite loop.

ASM uses mean eye shape as an initial estimation of the eye shape for any testing image. The mean eye shape represents one eye state at fixed scale factor, rotation angle and translation. On the other hand, ASM uses only mean profile vectors to modify the locations of eye shape points. Figure 3.10(d) shows a *1-D* profile which lies in one direction. Obviously, it does not provide significant information on the texture variation at the shape points. Morever, the few contained pixels make ASM incapable to detect the eye shape under various lighting conditions.

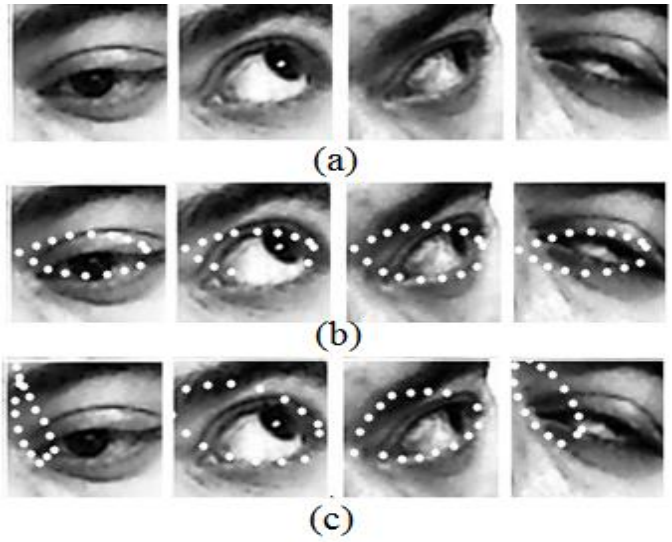


Fig. 3.12 Consequences of bad initial estimations. (a) Testing eye images. (b) Fitting the mean eye shape as initial estimations. (c) Detected eye shapes.

Figures 3.12(a,b) show a set of testing eye images fitted by the mean eye shape. Each eye image highlights different conditions. The eye in the first image is approximately straight without rotation angle. Eye is slightly rotated in the second image with wide opening angle. The third image represents the normal condition of the eye with a distance from the camera (different scale) and a considerable rotation angle. The eye is nearly closed with a considerable rotation angle in the fourth image. Figure 3.12(c) illustrates the consequences of running the standard ASM to detect the exact eye shape.

In the first image, a translation in ***Y*** direction occurs. The upper eyelid is very close from the lower eyelid. Therefore, the searching operation is not able to discriminate the correct location of lower and upper eyelids. The second image shows the left upper points are attracted to the eyebrow because the initial estimation was very close to the eyebrow as well as the searching operation did not *recognize* the upper eyelid and the eyebrow. In the third image, because of large difference in rotation and the eye has a smaller scale factor than the mean eye shape, the most of searching lines do not house the eye contour. The fourth image can be

considered as a complex case that combines the whole conditions in the previous three cases to shrink left the eye shape into a smaller shape.

In the next section, we improve the ASM performance by enhancing the initial estimation of the fitting operation as well as a robust searching operation is introduced.

3.7. Improved Active Shape Model

The ASM operations have been ideally explained in *section* 3.4.1. In fact, ASM has low capabilities to deal with various eye profiles due to many problems such as the different lighting conditions, scales, textures and positions. These problems may lead ASM to fail in detecting the eye shape. Moreover, there are many issues must be taken into consideration in real time processing such as the accuracy and time consumption.

3.7.1. Improved initial estimation and shape parameters

ASM uses the mean eye shape as an initial estimation of the eye contour in testing images. The mean eye shape has fixed parameters. Therefore, ASM is said to be a blind operation because it has no prior knowledge of where the eye lies or what are the translation, rotation angle, scale or eye state conditions. In order to give ASM more flexibility to deal with complicated eye states and to initialize a good estimation of eye profile parameters, object recognition based-PCA is employed to understand the topological eye structure using the eye image database [67]. Based on this understanding, an appropriate eye shape is initially selected and fitted in the testing images.

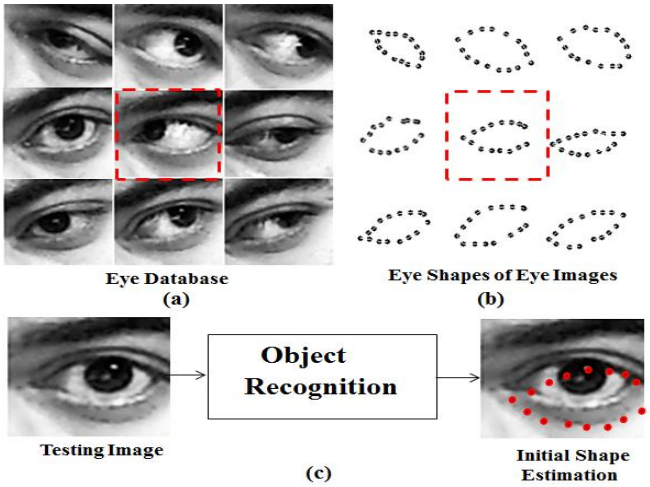


Fig. 3.13 Initial estimation enhancement. (a) Eye database. (b) Manually labeled eye shapes. (c) Initial estimation of eye shape based object recognition.

Suppose that the eye image database contains 9 eye images as shown in Fig. 3.13(a). Figure 3.13(b) shows the corresponding manually labeled eye shapes. Figure 3.13(c) shows a testing eye image $\Gamma_{unknown}$. Object recognition tries to find an image Γ_i in database that best matches the topological structure of the eye in $\Gamma_{unknown}$ as enclosed by a red rectangle in Fig. 3.13(b). The eye shape of Γ_i is then fitted as the initial estimation of the eye contour in $\Gamma_{unknown}$ as illustrated by the red shape in Fig. 3.13(c).

3.7.1.1. Object Recognition

Object recognition has become an important application in image processing [68-70]. It is being involved to achieve many tasks such as security systems, credit card verification and criminal identification systems. There are many proposed techniques for object recognition. In this research, we consider PCA as the key point to achieve this task due to its powerful capabilities such as data reduction, high accuracy and simple implementation [71].

Generally, object recognition is a technique that compares an image with those in database. Thus, the comparison between two images of the same object is an essential problem that must be solved for several issues:

-The objects can be captured at various conditions such as lighting, rotational, scale, position…etc.

-Comparison must be done with low time consumption and high accuracy.

In this section, PCA is approximately described in the same context of *section* 3.2.4 with respect to the image domain not to the shape domain. Therefore, the *M* eye images in database are converted into vectors by concatenating each row with the previous one.

$$\Gamma_i = (I_{00i}, I_{01i}, \dots, I_{jki}, \dots, I_{0ni}, I_{10i}, \dots \dots, I_{mni})^T \quad (3.24)$$

where Γ_i is the vector of i^{th}eye image in database, $I_{jk}$is the pixel value at j^{th}row and k^{th} column, *m* is the number of rows (height) and *n* is the number of columns (width) .

As the database contains eye images only, we expect that they form a narrow cluster in the image domain as demonstrated in Fig. 3.14(a). An eye image is represented by a point in *nxm* dimensions space. This point moves within the cluster according to the change of a few eye parameters such as eye state, scale, rotation, pupil location…etc. Based on this fact, the major and minor axes of the cluster describe different patterns of the change of eye parameters. Consequently, some of these patterns can be utilized to identify the state of the eye in testing

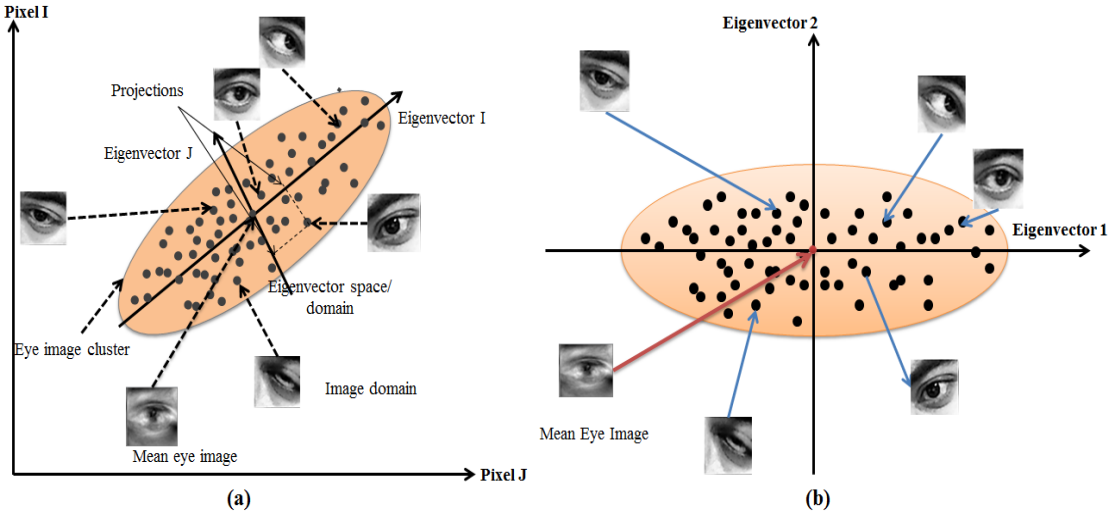


Fig. 3.14 Eye image cluster. (a) Eye images in image domain (pixel domain). (b) Eye images in eigeneye space.

images. PCA is utilized to describe the patterns by fitting the cluster by a set of eigenvectors in directions of the maximum variation as shown in Fig. 3.14(b).

The mean eye image $\bar{\Gamma}$ of the *M* eye images is then calculated.

$$\bar{\Gamma} = \frac{1}{M}\sum_{i=1}^{M} \Gamma_i \tag{3.25}$$

The mean eye image is subtracted from each eye image in the database and the *M* vectors are contained in a matrix ϕ. The covariance matrix *CI* is then calculated.

$$CI = \frac{1}{M}\phi\phi^T \tag{3.26}$$

The covariance matrix provides a significant representation of how each pixel value varies with respect to the other pixels changes. In order to discern patterns of these changes, the eigenvectors of the covariance matrix CI are computed.

$$CI\Lambda = \mu\Lambda \tag{3.27}$$

The obtained eigenvectors are called eigeneye vectors and arranged in a matrix Λ according to their corresponding eigenvalues μ in descending order (from largest to lowest).

One issue can be highlighted here is that as long as ϕ is a matrix consists of *M* rows and *n*x*m* columns, the dimension of the covariance matrix is very huge, *nm*x*nm*, and the corresponding eigenvectors may not be computed .

The covariance matrix is calculated using Eq. (3.26). On the other hand, $CII = \frac{1}{M}\phi^T\phi$ is a matrix with *M*x*M* elements and it has *M* eigenvectors calculated by $CII\omega = \nu\omega$. Multiplying both sides by ϕ we obtain, $\frac{1}{M}\phi\phi^T\phi\omega = \nu\phi\omega$. Comparting this equation with Eq. (3.26), we deduce that $\Lambda = \phi\omega$ and $\nu = \mu$. Accordingly, CII is firstly computed with the corresponding eigenvectors then Λ can be consequently calculated.

Each eye image Γ_i in database can be constructed by a linear combination of the

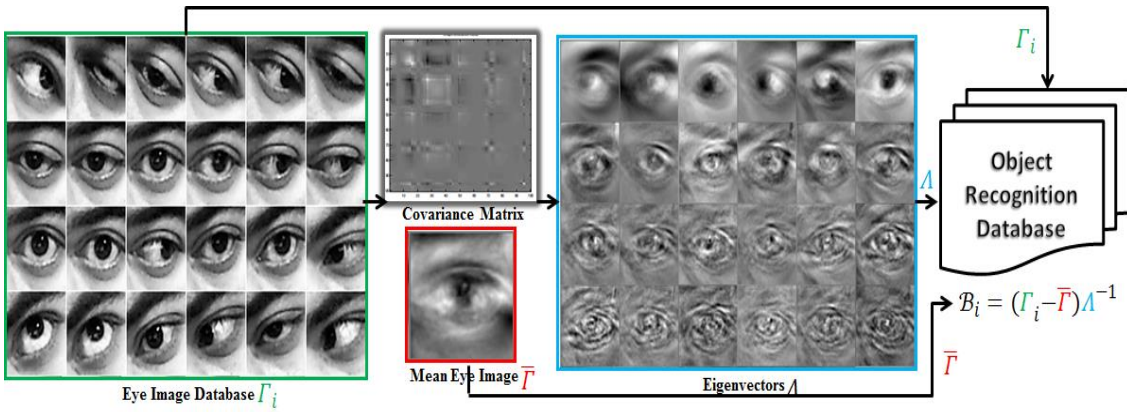


Fig. 3.15 Object recognition based PCA.

eigeneyes as shown in Fig. 3.15.

$$\Gamma_i = \bar{\Gamma} + \Lambda\mathcal{B}_i \tag{3.28}$$

where $\mathcal{B}_i = \mathfrak{b}_1, \mathfrak{b}_2, \mathfrak{b}_3, \dots, \mathfrak{b}_M$ is the projection vector which determines the position of the eye image in the eigenspace as it is represented by a point. The position (projection vector) of an eye image in the eigenspace can be obtained by Eq. (3.29).

$$\mathcal{B}_i = (\Gamma_i - \bar{\Gamma})\Lambda^{-1} \tag{3.29}$$

It can be observed that the mean eye image and eigeneyes are fixed. Therefore, the projection vector can be considered as the image representation in the eigenspace instead of thousands of pixels. By using Eq. (3.29), each eye image in the database is projected into the eigenspace and the corresponding projection vector is obtained. The *M* projection vectors are stored in a new database called Object recognition database ψ as flowcharted in Fig. 3.16.

$$\psi = \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \dots, \mathcal{B}_M$$

3.7.1.2. Understanding eye topological **structure**

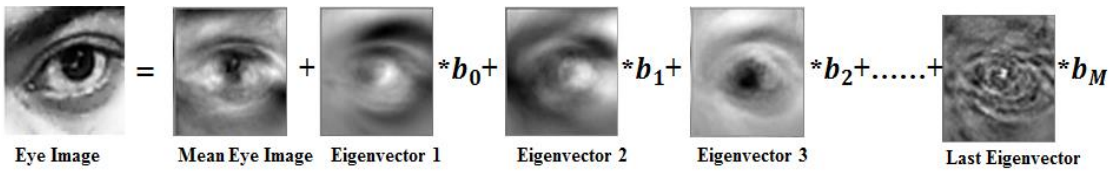


Fig. 3.16 Eye image reconstruction.

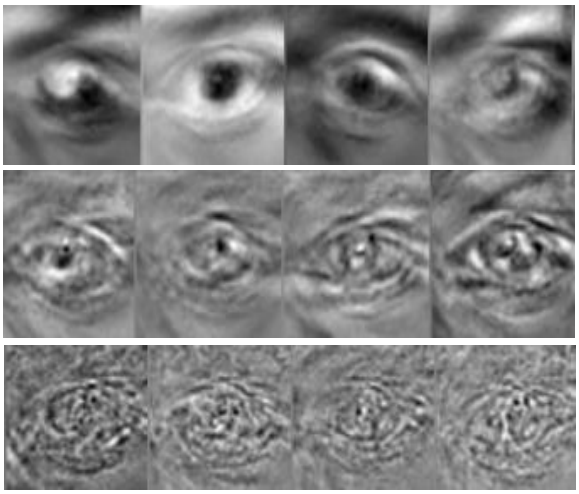


Fig. 3.17 Object recognition based PCA.

PCA is employed to understand the topological structure of the eye rather than to recognize a specific person [71]. The eye images in database are carefully chosen to accomplish as various scales, rotations eye states as possible. These parameters are expected to dominate the first few eigeneyes. Figure 3.17 proves this fact by showing in each row, the first, intermediate and last four eigeneyes; respectively. The first eigeneyes with largest eigenvalues contain information relative to the eye structure. These vectors are useful in understanding the eye structure. Eigenvectors with intermediate eigenvalues illustrate information that common to particular eye components such as pupil location and eye corners. Therefore, they can be used to give initial positions of these components. The last eigeneyes have smallest eigenvalues, represent the noise patterns in database and can be used to smooth the eye texture.

Obviously, the most significant information to understand the eye topological structure is encoded in the first eigeneyes. These eigeneyes are contained in a new matrix $\acute{\Lambda}$. An eye image can be approximately reconstructed using this condensed matrix using Eq. (3.30). Consequently, the projection vector $\acute{\mathcal{B}}_i$ of an eye image Γ_i is reduced to only few elements. The elements describe the most significant information on the eye structure and hence a condensed new object recognition database $\acute{\psi}$ is obtained as well.

$$\Gamma_i \approx \bar{\Gamma} + \acute{\Lambda}\acute{\mathcal{B}}_i \tag{3.30}$$

where $\acute{\mathcal{B}}_i = \acute{\mathfrak{b}}_1, \acute{\mathfrak{b}}_2, \acute{\mathfrak{b}}_3, \dots, \acute{\mathfrak{b}}_K; K \ll M$ is the condensed projection vector which consist of few elements that represent the projections on the first few eigeneyes in $\acute{\Lambda}$. Accordingly, time consumption of comparing two eye images is significantly reduced.

Based on the previous explanation, in detection stage and once a testing image is presented, the improved ASM projects it into the condensed eigenspace and the corresponding projection vector is obtained. The projection vector is compared with those in the object recognition database $\acute{\psi}$ in order to search on an image that best matches the topological structure of the eye in the testing image. As long as the eye shapes of the images in database are known and manually labeled, the eye shape of the closest image is considered as the initial estimation to be fitted to the testing image.

Figure 3.18(a) shows a set of images that have been used in Fig. 3.12. Some images from the database are found to best match the topological structures using PCA as illustrated in Fig. 3.18(b). The eye shapes of these images are considered as initial estimations as shown in Fig. 3.18(c).

Fig. 3.18 Improved initial estimation. (a) Testing images. (b) Eye structure understanding by PCA (c) Fitting testing images by initial estimations suggested by PCA.

3.7.2. Improved searching operation

The robustness of searching operation determines the number of iterations as well as the accuracy of ASM. ASM uses mean profile vectors to estimate the displacements of eye shape points. This weakens the performance of ASM against the change of lighting conditions, eye state and shape parameters.

In the improved ASM, we extend the description of the local structure at eye shape points to sub-images. A sub-image lies in two directions and outnumbers the pixels in a mean profile vector. Therefore, it provides significant information and increases the robustness of searching operation.

In training stage, we follow the same mapping strategy of the standard ASM to model pixel values around a shape point. The difference is that a sub-image is taken instead of a profile vector as shown in Fig. 3.19. The mean sub-image is also calculated over the *M* collected sub-images of each eye shape point. By the end of this step, if the eye shape is represented by *N* points and there are *M* eye images in the database, each eye shape point has a set of *M* sub-images and a mean sub-image.

In detection stage, we also follow the same strategy of creating searching lines. A sub-image along the searching direction that best matches the mean sub-image is considered as

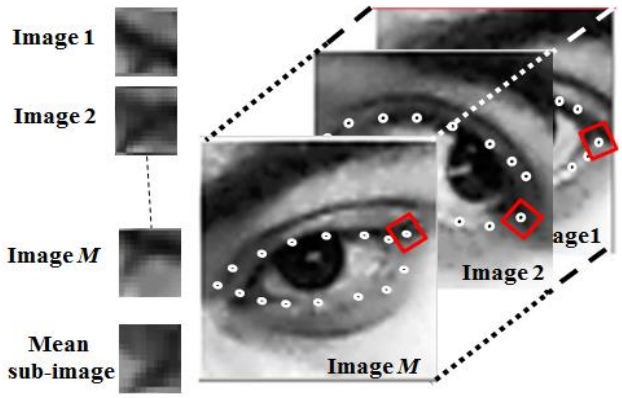


Fig. 3.19 The sub-images collected for the first eye shape point (eye corner).

the target and its location is the new location of the processed eye shape point.

The sub-image approach imposes the following three conditions:

- The searching operation must be able to measure the similarity between sub-images along a searching line and the collected sub-images for each shape point. This property ensures distinguishing the eye contour from the skin, eyeball and eyebrow.
- Searching operation must have an ability of generalization. Generalization means that the searching operation can predict/detect the target sub-image (location) even if it is not presented by the collected sub-images in the training stage. This property provides more flexibility to deal with various eye profiles and lighting conditions.
- Searching operation must be achieved very fast as less time consumption as possible. According to the sub-image approach, for each shape point, the mean sub-image will be compared with sub-images taken along the searching line. This means in each searching step, two sub-images will be compared.

In order to satisfy the above conditions in training stage, PCA is used to model the variation of pixel values that are exhibited in the collected sub-images. According to the same procedures explained in *section* 3.7.1.1, for i^{th} eye shape point, a covariance matrix is calculated using the corresponding *M* collected sub-images and mean sub-image. The eigenvectors and eigenvalues of the covariance matrix are computed. The first few eigenvectors which have the largest eigenvalues are taken to describe the variation of the local structure (pixel values) around the i^{th} processed point. These eigenvectors create an eigenspace called eigensub-image space of i^{th} point.

By the end of this step, each eye shape point has a mean sub-image and an eigensub-image space.

In detection stage: time consumption is sharply reduced because a sub-image is represented in an eigen sub-image space by the corresponding projection vector. The projection vector consists of a few elements. On the other hand, generalization is guaranteed by the elliptical distribution that formed by the first few eigenvectors (eigensub-image). Any sub-image belongs to this ellipse can be considered as a new representation of the i^{th} point's local structure. Moreover, distinguishing the eye contour from skin, eyeball and eyebrow is achieved inside the eigensub-image space by measuring the distance between the projection vectors of the mean sub-image and those of the sub-images along the searching line. The location of the sub-image that best matches the mean sub-image is considered as the new

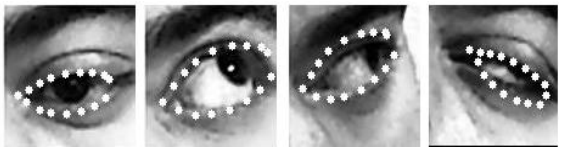


Fig. 3.20 Detected eye shapes.

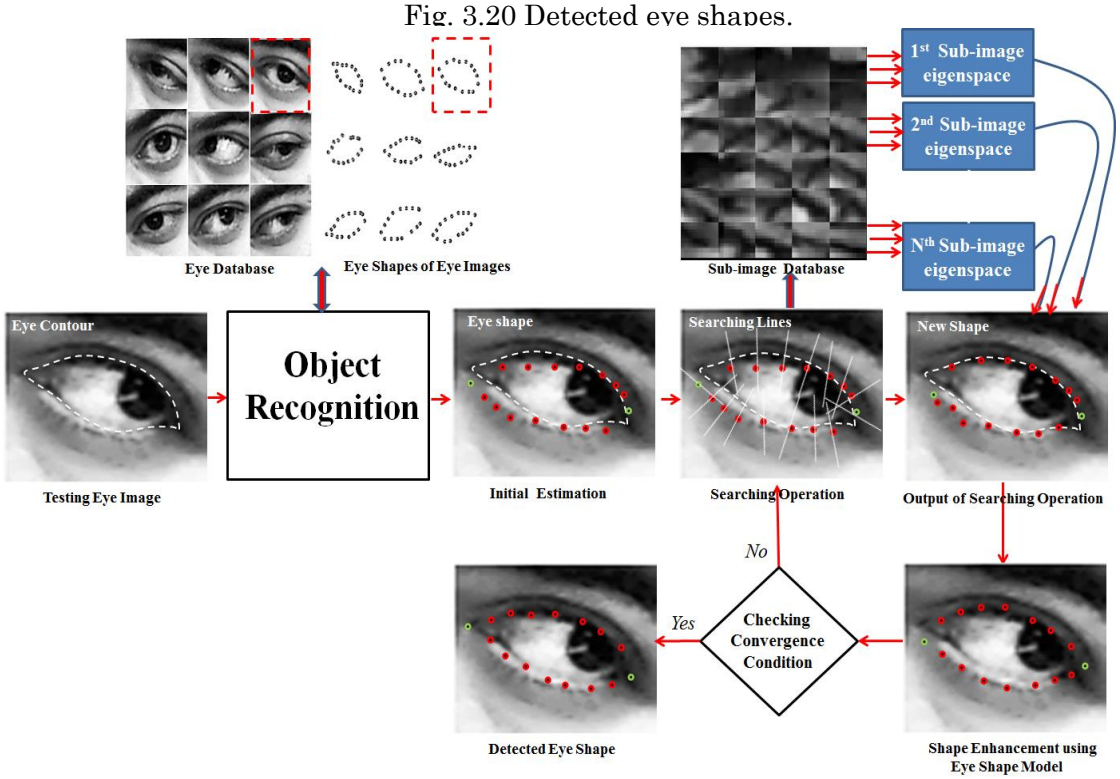


Fig. 3.21 Improved ASM structure.

location of the i^{th} point.

Figure 3.20 shows the detected eye shapes based on the improved estimation and improved searching operation.

Finally, the improved ASM flowchart is modified to represent all operation as shown in Fig. 3.21.

3.8. Experimental results and configurations

3.8.1. Setup

The improved and standard ASMs have been implemented in Visual Studio C++ 2008 environment, CPU quad-core processor 2.4 GHz. Since the aim is to detect the eye shape in an eye image, we used an eye detection technique based on Haar-Like Feature method to obtain the eye images [72].

The number of collected eye images for the experiments is 150; *M*=100 images are used for eye database and 50 testing images are used for evaluating the performance of the standard and improved ASMs. The eye images are resized into 130x130. The exact eye shapes of 150 images have been labeled manually using *N*=16 points. Two points, 1st and 9th, represent the eye corners and the remaining points are equally divided to represent the upper and lower eyelids. The images in database are carefully selected to satisfy two conditions.

1) Slight differences in rotation, scale, translation and eye states (nearly closed, open and in-between) to strong up the initial estimate decision.

2) The images are taken under different lighting conditions to make the searching operation more effective using sub-images strategy.

The geometrical structure of the eye is analyzed by the first seven eigeneyes of the

matrix Λ. These eigeneyes creates the eigeneye space and describe 85% of the variation. Consequently, an eye image is represented by only seven elements that contained in the projection vector. For the standard ASM, at each shape point, a profile vector is taken containing seven pixels. The length of a searching line is 20 pixels.

Equation (3.31) is used to measure the best correlation between two projection vectors B_1 and B_2 either in eigeneye space or eigensub-image space.

$$Crr = \frac{B_1 . B_2}{|B_1||B_2|} \tag{3.31}$$

Equation (3.32) is used to measure the match error *Mrr* between a manually labeled eye shape, exact shape, X_1 of a testing image and the detected eye shape X_2 by the standard or improved ASM. Finally, each method converges when the difference between the eye shapes of last two iterations is less than α=0.005.

$$Mrr\% = \left[1 - \frac{X_1 . X_2}{|X_1||X_2|}\right] * 100 \tag{3.32}$$

3.8.2. Experimental Results

Figure 3.22 shows time consumption as a function of the number of eye images in database. PCA consumes time to recognize and understand the geometrical structure of the eye in a testing image. The relationship is linear because the eye images are represented by a fixed number of eigenvectors.

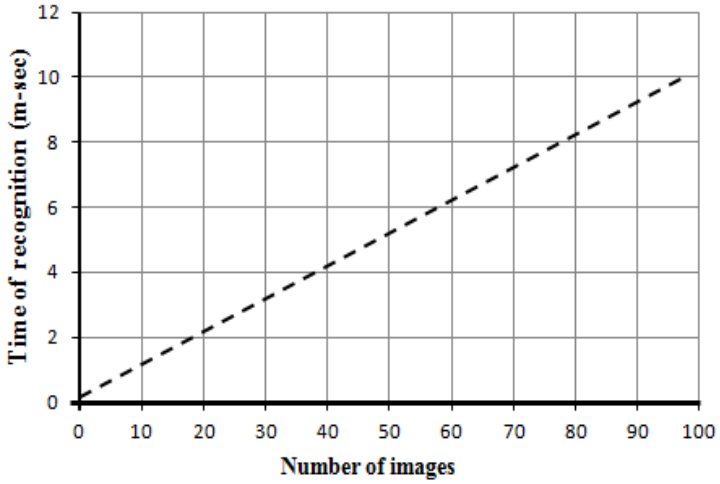


Fig. 3.22 The relationship between time of recognition and number of images in database.

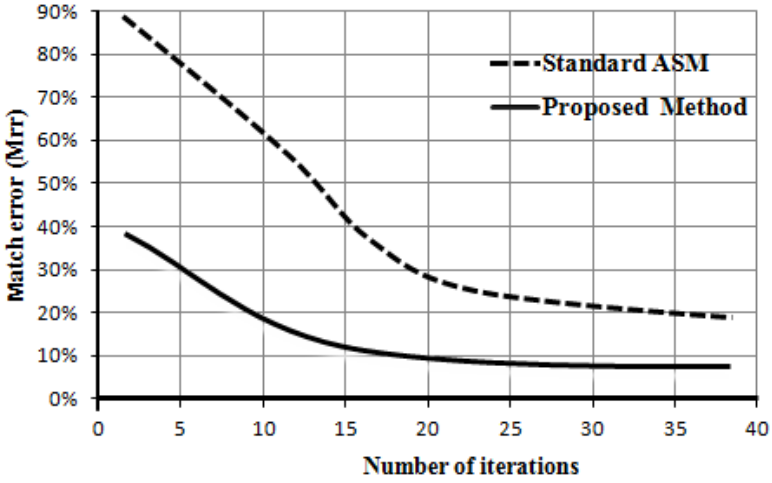


Fig. 3.23 The relationship between match error and number of iterations.

Figure 3.23 illustrates the relationship between the number of iterations and the match error. The curves demonstrate the development of the eye shape during the searching operation. The

beginning of each curve highlights the match error of the initial eye shape estimation. The ending points represent the match error of the detected eye shape.

Figure 3.24 represents the change of the match error with respect to the detection time. The curves can be utilized to evaluate the behavior of searching operation. The curve of the proposed method starts after around 10 msec that consumed by recognition operation.

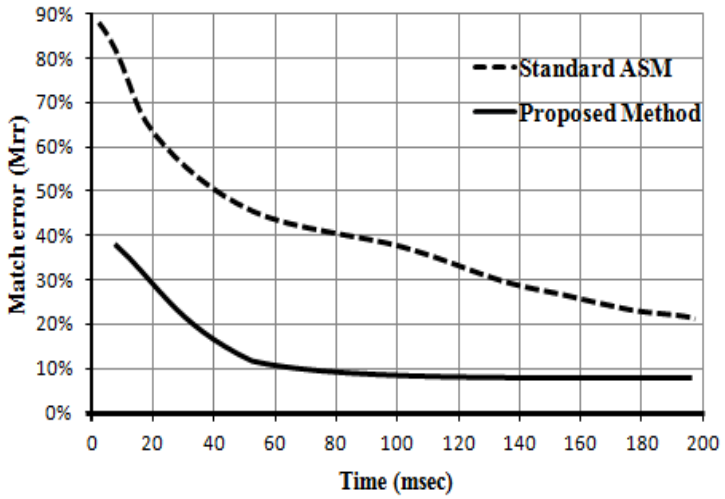


Fig. 3.24 The relationship between match error and time consumption.

3.8.3. Discussion

PCA is usually used for object recognition, especially for recognizing human faces [71]. In the improved ASM, PCA is employed for discriminating the eye profile at different parameters. Eye is surrounded by skin whereas human face is surrounded with various backgrounds. Therefore, the change of the eye profile is the dominant feature in the first few eigenvectors whereas in a face image, there are many features (nose, hair, two eyes states, background…,etc) that may dominate the first eigenvectors behavior frequently. Thus, we said earlier that PCA is used to understand the geometrical structure of the eye instead of recognizing the eye of a specific person.

In the proposed method, the initial estimation becomes more robust to process the change

of the eye structure. According to the conducted experiments, 100 eye shapes have been used as a training set. These eye shapes have different parameters (scale, rotation, translation and eye state). Based on object recognition and according to the eye profile in the testing image, one of these shapes is used as an initial estimate instead of always using the mean eye shape.

From another point of view, Fig. 3.22 shows that PCA consumes time to understand the eye geometrical structure. Thus, increasing the number of eye images in database must be taken into account especially in real time process such as tracking the eye shape in a sequence of images. 100 eye images consumes around 10msec which not affect the real time system comparing to the advantages gained by the improved initial estimation. Figure 3.23 proves this point by showing a big difference in the match error at the beginning of the curves. The standard ASM consumes a number of iterations to reach the initial match error of the improved ASM. This reflects significantly on time consumption as shown in Fig. 3.24.

The sub-image approach is used to reduce the number of iterations and increase the accuracy of the detected eye shape. The *1-D* mean profile vector that used by the standard ASM consists of 7 pixels and the number of pixels in a sub-image is 7x7 pixels. The sub-images are represented by only 7 projections in the eigenvector space. This means that the consumed time in detecting the new location of an eye shape point in both methods is same. On the other hand, two advantages are gained by using the improved searching operation. First, the sub image contains a larger number of pixels. Second, the correlation measurement is more robust in the eigenvector space. Figure 3.24 highlights that the searching operation becomes smoother and more stable compared to the standard ASM. Consequently, time consumption is reduced and high accuracy is achieved.

One may ask why an eigensub-image space is created for each eye shape point instead of one eigenspace for all points? Each eye shape point possesses a specific pattern of sub-images.

Logically and according to the conducted experiments, an eigenvector space tend to house a cluster such as that in Fig. 3.14(a). If sub-images of all eye shape points are combined to create one eigenspace, the tight elliptical distribution of the corresponding cluster is vanished and a random sphere is obtained. Consequently, non-linearity is increased and PCA loses the generalization property. On the other hand, an eye shape point possesses approximately the same local structure in the corresponding collected *M* sub-image. We expect that this makes somehow lighting condition dominate the eigensub-image spaces. Thus, ASM has a stable performance against the change of luminance.

Eye corner locations play the main role to determine the rotation angle of the eye shape whereas the other shape points can be thought of as the factors to describe the eye state, scale and translation. This little arrangement has showed a significant change in the ASM performance. Therefore in searching strategy, we search first on eye shape points that represent eyelids. Based on the new locations of these points, the eye corners are searched on.

3.9. Conclusion

In this chapter, the principles of implementing ASM have been detailed theoretically and mathematically. Two models of eye shape and change of pixel values have been constructed. The problems of the standard ASM have been highlighted with the corresponding consequences on accuracy and time consumption.

In the improved ASM, two approaches have been suggested for enhancing initial estimation of the eye shape in terms of eye state and shape parameters (scale, rotation and translation), and for enhancing searching operation in terms of dealing with different lighting conditions and discriminating the eye contour out of eyeball, eyebrow and flesh.

The initial estimation is enhanced by using object recognition to understand the topological

structure of the eye in a testing image. Based on the understanding, an eye image in the database is chosen as the best match of the topological eye structure. Accordingly, the eye shape of the chosen image should match closely the state and shape parameters of the eye in the testing image. Thus if the database contains *M* eye images, *M* manually labeled eye shapes are known and one of them will be suggested as an initial estimation instead of using the mean eye shape with fixed eye state and shape parameters.

The searching operation is enhanced by using the sub-image approach. The sub-image provides significant information on local structure and lighting condition of eye shape points. Measuring the correlation between sub-images in eigensub-image space has increased the robustness of determining the new location of points.

According to experimental results, the accuracy of the initial estimation has been increased around 50% compared to the standard ASM. The number of iterations has been decreased efficiently and consequently the detection time has been reduced to around 60 msec whereas the standard ASM needs more than 200 msec to reach the same level of accuracy.

Finally, we want highly to mention that the use of PCA was the essence of improvements whether in initial estimation or updating the locations of the eye shape points. Moreover, it is good to summarize the PCA advantages yielded in this chapter.

- In the eye shape modeling: PCA is used to model the change of the labeled eye shapes. Each principal component represents a pattern of this change. The eye shape model has a simple equation to describe these patterns, Eq. (3.19). By using this equation, new eye shapes can be generated easily. Therefore, the eye shape model can be used to refine these deformations.
- In the eye structure understanding: PCA is applied in image domain to demonstrate a manifold on the change of eye state and shape parameters. By carefully selecting the eye

images in database, the first principal components have been enforced to explain different patterns of the eye structure. Therefore, the eigeneye space represents different eye structure pattern in each particular position and a testing eye image can be evaluated based on its position in this space.

- In updating locations of eye shape points: PCA is employed to measure the similarity/correlation between two sub-images. The eigensub-image space represents in its volume the changes of lighting condition and local structure around the eye shape point. On the other hand, the representation of a sub-image was reduced to be same as the mean profile (seven elements of projection vector for sub-image and seven pixel values for mean profile). This is because of the powerful feature of PCA in data reduction with preserving the most significant information.

Chapter 4

4. Eye Shape Detection Based on Eye properties in Log-Polar Domain and Color Interpretation in Eigenspace

4.1. General

The improved ASM has increased the accuracy and reduced the time consumption. However, dealing with complex eye state is still a problem. A suggested solution is to increase the number of images in database to represent more different eye states. This leads to increase the recognition time accordingly. In order to reduce the number of images for representing particularly an eye state and increase the verity of describing different eye states in the database, the pupil is approximately filtered out. Thus, the eye structure dominates the first eigenvectors more efficiently. However, this solution imposes a very precise selection of images in database. Therefore, different scales, translations, eye states and rotation angles are still common problems to be dealt in real time applications.

In addition, the improved searching operation based sub-image has increased the robustness of ASM to deal with different lighting conditions. On the other hand, the performance is negatively influenced when the change of lighting occurs far from the middle range of brightness. In order to overcome this problem, ASM can be charged with various lighting condition databases. Based on lighting in a testing image, a database will be selected

and corresponding eigenvectors of object recognition and sub-images will be used to detect the eye shape.

An important issue that affects considerably on searching operation is the length of searching lines. Two contradicting requirements must be taken into account; the searching line should be long enough to house the target point as shown in Fig. 3.11(c). On the other hand, searching line should be short to reduce the searching time. If it is long and the target point is close to the current point position, it will be more probable to move far away and miss the target point.

The existence of the eyebrow affects negatively the ASM performance as well as it must be taken seriously into consideration in implementing any eye shape detection technique. The eyebrow possesses the same local structure of the eyelids. Therefore, the eye shape points could be attracted to the eyebrow instead of the upper eyelid. Moreover, the eyebrow may attract whole shape points if the initial estimation was not so appropriate.

In addition, ASM uses an iterative searching operation in order to detect the eye shape. As mentioned early, using an iterative strategy in real time applications is risky regardless the purposes. For example, the number of iterations that are needed to detect the eye shape in a testing image is not fixed and depends on the eye state. Consequently, time consumption is also not fixed and not known. Moreover and for complex eye states, the iterative technique may run in an infinite loop and prevent the process to continually running through time.

Finally, ASM and other existing techniques such as DT or AAM try to detect the eye shape in Cartesian coordinate system. This means a two coordinates (x,y) of each point must be detected. Any particular error in detecting these coordinates affects negatively on the scale, rotation and translation parameters of the detected eye shape and increases the deformations.

All of these challenging issues have motivated us to implement a new technique for eye

shape detection with taking into account the above mentioned problems as detailed in the next sections.

In this chapter, we propose an eye shape detection method based on the representation of the eye structure and eye shape in Log-polar coordinate system [73]. We aim to guarantee a stable performance regardless the change of illumination, lighting conditions, eye state, scale, rotation angle and translation. This is achieved by applying Principal Component Analysis (PCA) [74] on the red, green and blue (RGB) channels of the eye image in Log-polar domain based on two key ideas:

- The elliptical structure of the eye can uniformly be represented by Log-Polar Transform (LPT) regardless the changes of scale and rotation.

Fig. 4.1 Eye and its components.

- The first principal component of RGB channels of an eye image can be used to represent the color variation between skin and sclera, pupil and iris. This representation gives an initial estimation of the eye area against a wide change of lighting conditions.

By transforming the first principal component into Log-polar image domain and using some high level image processing operations, the eye shape can be detected accurately. The results of the proposed method are compared with ASM and DT.

4.2. Introduction

Eye appears at different scales, rotation angles and translations in an image. This weakens the

performance of conventional eye shape extraction methods and increases detection time. These three factors are dependent and influence each other. Suppose that an eye is rotated to an angle θ, a robust eye shape detection technique should deal as different change of scale at this angle θ as possible. The same specification must be considered when the scale factor is fixed and rotation angle is being changed. Therefore to reach up the optimistic solution, the effects of these factors must be addressed and accomplished in parallel. This issue is not related to object shape detection only. Although, it is a common problem for object detection, tracking and recognition techniques.

Let's take a moment and look deeply at an eye image shown in Fig. 4.1. The eye image is in x-y Cartesian coordinate system. Each point on the eye contour, that considered as an exact eye shape point, has two coordinates. From another perspective, this means the detection method must detect approximately the exact two (x,y) coordinates. Obviously, each coordinate affects the point position and changes the shape factors/parameters somehow particularly. This intuitive observation leads us to think about that if these two coordinates are transformed into only one coordinate, the detection operation is significantly and effectively facilitated.

The key idea is that suppose an eye is roughly located at the center of an eye image. The eye image is $360°$scanned by a line that originates from the center and meets the image edge. As long as the eye exists in the image, regardless scale and rotation factors, the line at each scanned angle θ intersects the eye contour in a point. If the scanned pixels that covered by the line at each θ are arranged in a stack, the task of eye shape detection is converted to detect only the intersection coordinate in each row in the stack. Obviously and in order to attain this conversion, polar coordinate system should be used. On the other hand, if the scale factor is small and the eye mass is located closer to the image center, the detection task becomes very hard because the geometrical details will be not distinguishable. Thus, the pixels at the center

should be enlarged as well as the pixels at the image edges should be compressed. Another simpler approach is to consider the pixel size at the image edge as the reference size and modify the other pixels' sizes accordingly. In order to meet all of these particular requirements, we use Logarithmic-polar Transform instead of Polar Transform.

4.3. Log-Polar Transform

Log-polar Transform (LPT) has been applied for solving many problems in image processing such as optical flow[75], active vision systems[76][77], pattern recognition[78], image registration[79], robotic applications[80]…etc. It is a geometrical transformation which resembles the visual information at *retina* [81]. LPT has a high resolution center called *fovea* that surrounded by a periphery. The resolution decreases as the radial distance from *fovea* increases. Therefore, it can be said that human vision system uses LPT to identify corners and assemble the edges between them.

4.3.1. Mapping Strategy

LPT is considered as a modification of Polar coordinate system. The difference is that instead of using the distance ρ as a coordinate of points, logarithmic of ρ ($\log\rho$) is used. Therefore, LPT is a nonlinear sampling transform which converts an image $I(x, y)$ in Cartesian coordinate system, which has width *W*, height *H* and center coordinates at x_c and y_c, into Log-polar image $\acute{I}(\text{Log}\rho, \theta)$ using Eq. (4.1).

$$\acute{I}(\text{Log}\rho, \theta) = \begin{bmatrix} \log(1 + \sqrt{(x - x_c)^2 + (y - y_c)^2}) \\ \arctan \dfrac{y - y_c}{x - x_c} \end{bmatrix} \tag{4.1}$$

The range of θ is $0 \leq \theta \leq 2\pi$ whereas the range of ρ is $0 \leq \rho \leq \rho_{max}$. ρ_{max} represents the radius of a largest disc that can be fitted fully to the image and it can be obtained

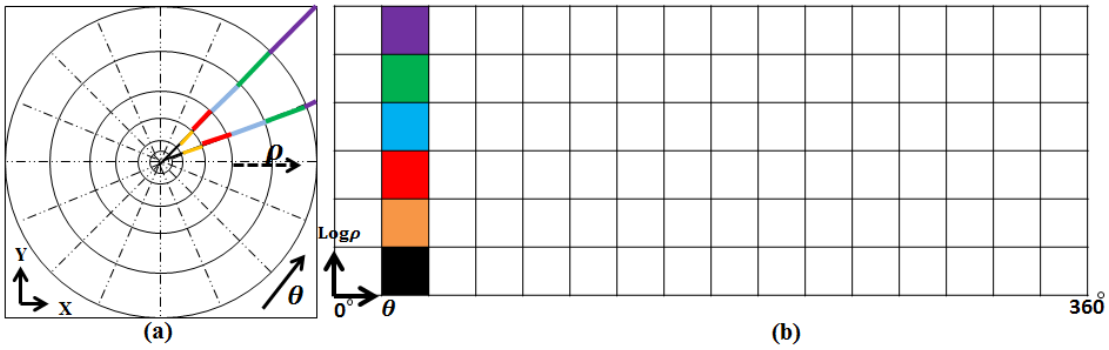


Fig. 4.2 Log-Polar transform. (a) Mapping strategy in Cartesian coordinate system. (b) Log-Polar transformed strategy.

by Eq. (4.2).

$$\rho_{max} = \sqrt{(\frac{W}{2})^2 + (\frac{H}{2})^2} \tag{4.2}$$

An image in Cartesian coordinate system is sampled by a set of circular rings and a set of radial sectors as shown in Fig. 4.2. Suppose that an image $I(x, y)$ with size W x H is Log-polar transformed into an image $\acute{I}(\text{Log}\rho, \theta)$ with size W_ρ x H_θ. The width of Log-polar image W_ρ determines how many rings will be used in the sampling operation whereas the height H_θ indicates the number of radial sectors. Accordingly, the sampling rate of θ is $2\pi/H_\theta$ whereas the sampling rate of ρ is ρ_{max}/W_ρ. Starting ρ at very small value makes the logarithmic function undefined. For that, 1 is added to Eq. (4.1). This small region is called the blind spot and can be resembled to the point that the central optic nerve meets *retina* in the human visual system.

4.3.2. Log-polar Transform properties

- Rotation-invariance: The image in Fig. 4.2(a) is decomposed into two sub-images as shown in Figs. 4.3(a-b). Figure 4.3(a) shows a set of circles that share a common center. This image is log-polar transformed. It can be observed that each circle is converted into a full vertical line. All angular positions of a point on a circle will be mapped into the

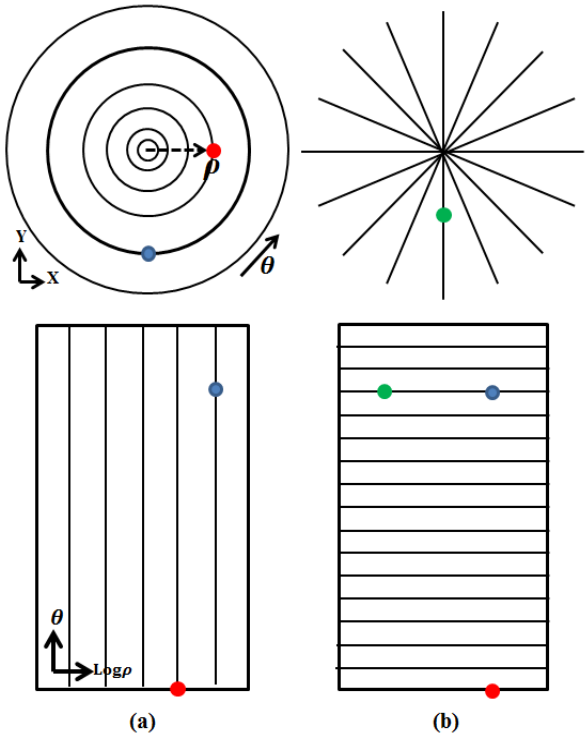


Fig. 4.3 Log-Polar properties. (a) Circle Log-polar representation. (b) Radial line Log-polar representation.

corresponding vertical line. From another point of view, the change of the radial position (moving from circle to another) is transformed into translation along log ρ axis.

- Similarity in translation (conformity): Figure 4.3(b) shows the radial lines are converted into a stack of lines in log-polar image. The position of each line is determined by its rotation angle. It emphasizes the previous property by preserving the angles between radial lines. The radial lines intersect at the center and hence an angular sector between each two lines is created in Cartesian coordinate system. On the other hand, the radial lines are Log-Polar transformed into straight horizontal lines with equal vertical distance. Translating on the radial lines is transformed into the same amount of translation in log-polar image.
- Size/scale invariance: It can be inferred in Fig 4.3(a) that the change of thickness of circles doesn't affect the representation in the log-polar image. They are equally represented by vertical lines and the difference is that they are shifted according to the scale factor. In other

words, the small objects near of the center appear in the same size of the big size distant objects. They only differ in log ρ coordinates.

Other properties of Log-Polar transform can be observed in Fig. 3.4 as well as the above properties.

- A fragment of line that its extension passes the center is represented as a straight -horizontal line in Log-polar domain (red line). On the other hand, the yellow line is shifted in parallel to the red one. This means its extension doesn't pass the center and the line becomes curvy in Log-polar domain. Accordingly, straight lines become curves except those passing the center. The intersections between lines are preserved except those passing the center as illustrated by the pink triangle.
- Corners are intersections between lines/curves. Thus, a corner appears as a vertex between two curvy lines as can be seen for two corners inside the gray rectangle and the green circle. Some other features of corners can be extracted according to corresponding Log-polar

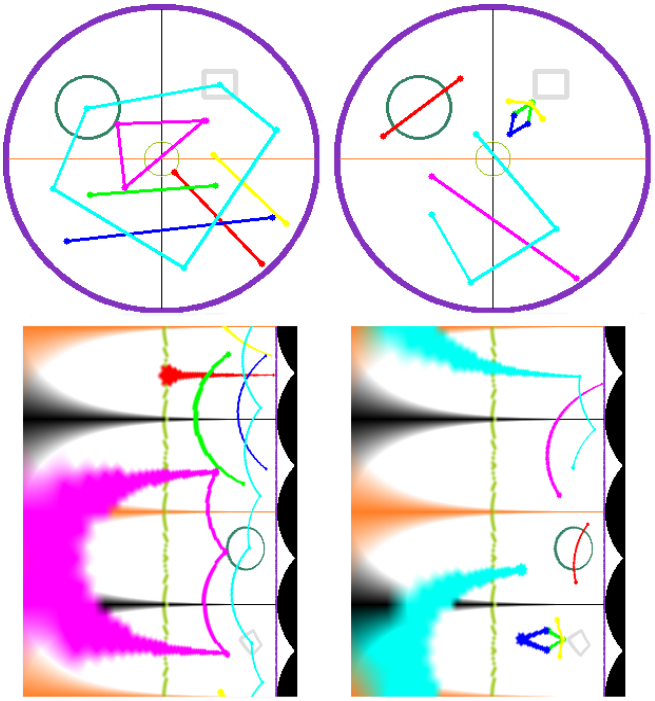


Fig. 4.4 Log-Polar presentations of geometrical shapes in image domain.

representations such as orientation and rotation degree.

- In Cartesian coordinate system, any geometrical shape surrounds the center regardless the distance is fully laid-on along θ axis in Log-Polar domain as the green circle and the bright blue polygon. On the other hand, the closed shapes that don't surround the center are deformed according to the position because their edges are considered as straight lines. Therefore, only circles have (out of the center) approximately the same representation in Log-polar domain.

All above properties can be interpreted mathematically as follows:

Given an image I(x, y)as a reference with corresponding Log-polar image I(ρ, θ). A new image G($\acute{x}, \acute{y}$) is obtained by rotating I(x, y) by a factor δ and scaling it by a factor *s* using Eq. (4.3).

$$\begin{bmatrix} \acute{x} \\ \acute{y} \end{bmatrix} = \begin{bmatrix} s\cos\delta & -s\sin\delta \\ s\sin\delta & s\cos\delta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.3}$$

The corresponding Log-polar image is G($\acute{\rho}, \acute{\theta}$). In Log-polar domain,

$$\acute{\rho} = \log\left(\sqrt{(\acute{x})^2 + (\acute{y})^2}\right) \tag{4.4}$$

This equation can be re-written as follows:

$$\acute{\rho} = \log\left(\sqrt{(s\,\rho\,cos\theta cos\delta - s\,\rho\,sin\theta sin\delta)^2 + (s\,\rho\,cos\theta sin\delta + s\,\rho\,sin\theta cos\delta)^2}\right)$$

$$\acute{\rho} = \log\left(\sqrt{(s\,\rho\cos(\theta+\delta))^2 + (s\,\rho\sin(\theta+\delta))^2}\right) = \log\sqrt{s^2\rho^2} = \log\rho + \log s \tag{4.5}$$

The change of scale in Cartesian coordinate system is transformed into translation in Log-polar domain along Log ρ axis. It can be graphically proven in Fig. 4.4 by observing the representations of two circles; green and purple.

On the other hand,

$$\theta' = \arctan\frac{\acute{y}}{\acute{x}} = \arctan\left(\frac{s\,x\sin\delta + s\,y\cos\delta}{s\,x\cos\delta + s\,y\sin\delta}\right) = \arctan\left(\frac{s\,\rho\cos\theta\sin\delta + s\,\rho\sin\theta\cos\delta}{s\,\rho\cos\theta\cos\delta - s\,\rho\sin\theta\sin\delta}\right)$$

$$\acute{\theta} = \arctan\left(\frac{s\,\rho\sin(\theta+\delta)}{s\,\rho\cos(\theta+\delta)}\right) = \theta + \delta \qquad (4.6)$$

Therefore, the change of rotation in Cartesian coordinate system is transformed into translation in Log-polar domain along θaxis.

4.4. Eye shape properties in Log-polar Domain

Let's take a deep look on an eye image from a geometrical point of view. Figure 4.1 shows an eye images. Eye possesses an elliptical shape approximately. The shape/contour can be divided into two parabolas. The parabolas meet in two points which are the eye corners. One eye corner is usually sharper than the other. The line passes the two corners is the major axis of the ellipse and determines the eye apparent angle. The minor axis can be obtained using the major axis and it determines the eye state; open, nearly closed and in-between. Therefore, many significant features on the eye state can be extracted from the eye shape. Let's try to match these eye shape features with Log-polar properties as follows:

- The eye contour is a closed shape and assumed to be approximately/nearly located at the eye image center. Therefore, corresponding Log-polar representation is a shape fully laid-on along θ axis.
- The Log-polar representation of the eye contour starts and ends at the same angle. If the eye shape is divided into two parabolas, each parabola has a curvy line around the center. Therefore, the corresponding Log-polar representations are two curvy lines as well.
- The orientation of the two eye corners is internal towards the image center. They will have the same Log-polar representation. In addition, as long as the topological structure of the eye contour changes smoothly, only two corners will appear in Log-polar domain between the two curvy lines.

- The line between the two eye corners (major axis) is assumed to go through the center. Therefore, a discontinuity occurs and corresponding Log-polar representation is two horizontal lines releasing from each corner. The rotation angle of the eye can be simply calculated by subtracting the corresponding θ coordinates of the two lines in Log-polar domain.
- Last but not least, the minor axis is also assumed to pass the center. Thus the Log-polar representation is also two horizontal lines but releasing from the vertexes of the two curvy lines. The two vertexes can be utilized with one eye corner to calculate the eye opening degree.

4.5. Eye shape representations in Log-polar domain

Figure 4.5 shows a set of eye shapes represented by N=16 points and their corresponding

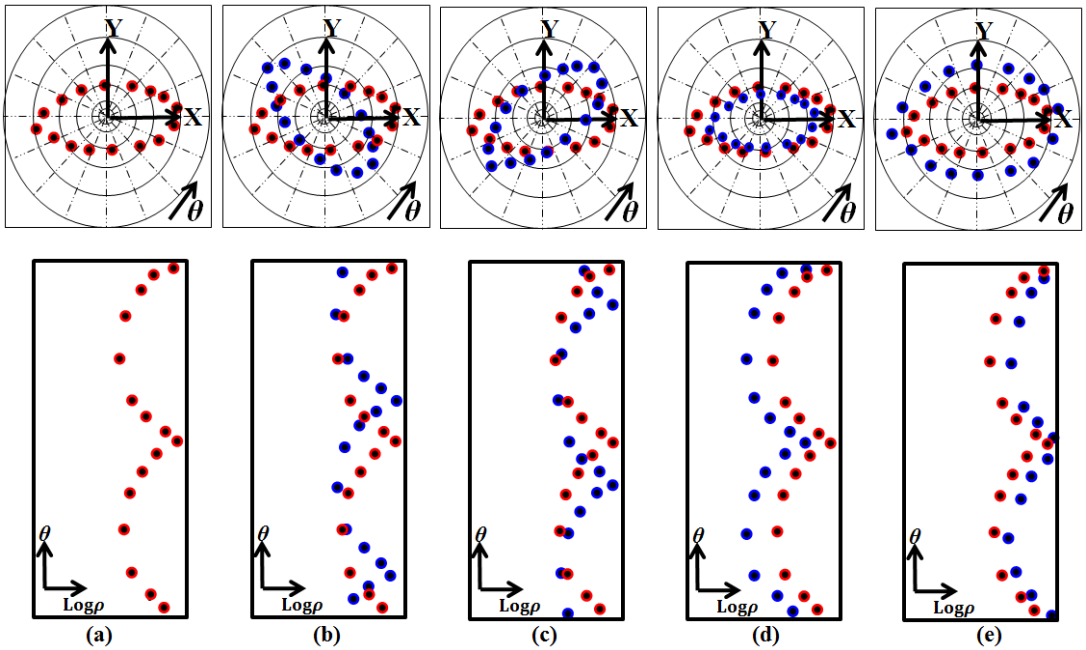


Fig. 4.5 Eye shape representations in Log-Polar domain. (a) Reference state $\theta = 0°, S = 1$. (b) Rotated by $\delta = +45°$. (c) Rotated by $\delta = -45°$. (d) Scaled by $S = 0.8$. (e) Scaled by $S = 1.2°$.

representations in Log-polar domain. Figure 4.5(a) shows the eye shape as a reference with a rotation angle $\theta = 0°$ and a scale factor $s = 1$. The shape can be said symmetric. Therefore, the point distribution forms ideally two similar curves in Log-polar domain as well as the two eye corners appear at the same log ρ coordinates.

The shape is rotated by angle $\delta = \pm 45°$ in Figs 4.5(b-c); respectively. The corresponding Log-polar representations are preserved and only shifted along θ axis accordingly; up and down. Figures 4.5(d-e) show the eye shape is scaled to $s = 0.8$ and $s = 1.2$. The corresponding Log-polar representations are shifted along log ρ axis accordingly; left and right.

Another critical issue to be mentioned is translation. The eye is not guaranteed to be located at the image center accurately and some translations may occur. The necessary and sufficient condition is that the eye contour must surround the image center regardless shifting effects. The shifting effects can be considered as a change of symmetry in Log-Polar domain.

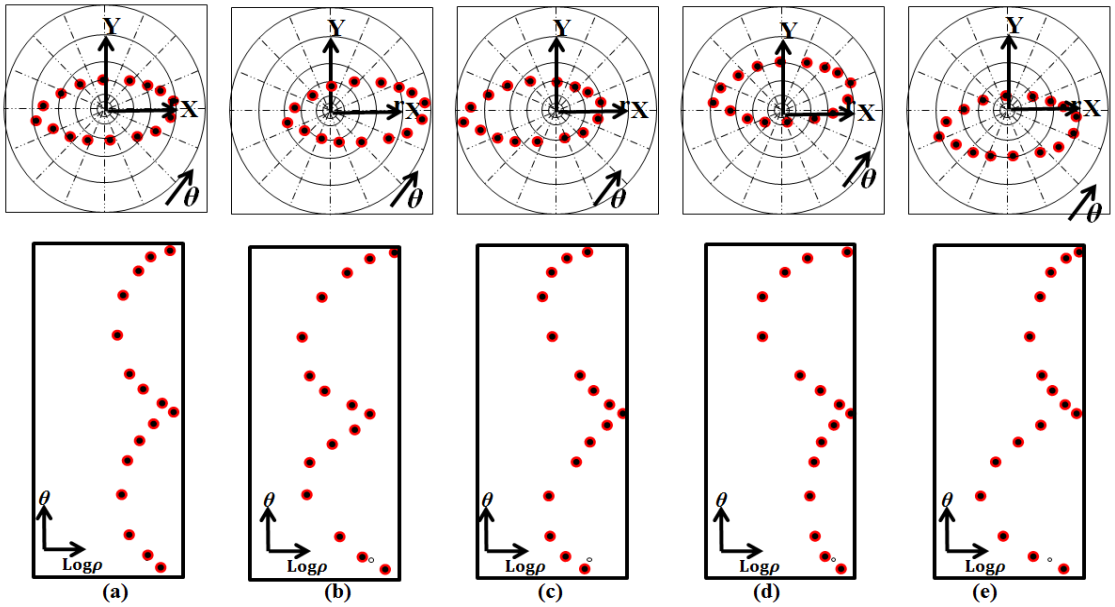


Fig. 4.6 Eye shape representations in Log-Polar domain for changing of translation. (a) Reference state (shape at the center). (b) Shifted along X axis positively. (c) Shifted along X axis negatively. (d) Shifted along Y axis positively. (e) Shifted along Y axis negatively.

Figure 4.6 demonstrates four cases of the eye shape that shifted along x and y axes; respectively. The corresponding Log-polar representations indicate that the eye shape is completely represented in Log-polar domain and changing in translation only occurs according to particular distance of each point from the image center.

Based on the previous discussion, Log-polar domain is the best domain to implement any eye shape detection technique.

4.6. Modeling eye shape in Log-polar domain

4.6.1. Modeling strategy

The eye shape in Log-polar domain is represented by two curvy point distributions along θ axis. The distribution can be analyzed by a mathematical function. On the other hand, PCA has achieved a very flexible and robust performance to analyze the change of the eye shape in Cartesian coordinate system [82]. Therefore, we employ PCA to model the eye shape in Log-polar domain. Due to the special and significant properties of Log-polar transform, there

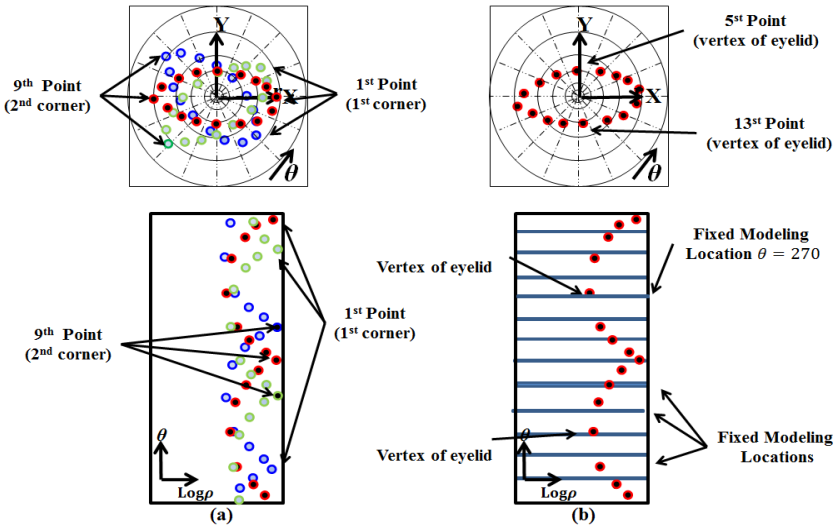


Fig. 4.7 Modeling strategy. (a) Fixed point identification according to the local structure on the eye contour. (b) Fixed modeling location strategy.

are two approaches of modeling as follows:

- The first one is to give each eye shape point an identity of a specific local part/feature on the eye contour. In other words and similar to ASM, 1^{st} and 9^{th} points indicate the locations of two eye corners as well as 5^{th} and 13^{th} points represent the middle points on upper and lower eyelids; respectively. The 5^{th} and 13^{th} represent the vertex locations of the two curvy distributions in Log-polar domain. This strategy provides immediate information on the eye feature locations once the eye shape is being detected. On the other hand, a massive difference in point distribution between the eye shapes will occur. For example as in Fig. 4.7(a), if the shape has $\delta = +45$ rotation degree, the 1^{st} eye corner appears at $\theta = 320°$ whereas it appears at $\theta = 40°$ if the shape has $\delta = -45$ rotation degree. Consequently, the alignment operation between eye shapes is impossible to be achieved. Moreover, this strategy is not compatible with the detection method that will be explained later.
- The second approach is to consider fixed modeling locations regardless the corresponding local representations on the eye contour. For example as in Fig. 4.7(b), analyzing the change of the eye contour at $\theta = 270°$. This strategy is more simple and reasonable. Each fixed modeling location illustrates the possible and particular change of the eye contour. Therefore, it is guaranteed that principal components will demonstrate very significant patterns of eye shape variation with increasing the capability of generalization. Moreover, the alignment operation will be smoothly achieved. The only negative aspect of this strategy is that the identity of the eye shape points must be searched on after detecting the eye shape. Based on eye shape properties in Log-polar domain, this issue can be neglected because some key points such as eye corners/vertex points can be distinguished easily out of the other points. Accordingly and as long as the number of eye shape points is fixed, *N*=16, the

other points can be easily identified as well.

4.6.2. Mathematical Model

As mentioned previously, PCA is employed for achieving the modeling task. The mathematical procedures are same as of that explained in *section* 3.2.4. Thus, a brief explanation with referring to the key difference points is only highlighted in this section.

We collect M eye images to represent, as usual, different lighting conditions, scales translations and rotation angles. The eye images are then Log-polar transformed as shown simply in Figs. 4.8 (a-b). The eye shape is described by N=16 points that are manually labeled at 16 fixed modeling locations along θ axis. Figure 4.8(c) shows M eye shapes. It can be observed that at each modeling location, the point distribution changes only along $\log\rho$ axis. The eye shape of i^{th} log-polar eye image is represented by a vector X_i and a shape point has two coordinates $\theta_{i1}, \log\rho_{i1}$ as follows:

$$X_i = \left(\theta_{i1}, \theta_{i2}, \ldots, \theta_{ij}, \ldots, \theta_{iN}, \log\rho_{i1}, \log\rho_{i2}, \ldots \log\rho_{ij}, \ldots, \log\rho_{iN}\right)^T \quad (4.7)$$

The alignment operation, as explained in *section 3.2.2.2*, is then applied on the Log-polar eye shapes. Figure 4.8(d) shows the eye shapes after being aligned to the mean Log-Polar eye

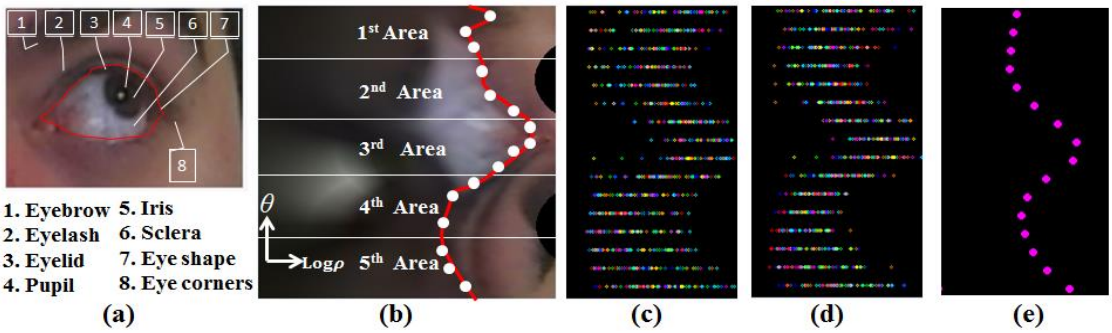


Fig. 4.8 Eye shape modeling steps. (a) Eye image in Cartesian coordinate system. (b) Eye image in Log-Polar domain and eye shape points manually labeled according to the fixed modeling location strategy. (c) Eye shapes before alignment. (d) Aligned eye shapes. (e) Mean Log-polar eye shape.

shape $\overline{X}$ in Fig. 4.8(e).

As long as the modeling locations are known in advance at fixed θ coordinates, a Log-polar eye shape vector is broken down to contain logρ coordinates only.

$$\acute{X}_i = \left(\log\rho_{i1}, \log\rho_{i2}, \ldots \log\rho_{ij}, .., \log\rho_{iN}\right)^T \quad (4.8)$$

The covariance matrix is then calculated as well as the corresponding eigenvectors and eigenvalues. The eigenvectors are arranged in a matrix Ω according to their eigenvalues in descending order. As known, the first *K* eigenvectors describe the direction of the most variation in Log-polar eye shapes. These few eigenvectors are combined into a matrix $\acute{\Omega}$. Hence, the eye shape can be approximately represented as a linear combination of the eigenvectors in matrix $\acute{\Omega}$ using the model equation as follows:

$$X_i = \bar{X} + \acute{\Omega}\mathbb{B}_i \quad (4.9)$$

where $\mathbb{B}_i = \mathbb{b}_{i1}, \mathbb{b}_{i2}, \mathbb{b}_{i3}, \ldots \mathbb{b}_{ij}, \ldots, \mathbb{b}_{ik}$ is the projection vector of i^{th} shape and $\mathbb{b}_{ij}$ is the projection on j^{th} eigenvector which represent the contribution amount of j^{th} eigenvector in reconstructing X_i .

4.6.3. Testing of eye shape model

The dynamical behavior of the eye shape in Log-polar domain can be divided into five areas as highlighted in Fig. 4.8(b). In this study, we consider the rotation angle ranges between $\theta = \pm 45°$ which matches the normal motion of a person's face setting down on a chair and looking at the camera. The intermediate area contains 4 eye shape points whereas the others contain 3 points for each. In the intermediate area, one of the four points will represent the eye corner according to the rotation angle. The second corner will appear in the 1st or 5th area. In 2nd and 4th areas, one point will represent the vertex of the curvy line (eyelid).

Figure 4.9(a) shows the behavior of the first four principal components in the range of

corresponding eigenvalue $\pm 3\sqrt{\lambda_i}$. The intermediate shape in each row is the mean Log-polar eye shape which has a zero projection vector. Figure 4.9(b) demonstrates the motion amount that occurs at each point in the studied range. The direction of movement can be discriminated by the green points which are the ending points at $+3\sqrt{\lambda_i}$.

- The first principal component has the largest eigenvalue and describes the most significant information. It represents general relationship of change of each point with respect to the others. It is necessary to observe that the eye corner in the intermediate area is represented by the second point continually. Accordingly, the first component illustrates the variation along $\log\rho$ axis.
- The second component describes the relationship between the two eye corners when the second corner appears in the first area. The θ coordinate of the first corner changes from

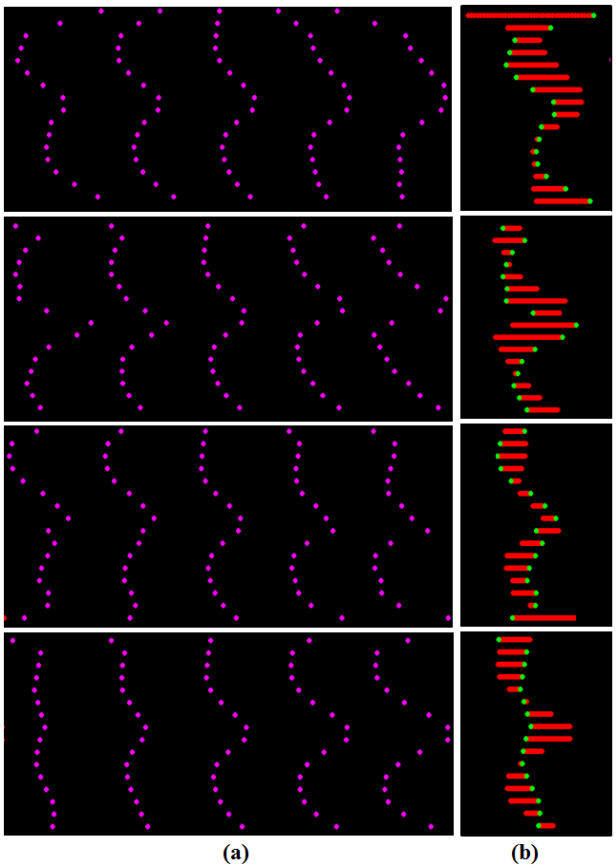


Fig. 4.9 Eye shape model testing. (a) First four eigenvectors. (b) Motion behavior.

the first point in the intermediate area to the third one. Therefore, besides of representing the change along $\log\rho$ axis, the second component demonstrates the change along θ axis as well. This can be proven by the corresponding motion profile in Fig. 4.9(b).

- The third component possesses the same features of the second one with one difference that the second corner appears in the fifth area.
- The fourth component can be considered as the representation of each point's change with less respect to the others' changes. It tends to line up the points at the ending of the studied range. Therefore, it can be said approximately that the motion profile is symmetric vertically.

It should be kept in mind that the eigenvalue decreases nonlinearly as the index of the principal component increases. Therefore, the represented variation between points by each component decreases accordingly as can be compared in Fig. 4.9(b).

Based on the above demonstration, the eye shape model is very robust and signifies the most information needed to describe the dynamical behavior of Log-polar eye shape points. Consequently, the model is expected to refine point deformations efficiently and effectively.

4.7. Eye texture in Log-polar domain

Figure 4.10 shows a set of eye images and corresponding representations in Log-polar domain. The eye images demonstrate different rotation angles, eye states and scale factors. The eye in Log-polar domain can be classified horizontally into two areas; eyeball and flesh. The intersection points between these two areas form the eye contour. On the other hand, the eye image can be classified vertically into the same five areas that mentioned in the previous section. In addition and as long as the eye corners are internally pointed to the image center in Cartesian coordinate system, they have the same local structure in Log-polar domain. Thus, the eye contour has a very smooth profile.

It is very important to highlight that the eye contour exists along θ axis regardless the rotation angle and scale factor. In order to investigate the LPT robustness in representing the eye with respect to the change of translation, we shift the center location instead of shifting the eye image in Cartesian coordinate system. These two actions are equivalent but the first one is simpler to be understood.

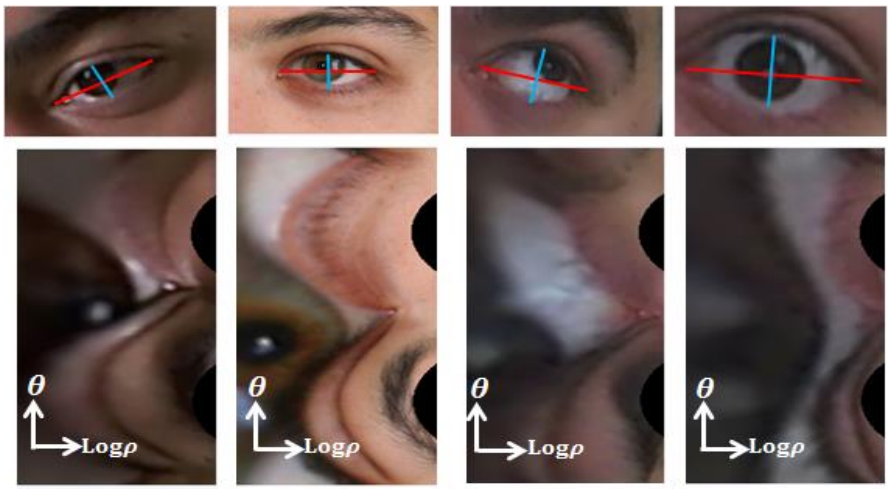


Fig. 4.10 Eye in Log-polar image domain.

Figure 4.11 demonstrates four cases of different positions of the image center (red point). This figure matches the concept highlighted in Fig. 4.6. The green circle illustrates the studying area to be transformed into Log-polar domain. It can be emphasized again that the change of translation in Cartesian coordinate system doesn't affect the existence of the eye contour along θ axis as long as the center is being in the eyeball area and the studying area houses completely the eyeball.

4.8. Modeling eye texture in Log-polar domain

As long as the eye contour exists along θ axis, the task of detecting an eye shape point is simplified to determine and search on $\log\rho$ coordinates. Therefore, the key idea is to study/analyze the pattern of pixel distribution along each modeling location. We then try to create a model to describe the patterns along any modeling location.

4.8.1. Eye color interpretation using PCA

Eye possesses a very important property that the skin color changes gradually because of the

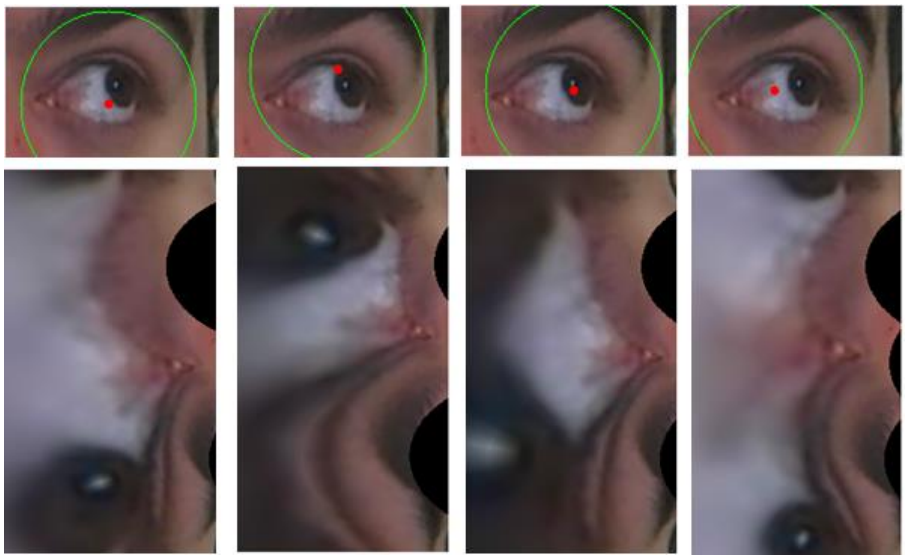


Fig. 4.11 Translation of eye in Log-polar image domain.

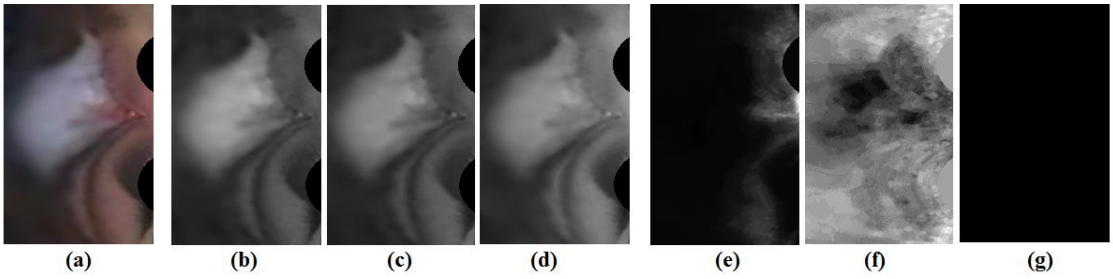


Fig. 4.12 Eye image analysis based color variance. (a) Log-polar eye image. (b) Red color channel. (c) Green color channel. (d) Blue color channel. (e) First principal component. (f) Second principal component. (g) Third principal component.

topological structure. On the other hand, eye components, sclera, iris and pupil, can be represented approximately using fixed colors; white and black. These two colors are obtained in RGB color system by the following two values respectively; (255,255,255) and (0,0,0). Consequently, the variance between eye color channels in the eyeball area is assumed to be fixed and very small because of high correlation whereas the variance in skin color changes gradually. The best efficient method to interpret the variance and to discover the meaningful information carried by each pixel is to use PCA. The aim is to enhance and maximize the contrast between color channels by combining like areas together. Hence, the area of eyeball can be readily discriminated in the first principal component.

Figures 4.12(a-c) show Log-Polar eye image and corresponding decomposed three color channels; Red, Green and Blue; respectively. PCA is applied on the color channel images in the same style explained in *section* 3.7.1.2. Each image is converted into a vector by concatenating each row with the previous.

$$I_p = \left(i_{p,0,0}, i_{p,0,1}, \dots\ i_{p,0,n}, i_{p,1,0} \dots\ i_{p,j,k}, \dots\ i_{p,n,m}\right)^T \tag{4.10}$$

where $p = 1,2,3$ refers to R, G and B color channels; respectively, $i_{p,j,k}$ is a pixel value at j^{th}row and k^{th}column, *n* and *m* are the width and height of the eye image; respectively and T denotes the vector transpose operation.

Consider a matrix $A = [(I_1 - \bar{I})\ (I_2 - \bar{I})\ (I_3 - \bar{I})] = [a_1\ a_2\ a_3]$ where $\bar{I} = (I_1 + I_2 +$

$I_3)/3$ is the mean channel vector which have the same size of $I_p$and whose entries are the average of RGB pixel values. A covariance matrix is then calculated to describe how each pixel value (color) varies with respect to the others along the three channels.

The eigenvectors of the covariance matrix are computed and shown in Figs 4.12(e-g); respectively. The first eigenvector represents the contrast between RGB channels and it is called Log-polar Eigeneye Image (LPEI). The eye area has low pixel values (black) whereas the skin area changes in grayscale. Based on this fact, the eye contour can be initially estimated at the boundary between these two areas. The estimation does not represent the exact eye shape because of noise. The noise may occur due to the existence of red nerves in the sclera, light reflectance and eyelids. On the other hand, the initial estimation can be easily reformed based on the eye shape properties in Log-polar domain. Figure 4.12(f) shows the second eigenvector which is considered to represent noise in the image. Fig. 4.12(g) shows the last eigenvector which is almost black because its eigenvalue is very small.

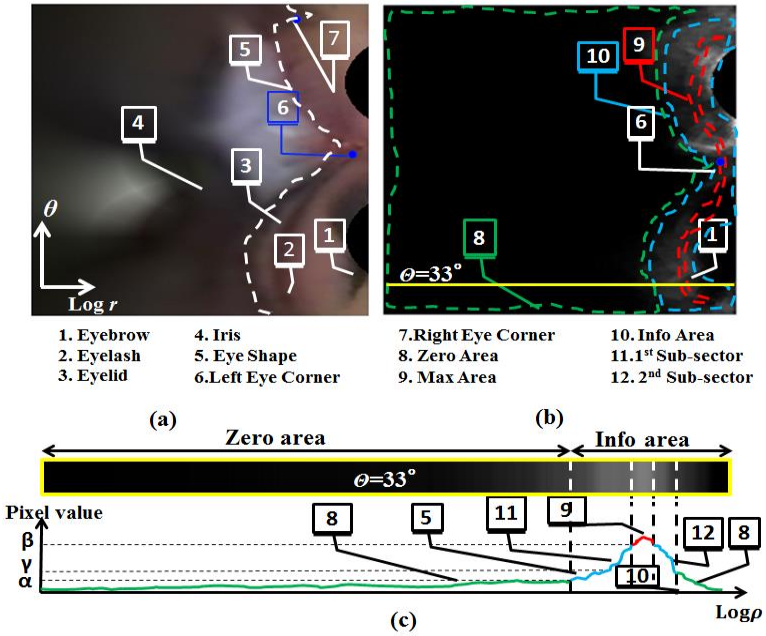


Fig. 4.13 Eye image in Log-polar domain. (a) Log-polar representation eye image. (b) Corresponding eigenvector image of eye (LPEI). (c) Uniform profile of pixel values at θ=33° in (LPEI).

4.8.2. Texture Modeling

LPEI shown in Fig. 4.13(b) can be classified into three areas in each row (or at each θ) based on the eye topological structure as follows:

- Zero area: this area represents the eyeball and it may contain iris, pupil and sclera depending on the eye state and gaze point. The contained pixel values are very low and smaller than a threshold α.
- Information area: this area represents skin around the eye. The corresponding pixel values are large, change in grayscale gradually and bounded between α and 255.
- Max area: this area is contained in the information area and the pixel values are larger than threshold β. It divides the information area into two sub-sectors:
  a) The first sub-sector has an ascending order of pixel values bounded from α to β. It lies from the beginning of the information area until the maximum area.

b) The second sub-sector has a descending order of pixel values bounded from β to α. It lies from the maximum area until the end of the information area. The existence of this sub-sector depends of the eye state and modeling locations. For example, in the area just below the lower eyelid ($\theta = \pi \sim 2\pi$), the second subsector may not exist according to the prominence of the eye. On the other hand, it mostly appears near the upper eyelid.

Figure 4.13(c) illustrates this classification by enlarging the row at $\theta = 33°$ that is covered by a yellow line in Fig. 4.13(b). The corresponding pixel profile is depicted below.

By the end of this discussion, the pixel distribution in each row can be represented by a uniform profile regardless the exact pixel values. Therefore the pattern starts with low pixel values (zero area), takes an ascending order gradually, reaches max area and then descends gradually or stays in the max area.

4.8.3. Eye contour location

The location of the eye contour is the boundary between information and zero areas. More accurately, the eye contour can be detected at the beginning of the first subsector. The first

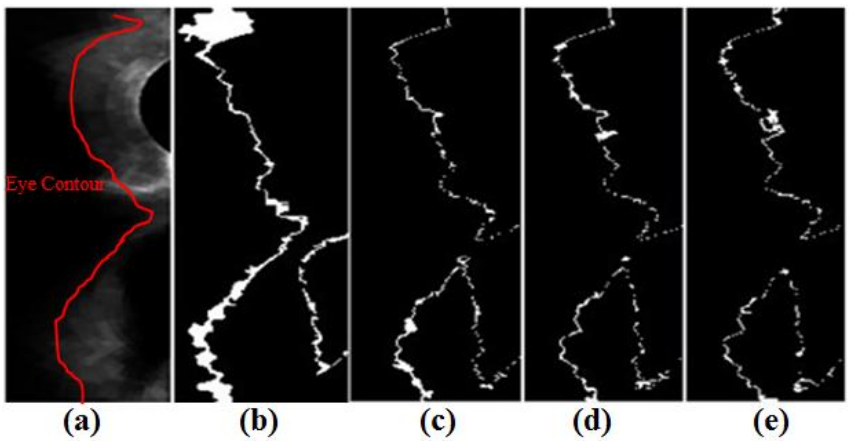


Fig. 4.14 Eye contour in LPEI. (a) LPEI. (b) Curve having 3 intensity values (pixel values). (c) Curve highlights locations of 5 pixel values in LPEI. (d) Curve highlights locations of 6 pixel values. (e) Curve highlights locations of 7 pixel values.

subsector has an ascending order of the pixel value. Figures 4.14 show an example on the ascending order in the first subsector. Four binary images are presented and the white curves illustrate the locations of arbitrary pixel values 3, 5, 6 and 7 in the first subsector; respectively. These curves can be considered as the eye contour in the testing image. The pixel values 3, 5, 6 and 7 are randomly chosen and they are not guaranteed to exist in each LPEI. On the other hand, the order of pixel values in the first subsector is fixed. Therefore, applying cross-correlation between a LPEI and a short template of a few pixel values that arranged in an ascending order can be used to highlight the proper location of the eye contour in a LPEI without discontinuity.

4.9. Cross correlation

Cross correlation is considered as the essence of many image processing techniques such as pattern recognition [83][84], image registration [85], object detection [86],…,etc. It is a very simple and effective operation to extract information from the images. The simplest strategy is to sum the square of the differences of pixel values between a template T whose size is $2a$ and the 1D segment of an image I that centered at x.

$$Co(x) = \sum_{i=-a}^{a} \big(T(i) - I(x+i)\big)^2 \tag{4.11}$$

The segment is compared at each x over the image. The result is an image in gray pixel values with a noticeable peak. The peak represents the location of the best correlation between the image and the template. Equation 4.11 is sensitive to the change of lighting conditions. In order to increase the robustness of correlation, normalization is applied and equation is modified as follows:

$$Co(x,y) = \frac{\sum_{i=-N}^{N}\sum_{j=-M}^{M} T(i,j)I(x+i,y+j)}{\sqrt{\sum_{i=-N}^{N}\sum_{j=-M}^{M} T(i,j)^2}\sqrt{\sum_{i=-N}^{N}\sum_{j=-M}^{M} I(x+i,y+j)^2}} \quad (4.12)$$

This equation must be calculated for all possible size of the template. Otherwise, the searched object must be in the same size exactly of that in the template. The template is usually scanned pixel by pixel over the image by sliding it to visit all positions in the two directions. This process is a cost operation for real time applications. In addition, it will be more highly computational process and complicated when it is computed for all possible sizes. The consumed time can be reduced significantly when the cross calculation is done in Fourier transform [87].

4.9.1. Fast Fourier Transform and Phase correlation

Fourier Transform considers an image as a function and decomposes it into a set of basic functions which are a series of sine and cosine waves [88]. These functions are orthogonal and differ from each other by two parameters; phase and frequency. Each pixel in the image is converted to represent a particular frequency in Fourier domain. Hence, the image is represented as a point in a vector space. This point possesses projections in the space and they can be obtained by calculating the inner product between the given function (image) and the basic functions. The basic functions are expressed in two equivalent formats as follows:

$$e^{-i2\pi\omega x} = \cos(2\pi\omega x) + i\sin(2\pi\omega x) \quad (4.13)$$

4.9.2. Fourier Transform Properties

- The transform has real and imaginary terms and the exponential format allows representing the two terms.
- Any two basic functions of different frequencies are orthogonal and their inner product is

zero.

- A given function *f*(*x*), its Fourier transform F(ω) is given by Eq. (4.14).

$$F(\omega) = \int f(x) e^{-i2\pi\omega x} dx \tag{4.14}$$

Obviously and in general,F(ω) consists of two components; real a(ω)and imaginary b(ω). F(ω) can be represented in polar form to have a normal amplitude component |F(w)| that called the Fourier spectrum as well as phase angle component $\Phi(\omega)$.

- In the image domain, an image is described as a discrete function $I(x, y)$. Consequently, the Discrete Fourier Transform (DFT) exists and discrete frequencies are used to describe the spatial image domain. The basic functions have a discrete format represented by the exponential term in Eq. (4.15).

$$F(j,k) = \frac{1}{NM} \sum_{x=1}^{M} \sum_{y=1}^{N} I(x,y)\, e^{-i2\pi \frac{j.x}{M} \frac{k.y}{N}} \tag{4.15}$$

The frequencies of basic functions (sine and cosine) increase as stepping far from the origin of the Fourier domain. The origin contains the DC-component which represents the average of the pixel values of the image.

- The amplitude of the basic functions of an image that consists of zero pixels and a gray area is not affected by the (*x*,*y*) translations of this area in the spatial image domain; this is called shift therom.

- Two images with same size are represented in Frequency domain with the same amplitude.

4.9.3. Phase Correlation in Log-polar domain

Based on the previous explanation, the images have the same magnitude because the magnitude of a complex exponential function is the radius of a unit circle. Hence, a template *T* and a LPEI I($\log\rho, \theta$) must be in the same size. This can be achieved by housing the template in a black

image $\mathbb{T}$ (zero pixel values) whose size is same. Figure 4.15(a) shows a template T that consists of six pixels in gradually change of grayscale. The template is located at the top-left corner of the template image $\mathbb{T}$. The template can be located at any position in the image due to the translation property of the Fourier transform.

The normalized cross power spectrum CPS is defined as follow:

$$CPS(u,v) = \frac{F_I(u,v).F_{\mathbb{T}}^*(u,v)}{|F_I(u,v).F_{\mathbb{T}}^*(u,v)|} = e^{-j(ut_{\log\rho}+vt_\theta)} \tag{4.16}$$

where u and v are pixel coordinates in Fourier domain, F_I is Fourier transform of image I, $F_{\mathbb{T}}^*$ is the complex conjugate of Fourier transform of the template image and, $t_{\log\rho}$ and t_θ are the difference of displacement between the two images along $\log\rho$ and θ axes, respectively. By taking the inverse Fourier transform $\mathcal{F}^{-1}$of the cross power spectrum, a Dirac pulse δ is generated and located at the offset of the first image relative to the second one.

$$\mathcal{F}^{-1}\{CPS(u,v)\} = \delta\left(u + t_{\log\rho}, v + t_\theta\right) \tag{4.17}$$

In order to increase the robustness of cross correlation, the template must be chosen properly to satisfy the following conditions:

- The template must contain few pixels to make the calculation of correlation not an expensive

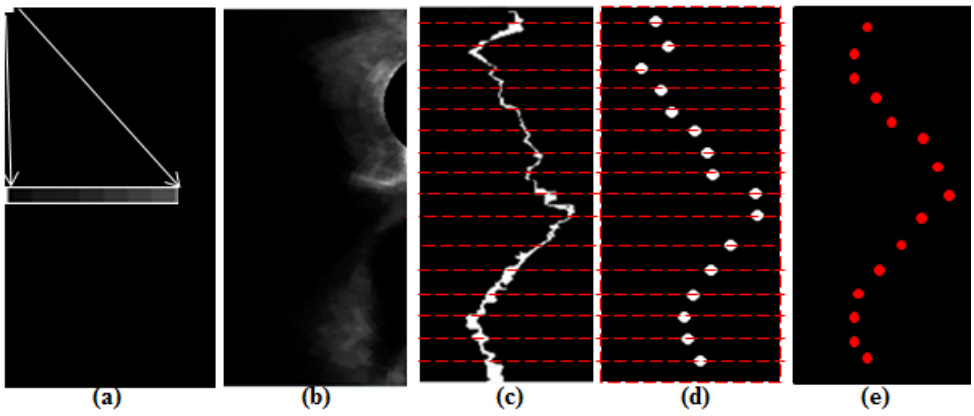


Fig. 4.15 Eye shape detection. (a) Template image. (b) LPEI. (c) Output of cross-correlation (extraction operation). (d) eye shape point extraction. (e) Refined eye shape based eye model (detected shape).

cost operation.

- The template pixel values γ must be in ascending order of the range $\propto > \gamma \ll \beta$ to match the order at the beginning of the first sub-sector.

However, the pixel values at the beginning of the first sub-sector may change according to lighting conditions and skin color with maintaining the same ascending order. To break the limitation of using one template, we can select a set of templates to represent numerous lighting conditions.

By using the template image $\mathbb{T}$, Fig. 4.15(c) shows the result of correlation applied on the LPEI in Fig. 4.15(b). The resultant curve significantly represents the eye contour without any discontinuity. It can be observed that the curve thickness changes in some areas. This indicates that the pixel values of LPEI satisfy the order of pixel values in the template at a few sequence locations in each row.

On the other hand, the satisfaction depends on the intensity range of the pixel values in the template. Therefore, the thickness is terminated and does not expand to highlight the whole first subsector.

4.10. Eye shape extraction

The obtained curve by cross-correlation represents the eye contour in the LPEI. Therefore, the $\log \rho$ coordinates of the eye contour can be obtained at any θ. The modeling locations of the eye shape are known at fixed θ coordinates as depicted in Fig. 4.15(c). Thus, eye shape points can be formed by matching θ coordinates with corresponding $\log \rho$ coordinates as illustrated in Fig 4.15(d).

4.11. Eye shape correction

The extracted eye shape points may carry some deformations in $\log \rho$ coordinates. The deformations can be easily refined by matching the point distribution according to the Log-polar mapping of the eye shape. This distribution is described by the eye shape model as explained in *section* 4.5. The same strategy of ASM correction based PCA eye model, *section* 3.2, is followed here.

Once an eye shape X_{ex} is extracted as shown by white points in Fig. 4.15(d), the proposed method calculates its decomposition vector B_{ex} using Eq. (4.18).

$$\mathbb{B}_{ex} = \acute{\Omega}^{-1}(X_{ex} - \bar{X}) \tag{4.18}$$

According to Gaussian distribution, the allowed variation of the eye shape points along i^{th} eigenvector must be in the range of $\pm 3\sqrt{\lambda_i}$. Therefore, particular elements in $\mathbb{B}_{ex}$ are checked and the outrange elements are shifted to the limit of the corresponding allowable range. The reformed eye shape is then generated by Eq. (4.9) using the corrected decomposition vector as shown in Fig. 4.15(e). Figure 4.16 shows some eye shapes of different eye images in white points. The shapes are obtained by the extraction operation and carry some deformations. The deformations are refined using the trained eye shape model as depicted by corresponding red

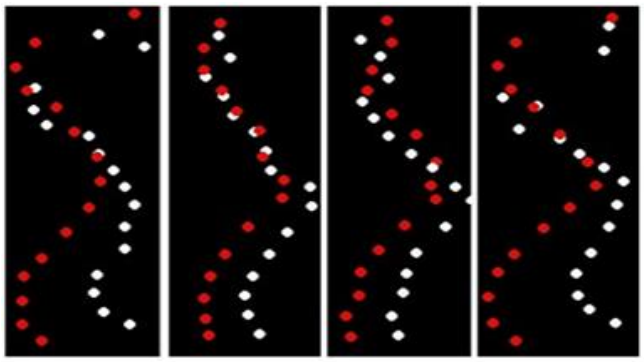


Fig. 4.16 Eye shape refinement, white shapes are obtained by cross-correlation and corresponding red shapes are the corrected shapes by the eye model.

points.

Finally, we summarize the proposed method. The eye image is first transformed from Cartesian coordinate system to Log-polar image domain. The Log-polar image is decomposed into three images representing RGB color channels. PCA is then applied on the color channels to obtain LPEI. Cross-correlation is utilized to filter out the eye contour using a template image and eye shape points are extracted. Finally, the eye shape model is employed to refine the deformations occurred in $\log \rho$ coordinates.

4.12. Results and discussion

4.12.1. Setup

Besides the database used in ASM that locally collected for different scale and rotation angle, a new database is needed to enrich the eye images in terms of different lighting conditions and skin color. Therefore, UBIRIS database is used [89][90]. The contained images were captured on non-constrained conditions under different lighting conditions in two sessions. Figure 4.17(a) shows some eye images in UBIRIS database as well as some of those in local database. The eye images in both databases are resized into a fixed size 150x100 for width which represents x pixel coordinates and for height which represents y pixel coordinates; respectively. The images

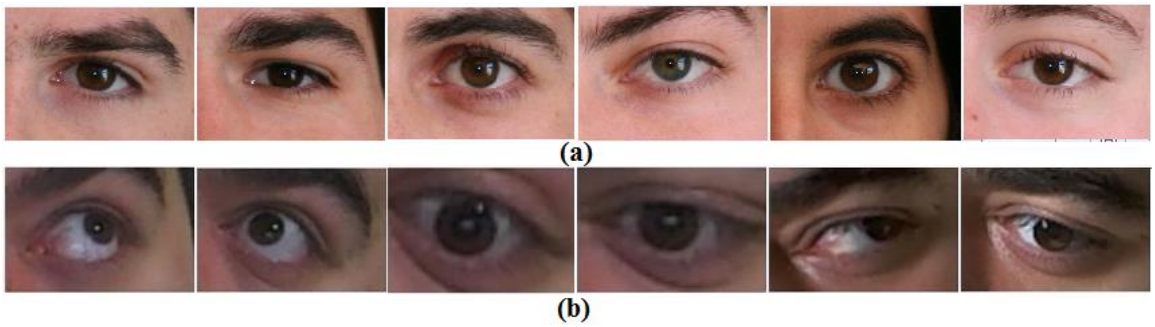


Fig. 4.17 Eye images taken from UBIRIS database [90]. (b) Eye images taken from the local database.

are then Log-polar transformed in the size of 256x360 for width which represents $\log \rho$ pixel coordinates and for height which represents θ pixel coordinates; respectively. One can easily notice that every angle degree is described by a row in the Log-polar image.

The eye shape of each image is described by N=16 points that manually labeled (exact eye shape). On the other hand, the same number of points is used to label the eye shape over the Log-polar images. Therefore, we have N=16 modeling locations at fixed θ coordinates. The θ sampling rate is $\frac{360}{16} = 22°$and we started from 5°. This sampling rate is fixed and used in eye shape detection stage as illustrated in Figs. 4.8(b) and 4.15(d).

Besides the proposed method, improved ASM explained in the previous chapter is used to evaluate the performance as well as Deformable Template (DT) is also involved in this comparison process. Because DT is classified as a method to represent the eye contour by a template as clarified in the first chapter, the template is sampled into 16 points after the detection operation is terminated. The time consumed by this procedure (sampling process) is not considered in the evaluation. The environment of visual studio C++ 2008 and CPU quad-core processor 2.4 GHz are used for implementation and evaluation.

The accuracy of the eye shape X_d detected by any method is calculated by measuring how much it correlates the corresponding manually labeled eye shape X_l using Eq. (4.19).

$$Acc = \frac{X_d . X_l}{|X_d||X_l|} * 100 \tag{4.19}$$

In the proposed method, two templates $T_1$and $T_2$are used to describe the ascending order of pixel values in the LPEI. Therefore, the cross-correlation will be applied for each template as well as the sampling and correction operations. Consequently, two eye shapes are obtained for each testing image accordingly. The corresponding decomposition vectors B_{T1} and B_{T2} of these two eye shapes are calculated using Eq. (4.19). The better shape that best represents the

exact eye shape in the testing image is determined by Eq. (4.20).

In order to determine the template pixel values and to make the proposed method more robust to deal with lighting conditions, 30 images have been randomly chosen from database. We manipulated the brightness of these images by changing the corresponding Y component in YCbCr color space and then observed the change of pixel values in the corresponding LPEI. Based on this test, we found the equivalent stable values of $[\alpha, \beta]$= [8,150].

$$B = Max({B_{T1}}^T B_{T1}, {B_{T2}}^T B_{T2}) \tag{4.20}$$

The amount of variance σ_i expressed by an i^{th} eigenvector can be obtained using its corresponding eigenvalue λ_i as follows:

$$\sigma_i\% = \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} * 100 \tag{4.21}$$

4.12.2. Results

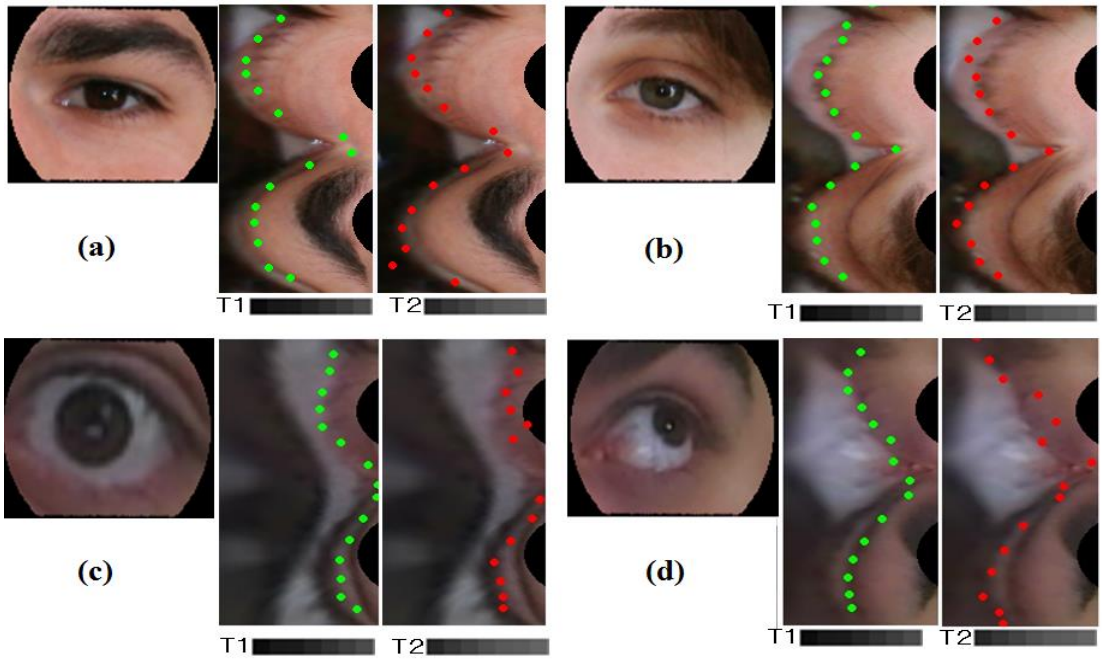


Fig. 4.18 Eye shape detection using the first (green) and second (red) templates. (a) Eye in normal lighting conditions. (b) Different skin color and green iris. (c) Different scale and eye state. (d) Different rotation.

Figure 4.18 shows some testing eye images and corresponding eye shapes detected using the first and second templates, green (T_1) and red (T_2); respectively. The better shape that best matches the eye contour in each case is determined using Eq. (4.20). The second template gives better match when the luminance is high as shown in Fig. 4.18(a-b) because its pixel values are larger than the first template. In contrast, the first template outperforms the second one when the luminance is low as can be observed in Fig. 4.18(c-d). On the other hand, Fig. 4.19 demonstrates the robustness of the proposed method against the change of scale, rotation and eye states as well.

Figure 4.19 shows the relationship between the accuracy and detection time. The performance of each method has been evaluated using the testing images. The figure demonstrates the detection strategy of each method. Each curve has some critical points highlighted by black square marks. The beginning point represents the accuracy of the initial eye shape estimation whereas the ending point indicates the accuracy of the detected shape.

In DT, the curve starts after around 63 msec because the algorithm tries to find roughly the center of pupil in the testing image. Based on the pupil coordinates, the eye template (circle and two parabolas) is initially fitted and then deformed over the image to best locate the eye features

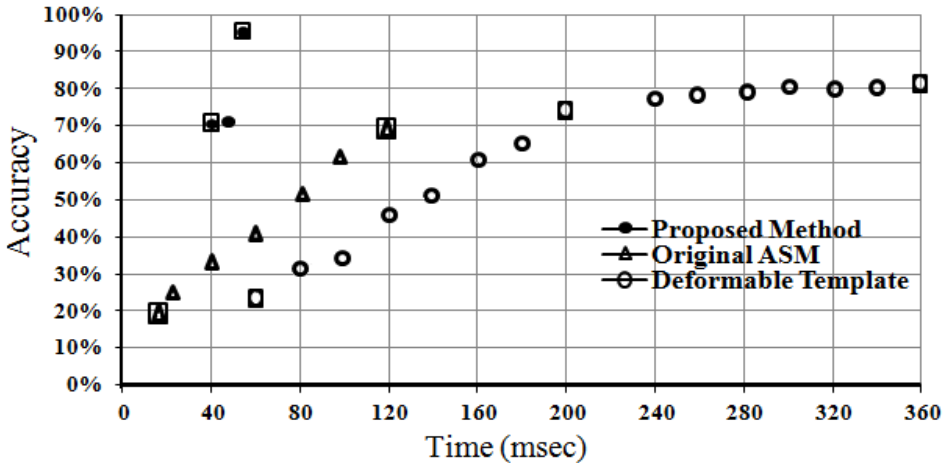


Fig. 4.19 The relationship between the accuracy and time.

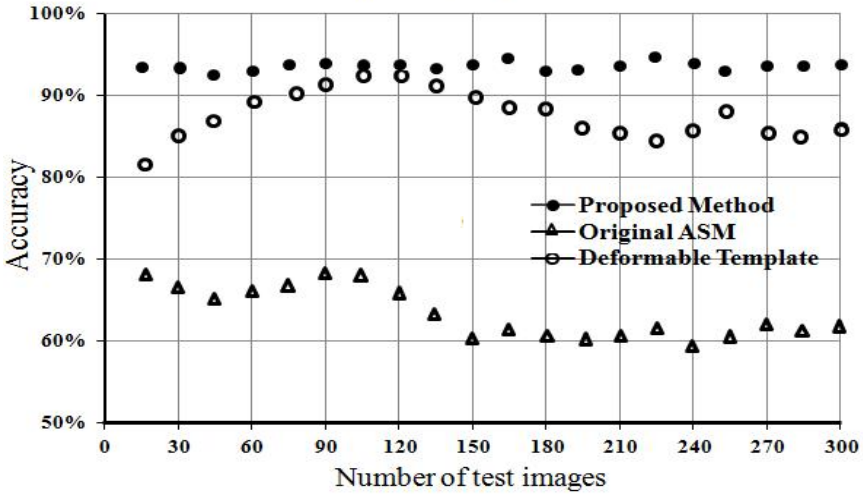


Fig. 4.20 The relationship between the accuracy and number of test images.

(eyelids and two eye corners). Consequently, the accuracy increases rapidly with respect to time until the template gets closer to the eye contour. The accuracy increases then slowly because small adjustments in rotation and scale are achieved.

In ASM, the mean eye shape is fitted to the image directly and the curve starts nearly after 8 msec. ASM then updates the locations of the eye shape points iteratively in order to approach the exact eye shape. Based on the development of the eye shape points, the accuracy increases accordingly.

In the proposed method, the curve starts after around 35 msec and the initial eye shape is obtained. This period is consumed by transforming the image into Log-polar domain, computing the first principal component as described in *sections (4.8and 4.10)* and sampling the LPEI for extracting the eye shape points. After 15 msec, the initial eye shape is refined as described in *section 4.11* and the final eye shape is obtained.

Figure 4.20 shows the accuracy as a function of the number of test images. The proposed method and ASM use the same strategy to refine the eye shape deformations. The difference is that in ASM, the eye shapes are labeled in Cartesian coordinate system whereas the eye shapes are labeled in Log-polar domain in the proposed method. Figure 4.21 depicts the variance

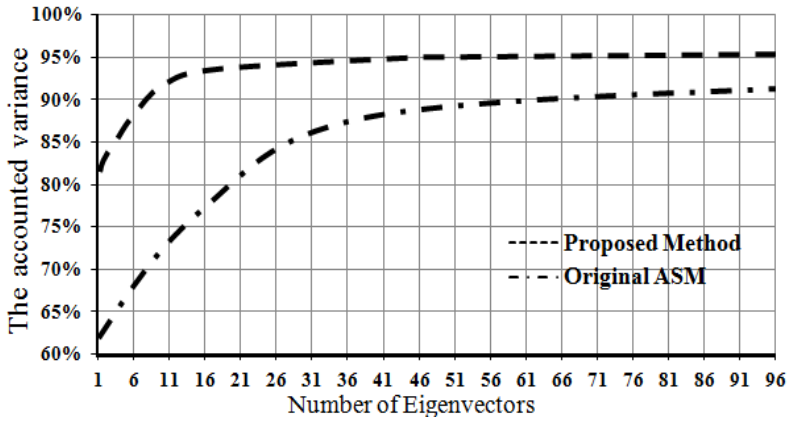


Fig. 4.21 The relationship between the accounted variance and number of eigenvectors.

accounted by each eigenvector based on Eq. (4.21).

4.12.3. Discussion

Illumination-insensitive eye shape detector has been implemented using PCA to interprete the

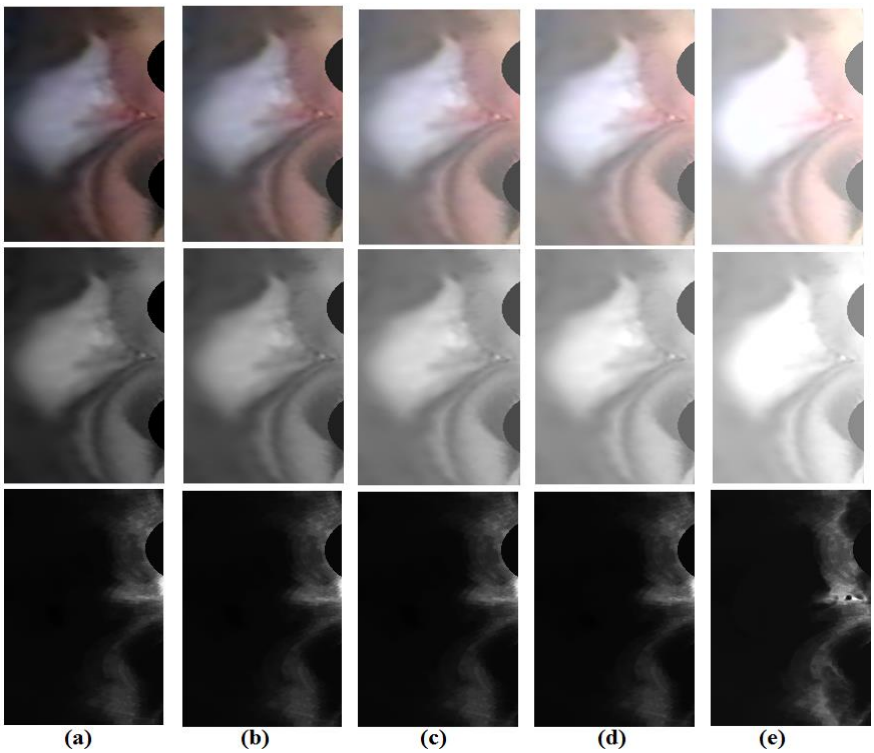


Fig. 4.22 The robustness of the proposed method against lighting conditions. (a) The reference case, Log-polar eye image, Y channel and LPEI. (b) Increasing Y channel by 30 intensity values. (c) Increasing Y channel by 70 intensity values. (d) Increasing Y channel 100 intensity values. (e) Increasing Y channel by 120 intensity values.

eye color information. When lighting conditions vary in the middel range of brightness, red, green and blue channels are affected by the same amount of change. Therefore, each pixel in the eye image is converted into a representation of color variation by the first eigenvector.

In order to investegate and emphsize this point, YCrCb color space is used. Y channel represents the brightness in the image. The brightness of an image is increased by converting the image into YCrCb color space, mainpulating the pixel values in the Y channel and finally conveting the channels (modified Y channel and CrCb) back into RGB color space.

Figure 4.22 (a), from top, shows a Log-polar eye image, corresponding Y channels and LPEIs. The Y channel has been changed four times as demonstrated in Figs. 4.22(b-e). The change occures by increacing the Y channel for each pixel 30 , 70, 100 and 120 intensity values; respectively. The corresponding effects of changing can be observed in RGB image domain as well as the LPEI domain. The coorsponding LPEIs are almost not influnced by the change of lighting conditions except the last case which is out form normal enviroment conditions. This proves the robustance of the proposed method agnaist the change of lighting conditions.

Moreover and besides the previous advantage of PCA, a uniform distrbution of pixel values in each row of LPEI is obtained. This means the change of pixels are discribed by patterns not by its values. This eases segneficantly the detection operation. Thus, two templates are used to describe the change in patterns at the first subsector in LPEI.

The existence of eyebrow must seriously be taken into account in Cartesian coordinate system. The eye shape may be attracted to the eyebrow instead of the eye contour (upper eyelid). In Log-polar domain, the eyebrow is guaranteed to be located in the second sub-sector. The second sub-sector has descending order of pixel values. Therefore, the eyebrow curve does not appear in the correlation result shown in Fig. 4.15(d). This makes the proposed method very robust and reduces eye shape deformations.

Figure 4.19 verifies that the initial eye shape suggested by the proposed method has a very high accuracy compared to the other two methods. The reason of this fact is the use of Log-polar transform. The eye contour exists along θ and the differences between the initial and exact eye shapes occur in $\log \rho$ coordinates only. Another reason is the strategy of describing the eye shape points freely without relating each one to a specific location/structure-area on the eye contour (ex. the fifth point indicates the location of eye corner).

Furthermore, detection time has been efficiently reduced because the proposed method does not use an iterative technique to search on the eye shape. It calculates the correlation between an LPEI and two templates that makes the detection time is fixed. This strategy prevents the eye shape to shrink into a point and ensures processing the eye images without running in an infinite loop for some complex eye states.

The achieved accuracy by the proposed method is very high comparing with ASM and DT. The use of the eye model to correct the $\log \rho$ deformations plays a main role in this result. By returning to labeling operation in order to implement this model, the correspondence between points from an image to another is preserved because the user has to focus only on $\log \rho$ coordinate at fixed θ coordinates for all images. On the other hand, in ASM, the user must label each point to match the corresponding structure location (ex. the first point represents the location of eye corner) as well as he must label this point in correspondence with the other images. In other words, suppose we try to label the second point which is on the upper eyelid and has only one property that it must be between the eye corner (1^{st} point) and the middle point on the upper eyelid(5^{th} point). The user cannot guarantee that the 2^{nd} point will be labeled in the next image to describe the same structure because of different scale, rotation, eye state, human observation…etc. This leads to uncertainty in describing the eye state even after applying the alignment operation. Therefore, the Log-polar eye shape model is very robust comparing to that

used in Cartesian coordinate system in ASM.

Figure 4.20 clarifies that the performance of the proposed method is very stable among the testing images. This is because of using PCA to deal with different lighting conditions and Log-polar transform to process different rotation, scale and rotation angles as well as the use of robust eye shape model to correct deformations. DT preformed nearly the same accuracy level in some cases. On the other hand, DT consumes long time to reach out this level. Therefore, if the aim is to track/detect the eye shape in real time, the proposed method is most appropriate.

The most significant information can be extracted from Fig. 4.21 is the accounted variance by the first eigenvector. The first eigenvector represents the direction of the most variation in the manually labeled shapes whether in Cartesian coordinate system or Log-polar domain. In Cartesian coordinate system, the first eigenvector accounts around 62% of the total variance whereas it accounts around 85% in Log-polar domain. The reason is that the eye model in the proposed method represents the variation in $\log\rho$ coordinates only as θ coordinates are fixed. Therefore, the reforming process of deformations in Log-polar domain is very robust and can be achieved using only the first a few eigenvectors. This contributes significantly in reducing detection time and increasing the accuracy level.

4.13. Conclusion

In this chapter, a new eye shape detection technique has been introduced and detailed. The common approach to detect the eye shape or any object shape is to enhance the image such as increasing the quality in some areas and killing the details in others and then detecting the desired object. Unlike the conventional methods, the proposed method studies and analyzes the eye texture/shape in different domain. Therefore, the eye image is converted into Log-polar

domain and, the shape and pixel value models are implemented accordingly.

Log-polar transform has been proven to be a baseline for implementing any eye shape detection technique. The eye structure matches the mapping strategy of Log-polar transform significantly. The eye contour exists along θ axis and the change of rotation is transformed into translation along this axis. This simplifies the problem of detecting eye shape points from searching on (x,y) coordinates in Cartesian coordinate system to search only on $\log \rho$ coordinates. On the other hand the change of scale/size is converted into translation along $\log \rho$ axis. Therefore, all eye states are explained by changing $\log \rho$ coordinates and consequently a very robust eye shape model has been obtained.

Out of the applications listed at the last section of the previous chapter, PCA is used to interpret the variance between eye color channels. The first principal component has demonstrated a very staple eye texture representation against a wide change of lighting conditions. Moreover, Log-polar transform is symmetric that enables a symmetric appearance of eye features. Upper and lower eyelids have convex curves as well as the two eye corners are in the same orientation. Therefore, by using these two properties, the pixel distribution in each row of an LPEI has been described by a uniform profile. The profile highlights a few patterns of changing of pixel values. One pattern of them indicates the location of the eye contour. Cross correlation based template matching is employed to detect the location of the eye contour contained in this pattern.

According to the proposed method, a set of fixed steps must be achieved. Therefore, there is no need for using an iterative technique. This makes the performance very stable regardless the complexity of the eye state because detection time for each testing image is fixed. Consequently, the structure of the proposed method is very promising and appropriate for real time applications.

.

The experimental results have varied the robustness of the proposed method in order to detect the eye shape under different scales, rotations, translations, eye states and lighting conditions. Therefore, the proposed method outperforms the conventional methods by achieving high accuracy with considerable low detection time.

Chapter 5

5. Eye Shape Detection in Log-Polar Domain Using Recurrent Neural Network

5.1. General

The Log-polar transform has been proven as a baseline for implementing an eye shape technique. As long as the eye contour exists along θ axis, whole eye states are represented by the change of $\log \rho$ coordinates. Therefore, the strategy of detecting these $\log \rho$ coordinates is the issue that can be investigated and achieved based on different concepts and approaches.

In the previous chapter, PCA has been employed to describe the variation between RGB color channels of Log-Polar eye images. Two templates have been used to perform cross correlation on the testing images. The templates illustrate the pattern of pixel values around the eye contour in each row of LPEIs. This strategy of searching on $\log \rho$ coordinates of the eye contour seems promising especially if the number of templates is increased to represent more verity of patterns.

However, nothing prevents us to try other strategies to enhance and improve the performance. In this chapter, instead of using a stack of templates, the generalization capability of neural networks is utilized in order to detect $\log \rho$ coordinates of eye contour. The pixel profile at each θ (modeling location) is considered as a discrete series of pixel values. First, a recurrent neural network (RNN) is used to model a pixel profile [91]. A short segment of pixel

values is moved pixel by pixel to visit all $\log \rho$ coordinates along the profile. The segment is continuously presented to RNN which is expected to learn the change of pixel values in order to predict one pixel value ahead. After training the RNN, the eye contour location in the pixel profile is detected.

In this chapter, a general biological concept of a neuron is given. The corresponding mathematical representation of the neuron (perceptron) is presented. A learning algorithm is explained to solve simple classification problems. The concept of perceptron is then extended to provide the rules of connecting neurons together. A multilayers neural network consisting of parallel connected neurons are discussed with corresponding developments of the learning algorithm. The notion of dynamic neurons is then highlighted and a recurrent neural network is built. The corresponding learning strategy is mathematically detailed. The prediction capability is then described in order to model the pixel values around the eye contour in a specific pixel profile. The recurrent network is expanded to increase the capability of modeling the pixel values and eye contour locations in different pixel profiles. All relative modifications, improvements and experimental results are deeply provided and discussed as well.

The results verified that using neural network can enhance the performance of eye shape detection. Moreover, it paves the way to investigate different structures on neural network to achieve the same task.

5.2. Neural Network

5.2.1. Introduction

Neural networks are a very active and challenging field to be studied and analyzed. In 1943, a valuable study has been reported by McCullock and PittsBack that a neuron behavior can be explained by two states according to the value of some key-thresholds [92]. In 1960, limitations of a simple perceptron has been reported and proven by Minsky and Papert [93]. This study caused a shock for scientists and terminated the hope of stepping ahead in this field. In 1980, a considerable attention and interest have been renewed by Hopfield [94]. After getting a little attention since it has been invented, back propagation learning algorithm is employed to train multilayer perceptrons [95]. Many sophisticated methods have been then proposed for describing various models of neural networks as well as various approaches for learning these models such as [96-98]. Based on the huge achieved progress, the neural networks has been revolutionary intervened with all scientific researches and domains; engineering, chemistry, biology, medical, finance…,etc.

The reason behind studying neural network is to better analyze and understand how the human brain works and behaves. In contrast, this leads to implement intelligent systems and develop machine learning algorithms. There are no common rules to address well how neural networks interact. Therefore, they are considered as a black box and classified under the concept of black mathematics that the mathematical model is known but the behavior is not predictable. On the other hand, this field is very interesting and challenging to be investigated. In this chapter, we try to provide simple and rough principles on the neural networks.

5.2.2. Biological structure of neuron

The neuron is the smallest particular component in the brain [99]. The brain consists of around 100 billion neurons interconnected by a very complex manner. Each neuron receives signals from a set of other neurons (around 10000) and sends its response to another set of neurons. The signals between neurons are sent by series via the connections electrically and chemically. Figure 5.1(a) shows the biological structure of a neuron. Neuron consists of cell body, nucleus, dendrites, axons and synapses. The cell body represents the human identity by nucleus which gives the form of DNA. A neuron contacts the other neurons using a vast net of dendrites as well as it transmits the processed information to the other neurons using axons. The connection points between the neurons whether in receiving or transmitting signals are called synapses.

A neuron possesses two states that are determined by the cell body; active and deactivate. The cell body generates the signals by summing up the incoming activations with respect to time [100]. If the received signals are strong enough, the neuron changes its state to be active. Hence, this activation is sent via synapses and may contribute in activating the other neurons. As long as the incoming signals are summed up through the time, the output signals are generated in a series in the range of frequency between few hertz to hundreds. The range is very

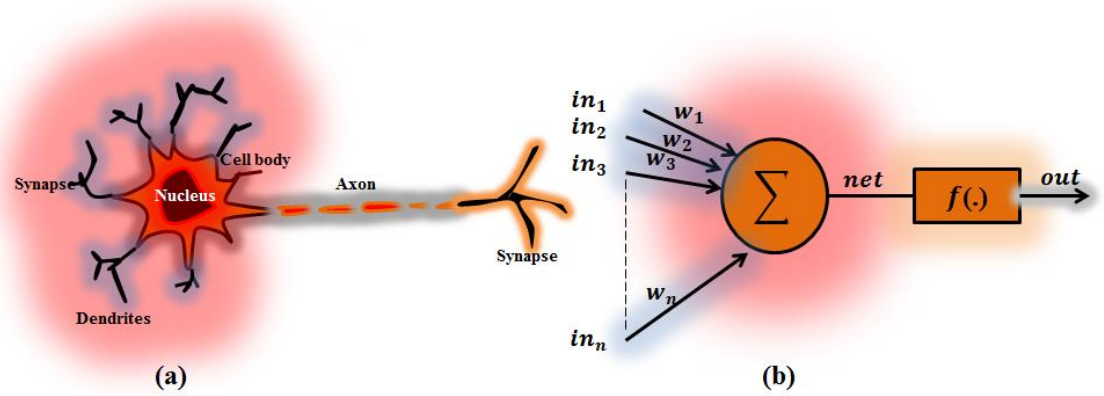


Fig. 5.1 Neuron structures. (a) Biological structure. (b) Mathematical structure (perceptron).

slow compared to any microcontroller cycling time. On the other hand, brain processes the information using a parallel distribution strategy [101]. Moreover, each neuron consumes around $10^{-6} joul$ whereas the artificial neuron in the supercomputers consumes around $10^{-10} joul$. This complex structure enables the brain to achieve very complex operations such as processing, analyzing, memorizing, recognizing, calculating and detecting all events that sensed by human.

5.2.3. Mathematical model of neuron

McCulloch and Pitts have proposed the first model of the neuron inspired by the biological structure [92]. Figure 5.1(b) shows the structure of the artificial neuron. The neuron receives the signals by dendrites. Therefore, the artificial neuron has *n* inputs. The received signals are sent by the other neurons in different activation levels. In the artificial model, a weight w_{th} is multiplied by each input in_{th}. A mathematical function $net(.)$ is used to determine the activation that simulates the process in the cell body and called internal output. Another function $f(.)$ that gives the identity of the neuron is used to represent the external out value of the neuron. This function is called the activation function and it can be mathematically linear or non-linear. The state of neuron is decided based on comparing the activation value with 0. In order to provide more flexibility, a threshold α is used to shift the output to the desired value of classification. This structure of neuron is called perceptron. The governing equation of the perceptron output O is given by Eq. (5.2).

$$net = in_1 w_1 + in_2 w_2 + in_3 w_3 + \cdots + in_n w_n - \alpha = \sum_{i=0}^{n} in_i w_i = In.W \tag{5.1}$$

$$O = f(\text{net}) \tag{5.2}$$

where $in_0 = -1$ is called bias and w_0 is the threshold α.

5.2.4. Mathematical operations by perceptron

Perceptron is a very powerful tool to calculate many mathematical and logical operations. First, some logical operations are presented with assuming the weights are 1 as well as the activation function is logical. Consequently, the neuron has two fixed states.

$$O = \begin{cases} 1 & net > \alpha \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Figure 5.2(a) illustrates three cases of logical functions by a neuron. The switching between functions is determined by changing only the threshold value.

Second, some mathematical functions are highlighted in Fig. 5.2(b) with assuming the activation function is linear. The sign of weights governs the calculation type whereas the corresponding values determine the function coefficients. Other algebraic functions can be typically represented using different activation functions and much more complex architecture.

5.2.5. Behavior analysis

Figure 5.3(a) shows an input space of a neuron that has two inputs and a linear activation function. The neuron output equation is linear and can be always considered as a line in two

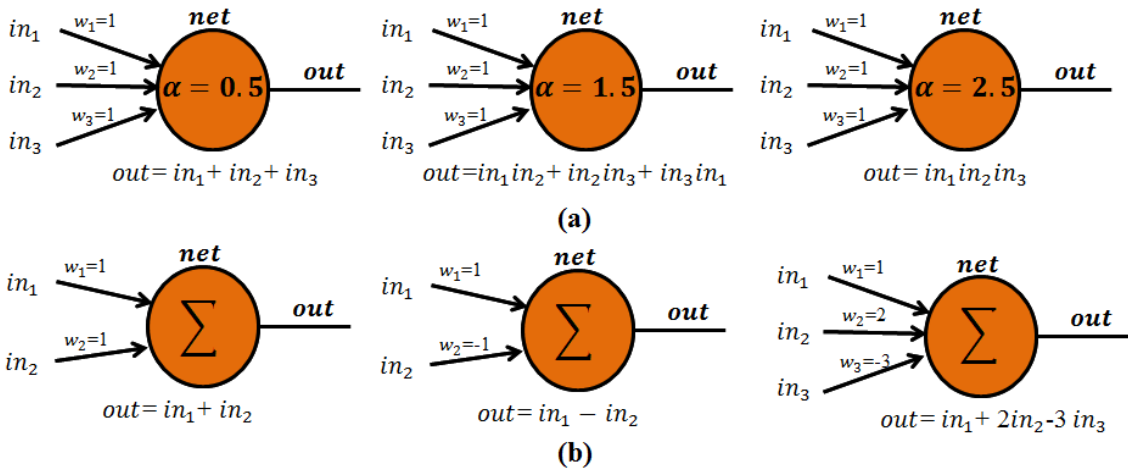


Fig. 5.2 Perceptron representation. (a) Logical operations. (b) Mathematical operations.

dimensions input space, Eq. (5.4).

$$in_2 = \frac{-w_1}{w_2} in_1 + \frac{\alpha}{w_2} \tag{5.4}$$

This line is called the decision boundary and it divides the input space into two regions; positive (shaded area above the boundary) and negative (non-shaded area below the boundary). The areas of the two regions are determined by w_1, w_2 and α which describe the slope and intersection point of the decision boundary with in_2 axis; respectively. The input sample is identified in the corresponding region based on the neuron's response. Figure 5.3(a) demonstrates four sampling points classified according to their positions. Likewise, the straight line in two dimensions space is converted into a straight plane in three dimensions space as shown in Fig. 5.3(b).

In conclusion, as long as the two groups of points are linearly separable by only one boundary, it is guaranteed to represent this boundary using a perceptron.

5.3. Constructing learning rules

A neuron can be simulated as a simple electrical circuit which consists of two on/off switches as inputs and an LED as the output as depicted in Fig. 5.4(a). The switches close the circuit and

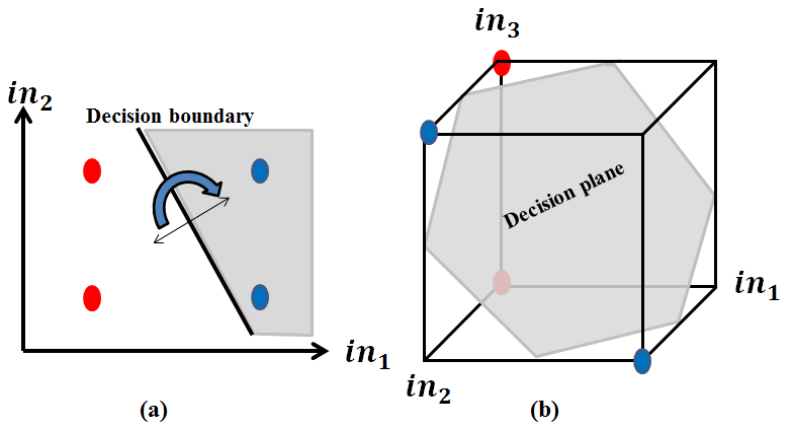


Fig. 5.3 Neuron input space. (a) In 2 dimensions. (b) In 3 dimensions.

permit the current to flow and reach the LED. We need the LED to be off when $S_1 =$ on and $S_2 =$ off. Obviously, this is impossible to be achieved by the current structure. However, if S_2 and S_1 have float terminals without any connection to the battery, the desired output can be obtained. Another order of switches and desired output impose a change in the circuit structure again.

In order to make the circuit more flexible for any order and desired output, a changeable resistance is added to each input as demonstrated in Fig. 5.4(b). The resistance value controls the amount of voltage applied on the LED.

In the neuron's mathematical structure, the changeable resistances are analogue to the weights. The resistances are changed by the human who observes the corresponding output. Similarly, the weights should be tuned mathematically and this tuning process is called the learning operation. Each change of a single weight produces a new separating function according to Eq. (5.4).

In order to address learning issue, we should answer on the question: if we have a point in a space, what is the best way to classify it by a boundary? The answer is to match the normal of this boundary with the point in the space as shown in Fig. 5.5(a). Consider a set of points I_n that are linear separable and we want to classify these points in two groups *P* and *N* according to the following conditions;

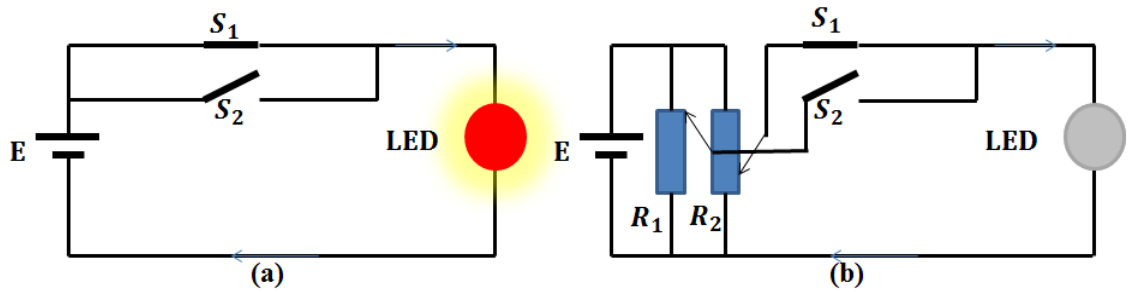


Fig. 5.4 Matching neuron and electrical circuit structures. (a) Convectional circuit. (b) Intelligent circuit.

$$I_n \in P \quad \text{if} \quad I_n.W > 0$$
$$I_n \in N \quad \text{if} \quad I_n.W < 0 \tag{5.5}$$

where W is a weight vector consists of $W = (w_1, w_2)$. The points are illustrated as vectors in Fig. 5.5(b) with referring to the red points as those belong to N group. If W is set to equal I_1, it would be guaranteed that I_1 is classified properly by the decision boundary which is perpendicular to the weight vector as shown in Fig. 5.5(b). On the other hand, this approach is not appropriate because the other points are misclassified probably. The same risk will be faced for applying this strategy on the next point in the space.

A promising solution is to add W to the current point. This procedure attracts W to the point in the same direction. Applying this operation iteratively on the points using the modified weight vector makes the decision boundary move until the correct classification is achieved. Figure 5.5(c) demonstrates the developing of the weight vector and classifying operation accordingly. One can observe that the length of the weight vector increases after each modification and the increment is less as the classification progresses. The weight vector is usually normalized to prevent any particular element to become largest and dominate the updating operation. Another important observation is that the decision boundary rotates around the origin according to the change of weights. This makes learning operation very slow and leads to fail in classifying some cases as illustrated in Fig. 5.5(d). Here we can deduce and

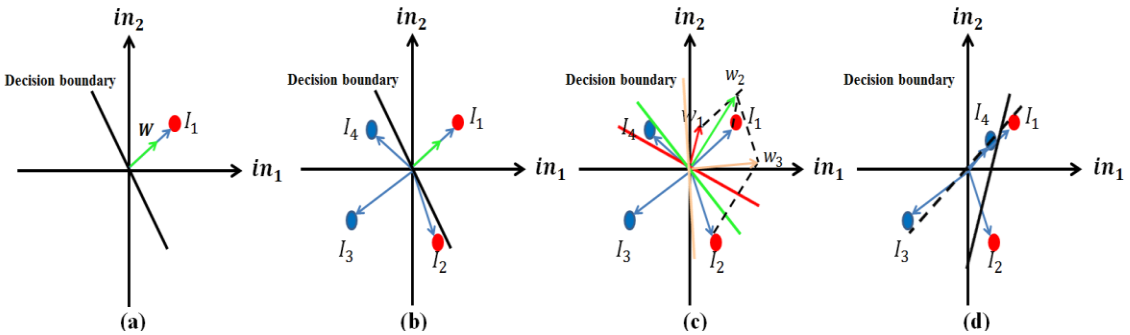


Fig. 5.5 Point classification by a neuron.

highly appreciate the interagency of using the threshold α to shift the decision boundary from the origin. To end this, each point tries to attract the weight vector into its direction and the operation is terminated when these forces become equal.

5.3.1. Learning conditions

The neural network is a supervised learning machine which means a teaching signal is used for each input to identify the corresponding desired output d_i (response) of the neuron [102]. The strategy of updating the weight vector is to add the current point. Adding (in general concept) attracts the weight vector to the points in the space. Therefore, adding and subtracting of the weight vector to/from the points is done according to the desired output d_i and the actual output of the neuron O_i. This rule is summarized as follows:

$$\text{if } d_i = 1 \text{ and } O_i < 0, \quad then\ W_{new} = W_{old} + In_i$$

$$\text{if } d_i = -1 \text{ and } O_i > 0 \quad then\ W_{new} = W_{old} - In_i$$

$$\text{if } d_i = O_i \quad \text{then } W_{new} = W_{old}$$

Accordingly, the two groups N and P are equivalent to the outputs -1 and 1; respectively. The above conditions can be rewritten as follows:

$$\text{if } e > 0, \quad then\ W_{new} = W_{old} + In_i$$

$$\text{if } e < 0 \quad then\ W_{new} = W_{old} - In_i$$

$$\text{if } e = 0 \quad \text{then } W_{new} = W_{old}$$

where $e = d_i - O_i$ is the error signal and it is also called the teaching signal. The teaching signal reflects the neuron response to the inputs with respect to the desired output. The sign of the teaching signal indicates the current belonging of the input point to the two groups whereas the value indicates how the current neuron's output far from the desired output.

One should carefully notice that the sign of e matches the mathematical operation between

current weight and input. Therefore, the previous three conditions can be combined by Eq. (5.5).

$$W_{new} = W_{old} + e * In_i \tag{5.5}$$

5.3.2. Learning rules

The aim in this section is to introduce the rules of how the weights can be tuned to match the desired outputs. Suppose a neuron is created with two inputs and a linear activation function as depicted in Fig. 5.2(b). A set of *K* inputs are presented with corresponding *K* desired outputs. An error function is defined by Eq. (5.6).

$$E = \sum_{i=1}^{k} \frac{1}{2} (d_i - O_i)^2 = \sum_{i=1}^{k} \frac{1}{2} e_i^2 \tag{5.6}$$

The error surface is constructed in the weigh space and consists of valleys, mountains and sallow areas. We need to search on a position on the surface that has the smallest error value. The position coordinates corresponds to the optimal solution of classifying the input points. The common rule is to update the weights iteratively until reaching the optimal solution coordinates. The common updating rule is applied as follows:

$$W_{i\,new} = W_{i\,old} + \eta \frac{dE}{dw_i} \tag{5.7}$$

where η is the learning step which determines the amount of steeping down ahead the minimal peak on the surface. The term $\mathrm{dE}/\mathrm{d}w_i$ describes how the error changes with respect to a small change in the weight w_i. In other words, it represents the slope in the direction of w_i at the current solution point on the error surface. This term cannot be calculated directly and Chain rule is used [102].

$$\frac{dE}{dw_i} = \frac{\partial E}{\partial O_i} \frac{\partial O_i}{\partial net_i} \frac{\partial net_i}{\partial w_i} \tag{5.8}$$

where $\partial \mathrm{E}/\partial O_i$ represents how the error changes when the neuron output changes and equals

to $-e_i$. The term $\partial O_i/\partial net_i$ represents how the neuron activation changes with respect to small change in the internal output of the neuron. This term actually is related directly to the activation function of the neuron and it is equal to one (1) when the activation function is linear. The term $\partial net_i/\partial w_i$ describes how the internal output changes when a particular weight changes. This term is the corresponding particular input. By substituting these terms in Eq. (5.8), Eq. (5.9) is obtained.

$$\frac{dE}{dw_i} = -e_i * 1 * In_i \tag{5.9}$$

Hence, the final version of the updating weight equation is given as follows:

$$W_{i\,new} = W_{i\,old} + \eta e_i In_i \tag{5.10}$$

One can observe that the minus sign is disappeared because we step down on the error surface.

5.4. Activation function

In the previous sections, we used a linear function to skin out the internal output net_i of the neuron. The activation function is very important in order to solve complex problems of classification and to make the neuron capable to deal with nonlinear issues. Figure 5.6 shows the profile of some activation functions. The most popular one to be used in neural networks is hyperbolic function which belongs to the sigmoidal function family.

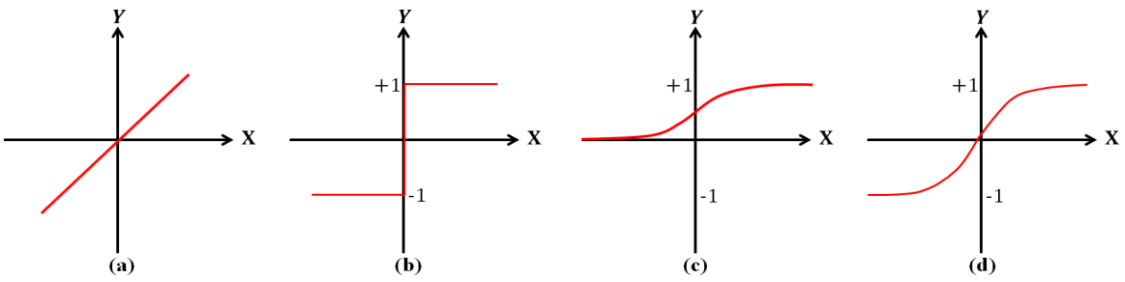


Fig. 5.6 Activation functions. (a) Linear function. (b) Step function. (c) Sigmoid function. (d) Hyperbolic function.

5.4.1. Hyperbolic activation function properties

Equation (5.11) represents the mathematical form of the Hyperbolic function in two dimensions space in Fig. 5.6(d).

$$\tanh(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \tag{5.11}$$

The horizontal axis represents the internal output of the neuron net_i whereas the vertical axis is the function value. The function is bounded in the range [-1 1]. Consequently, the neuron output will vary in the same range. This prevents the output to reach large values that may make the learning operation very hard. The function profile can be divided into three parts that to the nonlinearity characteristics: linear, exponential and saturated. In the linear part, the neuron is very sensitive to the change of the input. The sensitivity decreases towards the exponential part until vanishing in the saturated part. Based on this discussion, the function smoothly and continually changes from positive part to negative part and vice versa.

5.4.2. Behavior of activation function during learning phase

The activation function skins out the internal output of the neuron. Therefore, it must be differentiable to meet the requirements of the learning operation in Eq. (5.8), term $\partial O_i/\partial net_i$. Hence, the updating equation of weights becomes as follows:

$$W_{i\,new} = W_{i\,old} + \eta e_i(1 - O_i^2)In_i \tag{5.12}$$

where $1 - O_i^2$ is the differential form of the hyperbolic function. Figure 5.7(a) shows particularly, the differential profile $\left(1 - O_i^2\right)$, the output profile O_i and the error surface (error signal e_i) when the desired output d_i is -1 and 1; respectively.

One can deduce directly that the derivative of the function is positive. When the internal output is zero, the derivative is at maximum as well as the error signal is at 1 or -1. Consequently, the term $e_i(1 - O_i^2)$ is at the maximum value and In_i will fully contribute in updating the weights ($\eta = 1$). On the other hand, the linear part of the hyperbolic function lies approximately in the range of [-0.4 0.4] of the internal output. The corresponding derivative value in this range is considerably large. Similarly, the updating amount of the weight is large and the error is significantly reduced in this area. If the neuron output is in the saturated part and the output matches the desired output, the error signal is zero and it is at maximum if the output completely mismatches the desired output.

Based on the previous discussion, the neuron behavior is controlled by the change of the weight w. The value governs the slope (the neuron sensitivity) and the range of the linear part. The sign governs the slop direction and consequently the output sign. Opposite sign produces reverse function characteristics.

The neuron can deal with points that distributed around the origin. In some critical cases,

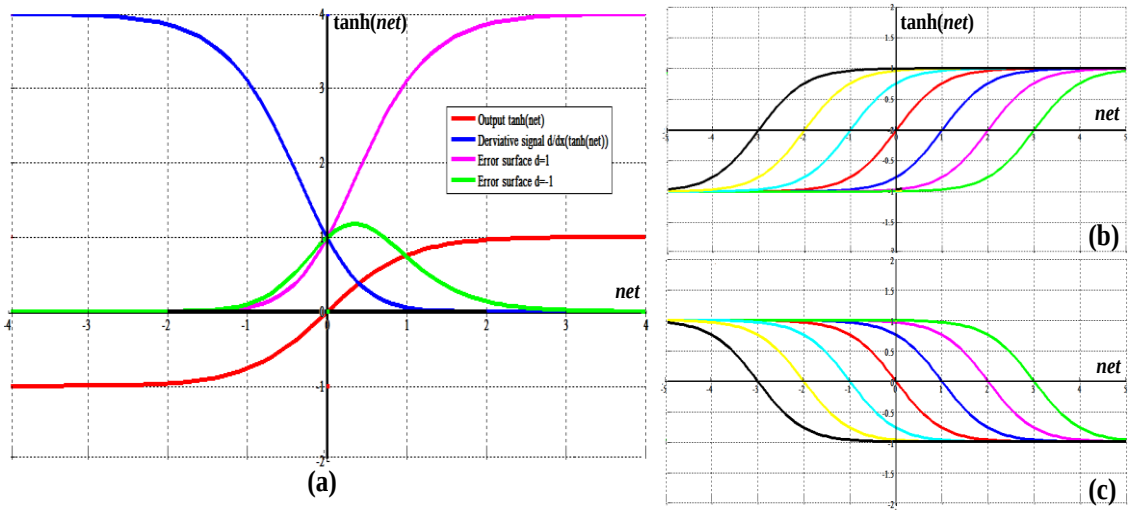


Fig. 5.7 Behavior of Hyperbolic function. (a) Learning operation signals. (b) Change of threshold. (c) Inversed function (inversed slop).

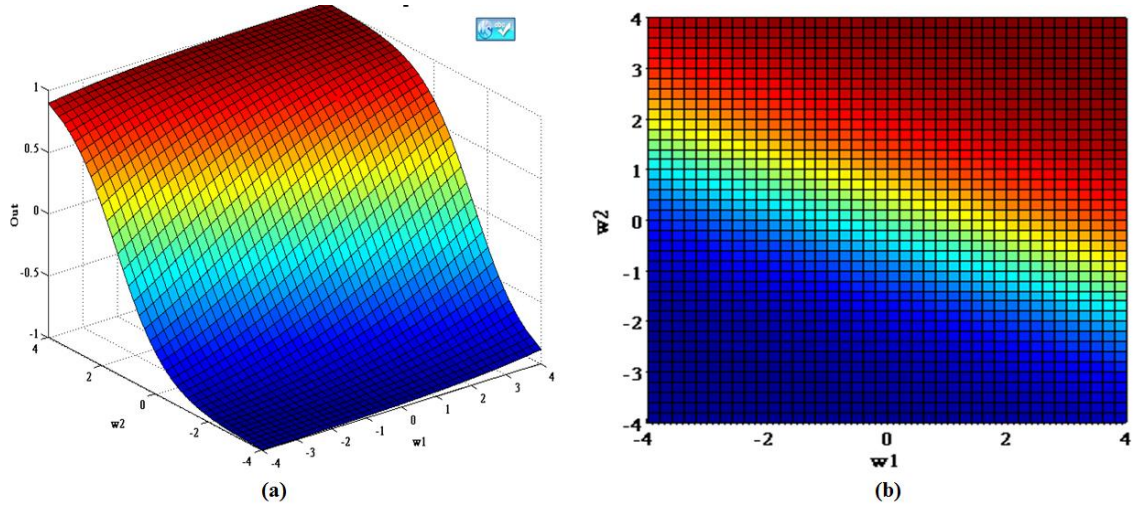


Fig. 5.8 Activation function. (a) Output surface in 3D. (b) Decision boundary by the surface in the weight space.

some points are very close to the origin and belong to two groups as illustrated in Fig. 5.5(d). The neuron slope will be very sharp and unstable to classify these points. Thus, the learning operation is very aggressive and the neuron gets saturated rapidly. Therefore, more dynamical properties in the neuron structure should be provided. This is achieved by adding the threshold $\alpha = w_0$ which shifts the function horizontally from the origin with preserving the slope characteristics. The threshold facilitates the learning operation efficiently without reaching the saturation or unstable state. The effects of α is plotted in Fig. 5.7(b-c) for positive and negative slope.

5.5. 3D visualization of the neuron surface

Figure 5.8(a) shows the neuron output surface in three dimensions space which is mathematically represented in Eq. (5.14).

$$net = in_0 w_0 + in_2 w_2 + in_2 w_2 = \sum_{i=0}^{2} in_i w_i = In.W \tag{5.13}$$

$$O = \tanh(\text{net}) \tag{5.14}$$

The same properties in the previous section are yielded. The difference is that w_1 and w_2 govern the slope in the vertical and horizontal directions. In other words, the change of slope in the vertical direction is similar to the slope concept in 2D space. On the other hand, the horizontal slope can be considered as rotation in the two dimensions space created by w_1 and w_2 as illustrated in Fig. 5.8(b). For example, if $w_1 = w_2$, the angle degree of the decision boundary is $\pm 45°$. If w_1 and w_2 have the same sign, the slope is negative and otherwise it is positive. w_0 has the same effect of shifting the surface far from the origin.

In conclusion, the task of a neuron is to tune its surface parameters to fit the input points according to the desired outputs. The input points are fixed during the tuning operation. Tuning can be achieved by the learning technique detailed in *section* 5.3.2. This demonstrates the beauty of neural networks to change the neurons' surfaces according to the problem in charge.

5.6. 3D visualization of the error surface

Intuitively, the need of presenting an example on a 3D error surface strongly arises to fully address the visualization concept. Figure 5.9(a) shows the error surface in the weight space constructed by summing two particular error surfaces. The two particular error surfaces are

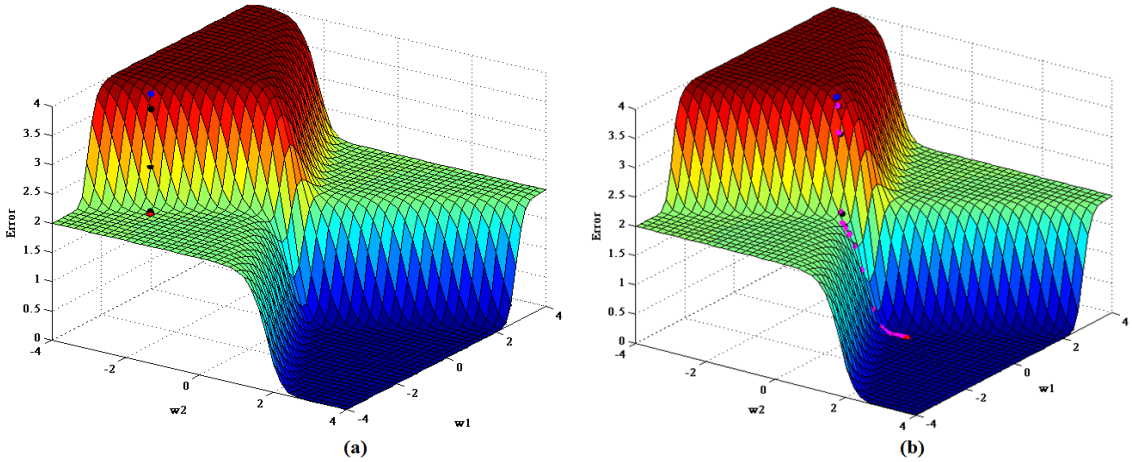


Fig. 5.9 Error surface. (a) Wrong weight initialization. (b) Proper initialization.

obtained by two input points that should be classified in two groups at different outputs; -1 and 1 respectively. The surface is symmetric and consists of valley, hill and shallow areas.

The learning technique searches on a point on the error surface that indicates the global minimum error value. Obviously, this point is located at the valley area. The corresponding point coordinates (weights) are the optimal solution of the classification issue. In other words, the learning technique updates the weights based on the current error value. Each update step moves the solution point on the error surface towards the global minima. Each update corresponds to a change in structure (parameters) of the neuron activation surface.

One question can be highlighted is how to initialize the learning operation? Which is equivalent to what is the first location of the solution point on the error surface? The initial weight values determine the location of the solution point. Therefore, these values should be in the range out of identifying the shallow area whether at the beginning or during the learning operation progress. In the shallow area, the derivative (slope) of the activation function is zero and thus the weight will be not updated at all. Figure 5.9(a) illustrates this key-issue by showing the trajectory of the weights updating process. The blue point indicates the initial values (at the hill area) whereas the red point is the ending or stacking location of the learning operation. Based on the ending point, the neuron fails to classify the input points correctly according to the desired outputs. Figure 5.9 (b) shows another initializing weight values and the update trajectory ends in the valley area.

It can be observed in the last case that the neuron successfully classifies the input points. The trajectory is very long and the learning operation consumes long time. On the other hand, the neuron has no bird's view (aerial view) to see the best location on the error surface. The current studying issue is very simple because it is investigated in 3D space and the geometrical structure of the error surface is available. This issue becomes more complicated and just

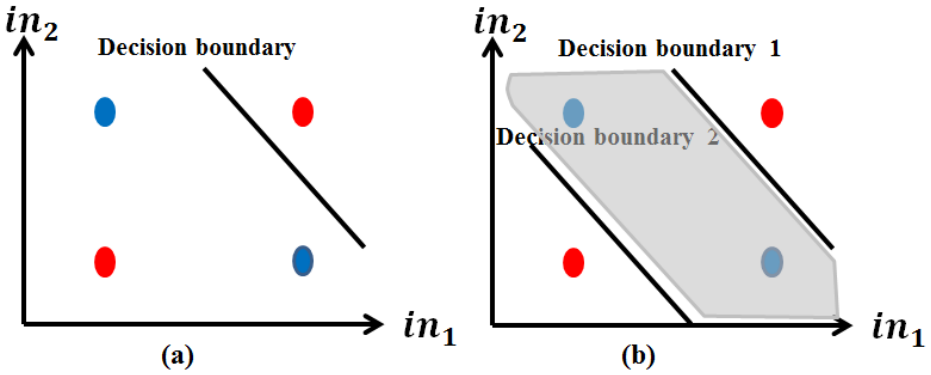


Fig. 5.10 XOR classification problem. (a) Wrong classification using one decision boundary. (b) Proper classification by two boundaries.

imaginable in higher dimensions. Thus, only the current error value, input point, weights and desired output are provided.

In conclusion, if the weights are very large, the neuron reaches the saturation region and the sensitivity becomes very high. Therefore, the best sufficient and safest choice of initializing the weights is to be in the linear part [-0.4 0.4] of the activation function where the updating step is very large.

5.7. Neuron limitations

The decision boundary of a neuron is represented as a straight line in 2D space and as a linear plane in the higher dimensions space. A single neuron has been presented as a very powerful classifier of two linear separable sets of points. On the other hand, this assumption has been disproved for some cases of point distribution. Figure 5.10(a) highlights the well-known XOR classifying problem by showing a set of points that are linearly separable. A single neuron cannot classify the points without misclassifying one point in the best solution. This is because only one straight line can be employed to fit the points.

The optimal solution is to use two straight lines as shown in Fig. 5.10(b). Obviously, two neurons should be used to represent the two lines in the input space. The question can be pinned

down is how to achieve this task and control the behavior of the two neurons in parallel? In order to answer this question, the multilayer perceptrons or neural network is presented.

5.8. Multilayer perceptrons or neural networks

Neural network is a tool inspired by how the human brain works. A neural network consists of three layers; input, hidden and output as shown in Fig. 5.11. The input layer represents the number of input junctions. The hidden layer contains a set of neurons that called the hidden neurons. The name is originated by the fact that these neurons have no direct terminal to the external environment. The output layer contains one neuron or more depending on the application. Each neuron receives the inputs from previous layer and sends its activation to the next layer. The activations are delivered via a set of weighted connections. The values of weights represent the strength between neurons.

The essence of the neural networks is the magical use of hidden neurons. By returning back to the XOR classification problem, the inputs are sent to the two hidden neurons. Each neuron tries to see the input space from a different perspective. The output neuron has an input space as well. This space is constructed by the two activations of the hidden neurons. In other words, the

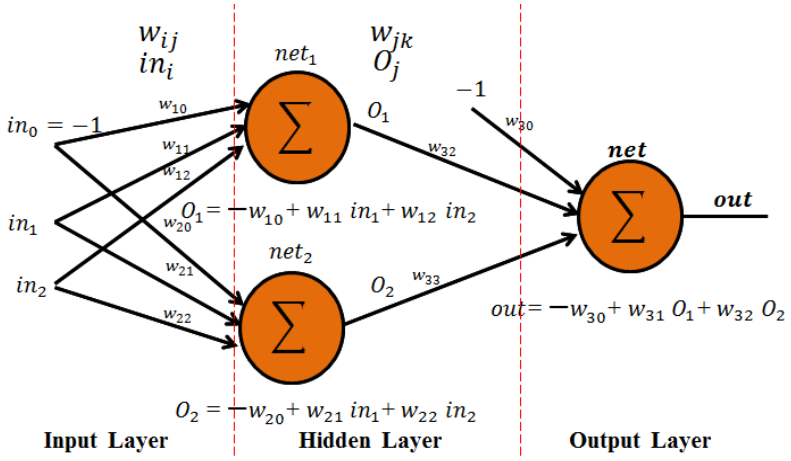


Fig. 5.11 Multilayer perceptrons (network).

output neuron sees the outputs of the hidden neurons and tries to classify them linearly. These operations in the hidden and output layers are achieved significantly by tuning the corresponding weights using learning technique (back-propagation technique) [91].

The error surface E is constructed by summing up the particular error surfaces e_k that generated by subtracting the network output Out_k from the corresponding desired outputs d_k.

$$E = \sum_{k=1}^{N} \frac{1}{2} (d_k - Out_k)^2 = \sum_{i=k}^{N} \frac{1}{2} e_k^2 \tag{5.15}$$

The same strategy of updating the weights of a neuron is used as in Eq. (5.16). On the other hand, the structure of the neural network imposes some particular updating rules related to each layer.

$$W_{new} = W_{old} + \eta \frac{dE}{dw} \tag{5.16}$$

Updating rules in the output layer:

The inputs of the neuron in the output layer are the outputs O_j of the hidden neurons O_1 and O_2. These outputs are sent via two weighted links. The neuron output equation is given as follows:

$$net_k = \sum_{j=1}^{2} w_{jk} O_j \tag{5.17}$$

$$Out_k = \tanh(net_k) \tag{5.18}$$

where w_{jk} is the weight between j^{th} hidden neuron and k^{th} output neuron.

$$\frac{dE}{dw_{jk}} = \frac{\partial E}{\partial Out_k} \frac{\partial Out_k}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} \tag{5.19}$$

$$\frac{dE}{dw_{jk}} = -e_k \left(1 - Out_k^2\right) O_j \tag{5.20}$$

Updating the weights in the hidden layer:

A hidden neuron receives the inputs from the input layer and calculates the output using Eq. (5.22).

$$net_i = \sum_{i=1}^{2} w_{ij}\, In_i \tag{5.21}$$

$$O_j = \tanh(net_i) \tag{5.22}$$

where w_{ij} is the weight between i^{th} input and j^{th}hidden neuron.

$$\frac{dE}{dw_{ij}} = \frac{\partial E}{\partial Out_k}\frac{\partial Out_k}{\partial net_k}\frac{\partial net_k}{\partial O_j}\frac{\partial O_j}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}} \tag{5.23}$$

$$\frac{dE}{dw_{ij}} = -e_i\left(1 - Out_k^2\right)w_{jk}\left(1 - O_j^2\right) In_i \tag{5.24}$$

It can be deduced that w_{jk} appears in the two updating rules. Therefore, the weight in the hidden layer are first updated and then those in the output layer. The term $\delta = -e_i\left(1 - Out_k^2\right)$ is back-propagated from the output layer to the input layer.

The working strategy of the neural network can be summarized by two steps; feedforward and back propagation. In the feedforward step, the weights are fixed, the inputs are changed and the outputs of all neurons are calculated. This is the reason of calling this kind of neural network feed-forward neural network. In the back-propagation step, the error is calculated and the weights are updated. This is the reason of calling the learning technique back-propagation.

5.9. Recurrent neural network

In the previous section, given a set of points that describe unknown distribution, the neural networks are able to discover and model this distribution based on some examples encoding the input/output behavior. The modeling is done by fitting the point distribution by a set of lines. The lines divide the distribution into independent regions. Each region identifies the points according to corresponding teaching signal (or desired output). The output neuron of the neural network classifies these regions into only two groups. If the neural network is trained properly, it can perform *generalization* instead of fitting the point distribution [104]. The neural network

intelligence is evaluated by increasing the capability of this main issue. Once the network has the capability of generalization, it can be also employed to predict the future values of the function through the time. The design of the neural network structure plays the main role to improve the term of prediction. Recurrent Neural Network (RNN) has been introduced as a very powerful structure for future prediction based past events learning [105].

RNN can be used in either continuous or discrete time domain. This increases the potential of wide applications in dynamical systems. The neurons in feed-forward neural networks are connected to each other by direct forward connections. In RNN, neurons have recurrent connections (feedback loop) to enable the network to attract and analyze the dynamic aspects of the modeling function and store some information for future use [106]. In other words, RNN has a memory space and it can control the angle between a vector and its projection in this space whereas the angle is fixed in the feedforward network [107][108]. Hence, the past states of the neurons are used as a part of the current input. This increases the generalization capability. Moreover, RNN runs continuously that meets the requirements of scanning process such as searching on the eye contour location in a pixel profile.

5.9.1. Recurrent neuron structure

Figure 5.12 shows the structure of a neuron that can be an element in RNN. The neuron has a recurrent connection. The recurrent connection sends back the output into the input. The recurrent connection is weighted as the feedforward connection of the input *in.* Intuitively, the recurrent weight w_1 can be tuned and updated as the feedforward one w_1 and the amount of the output value that sent back changes accordingly. Equation (5.25) governs the behavior of the recurrent neuron at $k+1$ discrete step.

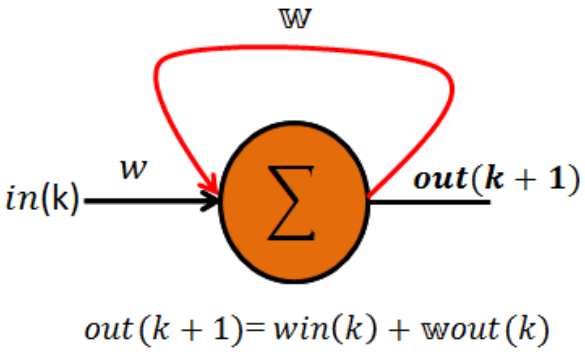


Fig. 5.12 Recurrent neuron.

$$net(k+1) = in(k) * w + Out(k) * \mathbb{w} \tag{5.25}$$

$$out(k+1) = \tanh\big(net(k+1)\big) \tag{5.26}$$

Suppose that the neuron runs and processes the input/previous-output continuously using Eq. (5.26). The term $out(k+1)$ means that the neuron tries to predicate one future value ahead. The prediction is done using the current input and weighted previous output (dynamics). Thus it can be said that the neuron memorizes the previous state somehow and $\mathbb{w}_1$ controls the memory resolution. The *memory resolution* is how much the amount of information will be stored in the memory for the next cycle.

5.9.2. Learning of single recurrent neuron

The same strategy of training feedforward neuron can be employed to train a recurrent neuron. On the other hand, some differences exist due to the recurrent neuron structure. A continuous input signal (series) $in(k)$is used as well as the corresponding series of desired outputs $d(k)$. The error $E(k)$ should be processed at the current moment.

$$E(k) = \frac{1}{2}\big(d(k) - Out(k)\big)^2 = \frac{1}{2}e^2(k) \tag{5.27}$$

The weights are updated using Eq. (5.28).

$$W(k)_{new} = W(k)_{old} + \eta \frac{dE(k)}{dW} \tag{5.28}$$

The term $dE(k)/dW$ describes how the error at the current input sample changes with respect to the weight change. Because of the structure, this term must be calculated individually for each type of weight.

$$\frac{dE(k)}{dw} = \frac{\partial E(k)}{\partial Out(k)} \frac{\partial Out(k)}{\partial net(k)} \frac{\partial net(k)}{\partial w} \tag{5.29}$$

$$\frac{dE(k)}{dw} = -e(k)\left(1 - Out^2(k)\right) \frac{\partial net(k)}{\partial w} \tag{5.30}$$

$$\frac{\partial net(k)}{\partial w} = in(k) + \frac{\partial Out(k-1)}{\partial w} \mathbb{w} \tag{5.31}$$

The above equations show the rule of updating the feedforward weight. The second term in the last equation is considered as an extra term of the normal updating rule. This term appears because of the recurrent connection.

$$\frac{dE(k)}{d\mathbb{w}} = \frac{\partial E(k)}{\partial Out(k)} \frac{\partial Out(k)}{\partial net(k)} \frac{\partial net(k)}{\partial \mathbb{w}} \tag{5.32}$$

$$\frac{dE(k)}{d\mathbb{w}} = -e(k)\left(1 - Out^2(k)\right) \frac{\partial net(k)}{\partial \mathbb{w}} \tag{5.33}$$

$$\frac{\partial net(k)}{\partial \mathbb{w}} = Out(k-1) + \frac{\partial Out(k-1)}{\partial \mathbb{w}} \mathbb{w} \tag{5.34}$$

The above equations address the rules of updating the recurrent weight. The input does not appear in this question because it is independent. On the other hand, this form can be interpreted by taking into account that the weight and output are changeable (variables).

In conclusion, two gradients information of the two weights affect the development of the learning operation. The change of the feedforward weight matters in the current input whereas the change of the recurrent weight lasts forever because if the input is zero, the output will be not zero and the weighted previous state will affect the output. This training or learning strategy of the RNN gives the ability of modeling the oscillations [109].

5.10. Recurrent neural network structure

After providing a simple explanation on the neural networks to construct a good background, we aim to involve this learning mechanism in order to detect the eye contour in the log-polar eye image. We consider a pixel profile at θ as a series of pixel values that change along $\log\rho$ axis (at each $\log\rho$ coordinate step). A recurrent neural network will be trained in order to predict the pixel value one step ahead (at the next $\log\rho$ coordinate) using a set of previous inputs, past states and current input. Obviously, this task cannot be achieved using only one recurrent neuron and the high capability of modeling nonlinearity in the hidden layer is needed. Consequently, a recurrent neural network is constructed.

According to the architectures, RNN is classified into two types; fully connected and partial connected networks. In the first type, a neuron can be connected to any neuron regardless the layer. Second type is created by breaking this freedom of connection style [110]. Therefore, there are different types of recurrent neural network structures [111].

Besides the advantages of using recurrent connections, the network suffers from stability

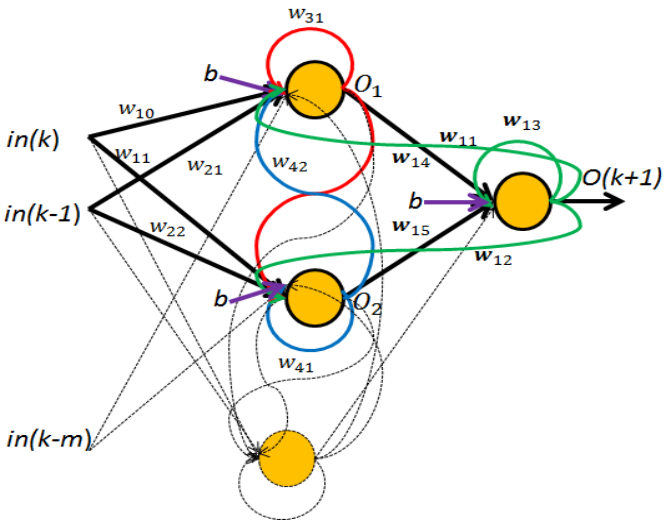


Fig. 5.13 Recurrent neural network.

because the recurrent connections may lead the network to an unstable behavior without reaching the error convergence. In order to settle this problem, a proper structure and learning technique should be used

Figure 5.13 shows the general structure of the network that will be used for detecting the eye contour. It is a fully connected network and consists of three layers; input, hidden and output. Hence, each neuron receives signals from the other neurons as well as its previous state. In other word, each neuron has explicit knowledge of the complete state of the network which increases the stability. The input layer consists of *l* inputs; one delivers the current input at the moment and the others are previous inputs. The output layer has one neuron and its output is one step ahead. This means the prediction capability of the network is used.

5.10.1. Learning recurrent neural network

In order to train the network, the error $E(k)$ between the desired output *d*(*k*) and the actual network output *Out*(*k*) must be minimized. As long as the network is employed for prediction, the desired output is one step ahead. This can be done using the gradient descent rule shown in Eq. (5.36).

$$E(k) = \frac{1}{2}\big(d(k+1) - Out(k)\big)^2 = \frac{1}{2}e^2(k) \tag{5.35}$$

$$\Delta w = -\eta \frac{dE(k)}{dw_{ij}} \tag{5.36}$$

In order to update the neurons' weights, we unify the representation of neuron terminals (input/output) by y_h. Thus, *I* denotes current/previous inputs, *C* denotes bais *b* and *U* denotes a neuron output *O*, as follows:

$$y_h(t) = \begin{cases} in_h(k) & \text{if } h \in I \\ b = -1 & \text{if } h \in C \\ O_h(k) & \text{if } h \in U \end{cases} \tag{5.37}$$

The output of h^{th} neuron is given by Eq. (5.39).

$$net_h(k) = \sum_{l \in U \cup I} w_{hl} y_l(k) \tag{5.38}$$

$$O_h(k+1) = f\big(net_h(k)\big) \tag{5.39}$$

where f (.) is the output activation function. The activation function has a significant role to model the nonlinearities of pixel values in the eye profiles. Therefore, the standard sigmoid function is used to represent the neurons' outputs. Notice that the output of a neuron described by Eq. (5.39) is not affected by the previous input until step $k+1$. This fact matches the prediction purpose.

The gradient step of a neuron output (activation $O_h(k)$) with respect to its internal output ($net_h(k)$) is given by Eq. (5.40).

$$\frac{\partial O_h(k)}{\partial net_h(k)} = O_h(k) * \big(1 - O_h(k)\big) \tag{5.40}$$

The sensitivity of a neuron output to a small change in weights of its input connections is calculated as follows:

$$\frac{\partial O_h(k)}{\partial w_{ij}} = \frac{\partial O_h(k)}{\partial net_h(k)} \frac{\partial net_h(k)}{\partial w_{ij}} \tag{5.41}$$

$$\frac{\partial net_h(k)}{\partial w_{ij}} = \sum_{l \in U \cup I} w_{hl} \frac{\partial O_l(k)}{\partial w_{ij}} + \delta_{ih} y_j(k) \tag{5.42}$$

where δ_{ih} is the Kronecker delta function and equals to one if and only if *i*=*h*, otherwise it is zero. The term $w_{hl} * \partial O_l(t)/\partial w_{ij}$ represents explicitly the effect of changing the other neurons' outputs with respect to w_{ij} of other neurons.

Recurrent neural network seems to be a very promising tool to model the change of pixel values around the eye contour in a pixel profile. Moreover, the generalization capability can be used for dealing with different lighting conditions instead of using only two templates as in the previous chapter. In the next section, we explain the proposed approach for detecting the eye

contour in the pixel profiles.

5.11. Experimental Results

5.11.1. Setup and methodology

In this section, we want to highlight setup parameters of the neural network with. The setups of ASM and DT are the same as explained in the previous chapter as well as the same two databases, eye shape description, manually labeling operation and the accuracy/comparison rules are used.

The recurrent network has the order of neurons of 20-20-1 for input, hidden and output layers; respectively. The activation function of a neuron output is the standard sigmoid function. The θ sampling rate is S=22. Therefore, each eye profile contains 1X256 pixels and each image is represented by a series of N=16 eye profiles. Consequently, this series consists of 16*256 pixel values.

In training stage, we used 50 eye images to train the recurrent neural network by 5000 epochs. Each eye image is converted into grayscale, rescaled between 0 and 0.9 and then transformed into a series of eye profiles according to the θ sampling rate S. The pixel values at

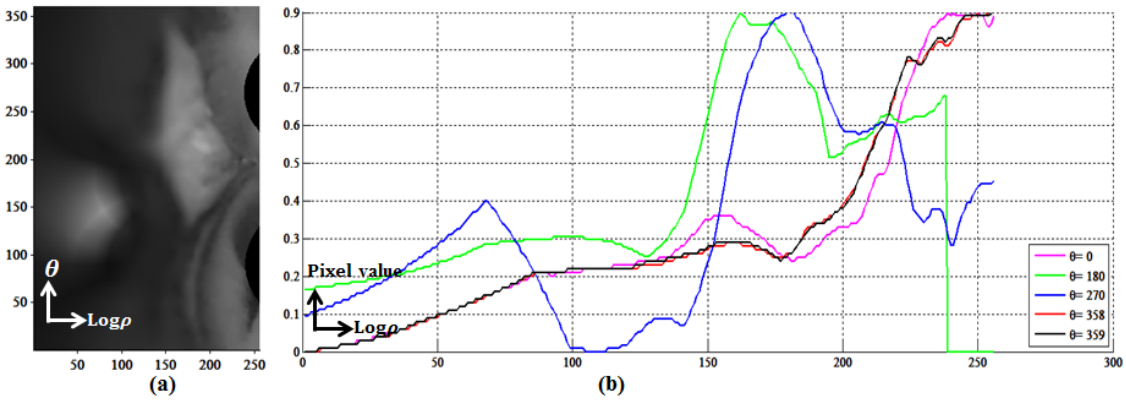


Fig. 5.14 (a) Log-polar eye image in grayscale. (b) Pixel profiles at different θ.

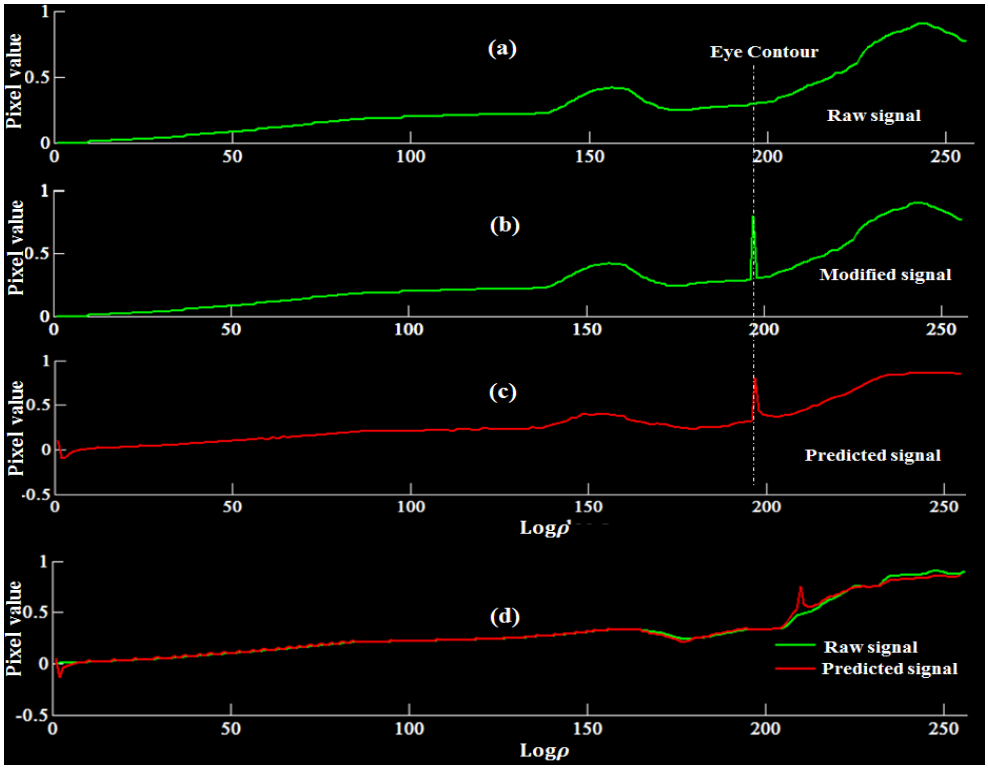


Fig. 5.15 (a) Training strategy using one pixel profile. (a) Raw signal. (b) Modified signal. (c) Predicted signal. (d) Predicted signal of different profile.

the locations of the eye contour, obtained by manually labeling operation, have been modified to take 0.8. It has not been changed to 1 to avoid the saturation of the activation function that may lead the network to be unstable.

In detection stage, once a new image is given, the image is converted into grayscale, rescaled to the range [0 0.9] and Log-polar transformed. The Log-polar transformed image is then converted into a series of N=16 eye profiles. The recurrent network then tries to predict/detect the eye contour locations in these profiles.

5.11.2. Results

Figure 5.14 shows a Log-Polar eye image with some pixel profiles taken at different θ. Figure 5.15(a) shows particularly the eye profile, green signal, at $\theta = 22°$. This profile is called raw signal. The raw signal is modified at the eye contour location to have a pulse with value of

0.8 as illustrated in Fig. 5.15(b). The modified signal is used to train the recurrent network in order to predict one future step. Once learning operation is done, the raw signal is presented to the trained network. The network predicts one value ahead and generates correctly a pulse at the location of the eye contour as shown in Fig. 5.15(c).

Figure 5.15(d) shows the eye profile at $\theta = 34°$ of the same eye image. This profile is presented to the trained network and a predicted signal is generated. The predicted signal possesses a sharp pulse that is expected to indicate the location of the eye contour in the eye profile.

Figure 5.16(a) shows a series of N=16 eye profiles. These eye profiles represent an eye image sampled according to the modeling location strategy (sampling rate S=22). The series is modified to have sharp pulses at the actual locations of the eye contour in the image as shown in Fig. 5.16(b). The modified series is considered as a training signal and used to train the network.

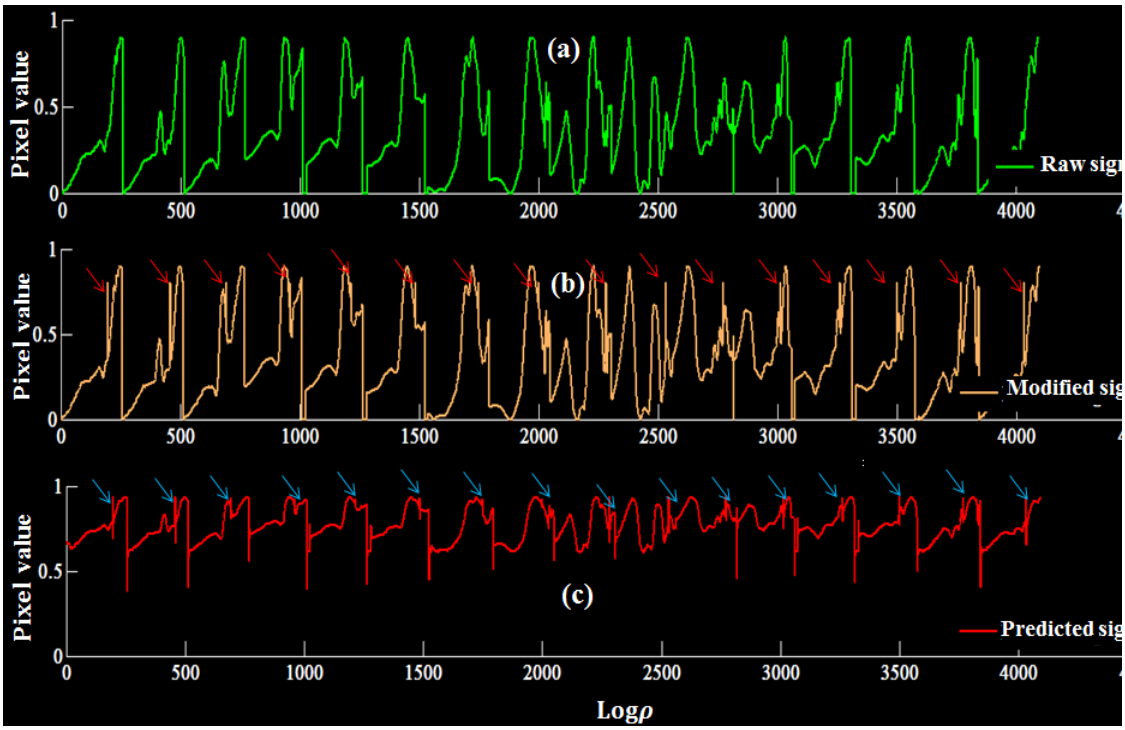


Fig. 5.16 (a) Training strategy using a set of pixel profiles. (a) Raw signal. (b) Modified signal. (c) Predicted signal.

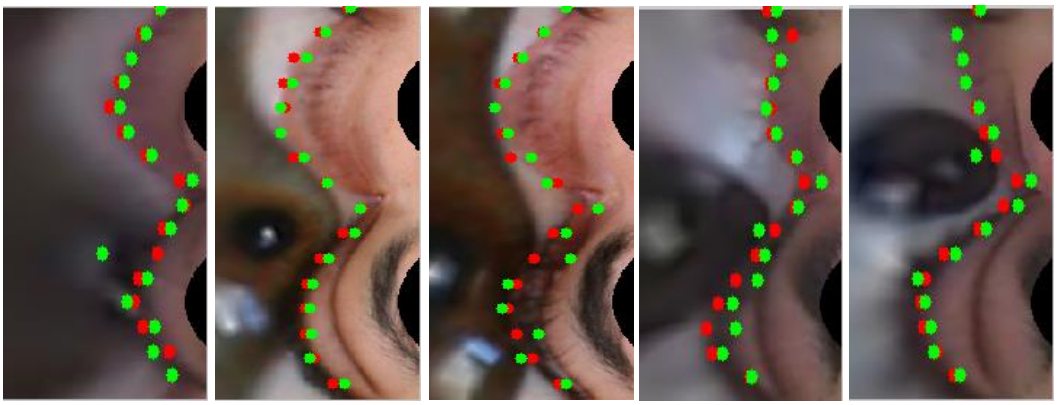


Fig. 5.17 Eye shape detection. The green shapes represent the detected shape by the trained network. The red shapes represent the exact eye shapes that manually labeled.

Table 1. Comparison between ASM, DT and the proposed method.

Method	Accuracy%	Iteration	Detection Time(msec)
ASM	69	20	85
DT	85	12	240
Proposed Method	90	-	65

After training, the raw series, which has no sharp pulses, is presented and a predicted signal is obtained as illustrated in Fig. 5.16(c).

Figure 5.17 shows some testing eye images with different scales, rotation angles and lighting conditions. The corresponding exact eye shapes are manually labeled by N=16 points as illustrated in red points. The predicted eye shapes by the trained recurrent neural network are shown in green points.

Table 1 concludes the results of eye shape detection using the proposed method, ASM and DT. 300 eye images have been randomly taken from the two databases for this experiment. The table includes three comparative components; detection time, accuracy obtained and number of iterations.

5.11.3. Discussion

The eye profile is simple and its nonlinearity is not so high. The structure in Fig. 5.13 enables the network to learn very high nonlinear functions. This powerful performance is governed by the recurrent connections between the neurons themselves and each other. Figure 5.15 proves this point by showing the predicted signal of an eye profile. On can observe that there is a difference in magnitude between the training and predicted signals especially at the actual location of the eye contour. This difference doesn't affect the results because the most significant information is the location of the rapid and sudden pixel change in the predicted signal. As long as the pixel values of an eye profile change smoothly, a sharp change is guaranteed to be generated by the recurrent neural network one time along the eye profile.

On the other hand, the difference between two eye profiles in the same eye image is proportional to $\Delta\theta$ as highlighted in Fig. 5.14(b). Based on learning the raw eye profile in Fig. 5.15(a), the network was able to predict correctly the location of the eye contour which is 6 degrees distant, Fig. 5.15(d).

According to the conducted experiments, the recurrent network has provided low performance and almost failed to predict the correct locations of the eye contour when the distance of an eye profile is approximately larger than 10 degrees from the training eye profile. Therefore, the sampling rate *S* must be selected carefully.

In order to make the network more flexible and robust to deal with different eye profiles of various eye images, we trained the network using a series of eye profiles that extracted from different eye images at fixed sampling rate *S* as shown in Fig. 5.16(a). We guess that this strategy enables the network to learn various patterns of the pixel profiles. Although, it enables the network to learn the frequency of the sharp pulses. Consequently, RNN models the eye shape as well. Therefore, the performance has surprisingly improved as illustrated in Fig. 5.16.

Figure 5.17 demonstrates the performance of the proposed method to detect the exact eye shapes of the testing images. The accuracy is high and the detected points almost match the manually labeled points (exact eye shape).

The proposed method outperforms ASM and DT as highlighted in table 1. ASM and DT use an iterative strategy to detect the eye shape whereas the proposed method needs to achieve constant steps. This reduces significantly the detection time. The accuracy yielded is high because of using the prediction ability of the recurrent neural network to detect the eye contour as well as its generalization capabilities to deal with new eye profiles.

We don't claim that the presented structure of the network is the best for modelling and predicting the eye profiles. On the other hand, we tried to use the same structure as a feedforward network without any recurrent connections. The prediction ability was very low. Moreover, recurrent neural networks have widely been employed in neuro-control applications to be learned in real time. Therefore, the weights of the learned recurrent network can be updated online to track the eye shape in real time. This work is left to be investigated in future.

5.12. Conclusion

As it has been mentioned in the previous chapter, processing the eye shape in the Log-polar domain is the baseline for implementing any detection technique. On the other hand, searching on the eye contour location in a pixel profile can be achieved using many techniques.

In this chapter, a brief introduction on neural network has been given. Feedfroward neural network has been investigated in details for initializing a good understanding of the concept of how a neural network can learn different type of function.

The eye profile is considered as a series that changes at each proceeding step along log*p*

axis. Therefore, a recurrent neural network has been presented as a powerful tool of prediction. The recurrent network has been first trained to predict the location of the eye contour in a pixel profile. This approach has showed a weak performance in terms of generalization. The network was not able to detect correctly the eye contour location in different profiles. Therefore, the learning strategy has been changed to train the network using a set of eye profiles from different images. The performance has been significantly improved and the network has modeled the change of pixel values in the eye profiles as well as the change of the eye shape.

The results have verified that the proposed method performs high accuracy with noticeably reducing of detection time compared to the conventional methods.

Chapter 6

6. Thesis Summary and Common Conclusion

The human eyes provide very significant information to be extracted and analyzed. This information can be utilized to solve various real time problems and to facilitate achieving many actions that human does using hands. Therefore, eye is a very promising and interesting interactive element. We expect that the relevant applications in real time systems will be rapidly grown in the close future. Consequently, we took this fact into account and turned our attention to propose and improve some eye feature extraction techniques. On the other hand and in order to make this thesis more valuable, some mathematical issues have been explained to simulate the human visual system. Therefore, a brief description on this system has been presented earlier with referring frequently to the related concepts through the thesis context.

In this thesis, eye features are described by the eye contour which consists of two eye corners, upper eyelid and lower eyelid. These features are combined into a common representation called eye shape. The eye shape consists of a set of points that distributed equally on the eye contour. This approach of the eye contour representation is very efficient to provide more flexibility of general use according to the user's target application.

A lot of difficulties can be faced in order to detect the eye shape as well as vast parameters can influence the detection operation. We constrained these parameters and difficulties into some key points that are most important according to our opinion, or at least, have the higher priority to be solved in the implementation process. The problems are as follows:

- Different lighting conditions: human eye has a very complex texture. Light influences this texture by changing the pixel values in the digital image domain. On the other hand, different skin color can be thought of as a part of the problem.
- Different scale: human eye has a complex structure to be studied geometrically. The eye shape provides somehow a common representation on this structure. The scale is described as the change of the eye state or the distance from the camera. This problem is related to how the pixels of the eye contour change in their positions individually.
- Different rotation angles: this problem is also related to the eye structure but it can be described as different appearance of the eye at different angles. The eye is considered as a rigid component and hence the positions of pixels of the eye contour change unitedly and collectively.
- Different translations: translation can be thought of as rotating the human face around Y axis in 3D system. As long as we deal with 2D digital image domain, the change of this parameter can be approximated to change in scale. On the other hand, the eye area "eye image" is detected using some object detector [72]. Due to some unexpected conditions such as noise, the detector may fail in locating the eye at the image center perfectly. Therefore, some slight translations along X and Y axes may occur.

Active Shape Model was the first technique that put in the picture to be investigated and applied for eye shape detection. The standard ASM has performed very low accuracy and consumed long detection time. This is because of suffering and low capabilities to deal with all above problems.

We tried to enhance the performance in two phases; initial estimation and searching operation. In the initial estimation phase, ASM uses a global shape called mean eye shape to explain the general eye state. Therefore, ASM considers the mean shape as the initial estimation

of the eye contour in testing images. This means that ASM is a blind technique because it has no prior knowledge on the eye structure. Thus we provided some intelligence to this technique using object recognition based PCA. The object recognition technique tries to understand the eye geometrical structure in the testing images. A testing eye image is compared with a set of eye images in a database. Based on the comparison result, an eye image that best describes the eye structure in the testing image is recognized. The exact eye shapes of the images in the database are known and manually labeled. Therefore, the eye shape of the recognized image is fitted to the testing eye image as an initial estimation. Hence, if the database contains *M* images, that means one of the corresponding *M* exact eye shapes will be used as an initial estimation instead of using mean eye shape for each testing image.

In the second phase, we enhanced the local representation of the pixel values around the eye shape points using sub-image strategy. ASM uses mean profile vector as a common descriptor of the pixel values around the eye shape points. This vector contains only a few pixels and lies in one direction. It provides a weak performance in order to detect the eye contour location accurately. In contrast, a sub-image lies in two directions and contains larger number of pixels. It provides significant information on the local structure at the eye contour as well as it describes the change of pixel values more effectively. As long as a sub-image contains a large number of pixels, the searching operation consumes extra time comparing to the use of mean profile vectors. This problem has been efficiently solved using PCA capabilities in terms of data reduction. Moreover, searching on the eye contour location in principal component space has achieved better performance against the change of lighting conditions.

The improved ASM outperformed the standard ASM in terms of accuracy, detection time and stability. The use of PCA has reduced the error of the initial estimation of the eye contour around 50% compared with the use of the mean eye shape. On the other hand, the use of

sub-image approach has enabled the searching operation to be very smooth, increased the accuracy of the final detection and decreases the number of iterations around 40%. Consequently, the detection time has been reduced to 60 msec compared to around 200 msec consumed by standard ASM.

ASM has turned our attention into a very considerable issue that can affect the performance of any real time algorithm negatively. This issue is the use of an iterative strategy in order to search on eye shape points. During the evaluating of ASM performance, we observed that each testing image consumes different number of iterations depending on the eye structure and state. Therefore, the corresponding detection time is not fixed. If any technique that uses an iterative strategy (ASM, AAM,DT) is employed in a real time system, it is not guaranteed that the system will continually tracking the eye contour because ASM may stuck and run in an infinite process loop. Consequently, the algorithm stops to process the eye shape through time. Moreover and instead of the proposed improvements, ASM is still a risky choice due to the high freedom degree of changing of the eye contour parameters in Cartesian coordinate system.

Based on the above observations, we turned to think about implementing a more robust technique. Unlike the other approaches of implementing or involving many techniques to enhance the image texture or to highlight the eye features by applying some filters or high level image processing algorithms, we thought of changing the processing domain before applying such techniques. We considered perfect appearance conditions of the eye in the image that is *located in the image center (no translations), has an open state with a good scale and has a zero rotation angle (the two eye corners are perfectly aligned horizontally).*

From a theoretical point of view, if the origin of Cartesian system is shifted to be at the image center, the image can be fully scanned by a line originates from the center and rotates

around for $360°$. The scanned pixels at each particular angle can be remapped into a vertical stack. This stack creates a new image whose vertical axis can be named as θ axis and the horizontal axis is the ρ axis. Obviously, this mapping strategy expresses Polar coordinate system. The scanned line intersects with the eye contour at each θ . Therefore, each row in the polar "image represents the eye contour in a pixel. The location of this pixel changes from row to row to fully represent the eye contour along θ axis.

A question can be highlighted is "what happens if the eye rotates around the center in Cartesian coordinate system?" Intuitively, the image is scanned again by the line and the eye contour will be fully represented along θ axis. The difference is only in changing the eye contour location in each row (the change occurs in ρ coodinates). In other words, the change of rotation angle is transformed into translation along θ axis. *Therefore, it is not necessary for the eye to appear at zero angle*.

From a practical point of view, the above approach of using polar domain is not appropriate because the pixels that near from the image center are not sufficiently represented with geometrical details compared to those far from the center. Accordingly, suppose the eye, or part of it, is located at the area near the center due to small scaling such as far distance from the camera or nearly close eye state. A very large shift in ρ coordinates will occur as well as the eye contour will lose its geometrical appearance and appears as a vertical line along θ axis in the polar domain. Moreover and if the eye appears in a good scale, the description of the local structure will not equally presented. For example, the area of the eye corners (which is far from the center) will be presented with more geometrical details compared to the middle regions of the eyelids (which are much closer to the center).

The above problem can be overcome by resizing the pixel representation according to the distance from the image center. This procedure represents the geometrical details of any area

equally regardless its distance from the image center. Therefore, Log-polar transform is used. In Log polar domain, the change of scale is converted into change in translation along Logρ axis with preserving the geometrical details. *Therefore, it is not necessary for the eye to appear in a good scale as well as the eye is not necessary to be in a wide open state.*

Suppose that the eye is not located at the image center perfectly and shifted along X and Y axes. As long as the eye contour surrounds the image center and the geometrical structure of any region is represented equally regardless the distance from the image center, the change of translation is transformed by the same amount into translation along Logρ axis. *Therefore, it is not necessary for the eye to appear perfectly at the image center.*

Based on the previous interesting discussion and because the eye contour exists along θ axis, *the task of detecting the x,y coordinates of the eye contour in Cartesian coordinate system is simplified into just searching on Logρ coordinates in the Log-polar domain*. In order to achieve this task, Principal Component Analysis has been used to describe the variance between RGB eye color channels. The eyeball consists of sclera, iris and pupil. These components can be approximately described by two fixed colors; white and black. On the other hand, skin around the eye is influenced by light conditions and changes gradually. Therefore, the variance between the eye color channels in the eyeball area is very small whereas it has larger values in the skin area. The first eigenvector which has the largest eigenvalue represents the direction of the most variation in a data. Consequently, it is the best way to explain the variance between the eye color channels and it is called Log-polar Eigeneye Image (LPEI).

In a LPEI, the eye region is approximately represented by black color whereas the skin changes gradually in grayscale. Therefore, the pixel profile of any row can be described uniformly. In more details, the pixel profile can be classified into three areas; zero (eyeball area), information (skin area) and max (included in the information area and contains the maximum

pixel values). Due to the topological eye structure, information area is divided into two subsections. The first subsection has ascending order of pixel values and lies from the beginning of the info area until the max area. The second subsection has descending order of pixel values and lies from the max area until the end of the row.

Obviously, the eye contour is supposed to be located at the beginning of the information area. Therefore, cross correlation based template matching has been utilized to identify the ascending order of the pixel values at the beginning of the first subsector. In addition, an eye shape model based-PCA has been implemented in Log-polar domain to correct the deformations of the detected eye shape.

Log-polar transform has been proven as the best domain to detect the eye contour and features. It guarantees that the geometrical details of the eye structure are described equally regardless the distance from the image center in Cartesian coordinate system. The change of scale is converted into translation along $\log\rho$ axis whereas the change of rotation is converted into translation along θ axis. In addition, Log-polar transform represents the eye corners in the same characteristics regardless the orientation. Based on these properties, the task of detection x,y coordinates of the eye contour is converted into only searching on $\log\rho$ coordinates in Log-polar domain.

PCA has been proven as a significant tool in describing the variance between eye color channels against lighting conditions. It provides a uniform classification of the change of pixel values in any row in an LPEI.

Based on all above properties, a very robust performance has been achieved with very high accuracy around 95%. The proposed method does not use an iterative strategy and it attains fixed steps. This means that the detection time is fixed as well. Detection time has been significantly reduced to only 40 msec. Thus, the proposed method meets all requirements to be

employed in real time system. As long as the eye contour exists along θ axis, the description of different eye states is contained in log ρ coordinates. Therefore, a very robust eye shape model has been obtained. The deformations of the detected eye shape occur in log ρ coordinates only and hence the correction process using the eye shape model was very effective.

As the Log-polar has been proven as a baseline for implementing any eye shape detection technique, the development door is still open in terms of searching on the log ρ coordinates of the eye contour. For achieving this task, two templates have been used to describe the patterns of pixel values in LPEI (first subsection). This approach has performed very good results. On the other hand, patterns can be changed out of these templates. One solution is to increase the number of templates but the same issue of changing patterns out of the templates should be concerned. Thus generalization is very necessary to be incorporated in the implementation.

Neural networks have a very robust generalization capability. For extremely utilizing neural network properties, we tried to employ the prediction capability as well. Therefore, Recurrent Neural Network (RNN) has been used. The strategy is to consider a pixel profile at any θ as a signal namely; raw signal. As long as the change of pixel values along a pixel profile is very smooth (has no sudden or rapid change of pixel values), the signal is modified to have a pulse (a sudden change of pixel value) at the location of the eye contour. RNN is then trained using the modified signal in order to predict one step ahead. According to the experimental results, the trained RNN given the raw signal was capable to remember the correct location of the eye contour by generating a pulse. On the other hand, RNN was incapable to predict properly the location of the eye contour when a pixel profile differs more than 10 degrees from the θ coordinate of the training pixel profile. This is because of the change in log ρ locations of the eye contour.

In order to increase the flexibility of RNN to deal with various pixel profiles at different lighting condition, an eye image is considered as a series of N=16 pixel profiles. These pixel profiles are taken according to fixed θ coordinates in the Log-polar domain. A set of 50 modified images (have sudden change of pixel values at the eye contour) have been used for training the RNN and the performance is surprisingly and significantly improved. This is because besides of modeling the change of pixel values in the modified pixel profiles, RNN has learned the frequency of the eye contour locations in N pixel profiles (the eye image). Therefore, RNN models the eye shape as well.

The proposed method based RNN has achieved high accuracy (around 90%) at low detection time (around 70 msec). If these results are compared with those obtained by the proposed method based LPEI, the latter method achieves better results.

On the other hand, it should be highlighted that the eye images have been processed by RNN in grayscale without using PCA to explain the variance between color channels. If RNN were trained using LPEI, we would expect that RNN would achieve a very robust performance with very high accuracy. Moreover, RNN has modeled both the eye shape and the change of pixel values. Therefore, the results have been achieved without using an additional eye shape model based PCA to refine the $\log \rho$ deformations. Consequently, the use of neural networks is very promising to achieve very high accuracy. These improvement issues are left to the future plan to be investigated.

This thesis has addressed and highlighted many problems and issues that should be considered in implementing an eye shape detection technique. The relevant solutions have been presented and discussed. Some of these solutions can be applied individually for solving different image processing problems. The importance of eye features has been highlighted with different

applications. Log-polar transform has been deeply investigated and proven as the best domain to detect the eye contour and features. Different applications of PCA have been mentioned in modeling eye shape, understanding the eye topological structure, comparing two sub-images based on data reduction property and interpreting the variance between RGB eye color channels. The basic concepts of neural networks have been studied and detailed in terms of generalization and prediction capabilities.

On the other hand, the above mentioned techniques can be differently incorporated to implement new techniques for eye shape detection and investigate/evaluate the corresponding behaviors such as applying the ASM searching strategy to detect the eye contour location in Log-polar domain, applying feedforward neural network for achieving the same task with using prediction property, investigating the improvement in accuracy of an eye shape model based PCA to refine RNN output deformations,…,etc.

Moreover, some applications can be inspired by investigating some particular key points such employing eye representation in Log-polar domain to construct a 3D eye model, calculating gaze point, generating different eye shapes and eye textures by changing $\log \rho$ coordinates, computing the eye opening angle, extracting more eye features such as pupil location and eyebrow shape, iris finger-print identifying, calculating the distance between the camera and the eye based on translation property in Log-Polar domain…etc.

Therefore, this field is still challenging, interesting and promising…

Bibliography

[1] Gorodinichy, D. and Malik, S: "Use Nose: Your Nose as a Mouse - a New Technology for Hands-Free Games and Interfaces", *Proc. International Conference on Vision Interface* (2002), 354-361.

[2] Zhu, J. and Yang, Y.: "Subpixel Eye Gaze Tracking", *Proc. International Conference on Automatic Face and Gesture Recognition,* (2002), 131-136.

[3] Chang, Y., Lwin, Y. and Yamamoto Y.: "Sensor-based Trajectory Planning Strategy for Non-holonomic Mobile Robot with Laser Range Sensors", *IEEE International Symposium on Industrial Electronics,* (2009), 1755–1760.

[4] Hoon, C. and Wan, Y.: "Improving the Classification of Landsat Data Using Standardized Principal Component Analysis", *KSCE Journal of Civil Engineering* , Vol. 7, No. 4, (2003), 469-474.

[5] Wiecek B., Peszynski-Drews C., Wysocki M., Jakubowska T., Danych R. and Zwolenik S.: "Advanced Methods of Thermal Image Processing for Medical and Biological Applications", *Thermography and Lasers in Medcine*, (2003), p 29-40.

[6] Frank, J. D., Kunz T. H., Horn J., Cleveland C., and Petronio S.' "Advanced Infrared Detection and Image Processing for Automated Bat Censusing", *Proc. SPIE 5074, Infrared Technology and Applications XXIX*, (2003), 261–271.

[7] Genzhi, Y., Yebin, L., Nils, H., Xiangyang, J., Qionghai, D. and Christian, T., "Performance Capture of Interacting Characters with Handheld Kinects", *Proc. European Conf. Computer Vision*, (2012), 828-841.

[8] Schlegel, D., Chen, A., Xiong, C., Delmerico, J., and Corso, J.: "Airtouch: Interacting with computer systems at a distance", *In Applications of Computer Vision (WACV), IEEE Workshop on*, (2011), 1-8.

[9] Alan, Y., Peter, H., and David, C.: "Feature Extraction from Faces Using Deformable Templates", *International Journal of Computer Vision*, Vol. 8, No. 2, (1992), 99–111.

[10] Timothy, F. C., Gareth, J. E. and Christopher, J. T.: "Active Shape Models-Their Training and Application Computer", *Vision and Image Understanding*, Vol. 61, No. 1, (1995). 38-59.

[11] Kwon, Y. M., Jeon, K. W., Ki, J., Shahab, Q. M., Jo, S. and Kim, S. K.: "3D Gaze Estimation and Interaction to Stereo Display", *Int. J. Virtual Reality*, Vol. 5, No. 3, (2006), 41 -45.

[12] Beymer, D. and Flickner, M., "Eye Gaze Tracking Using an Active Stereo Head", *IEEE Conference in CVPR03*, 2003.

[13] Chen, J., Tong, Y., Gray, W. and Ji, Q.: A Robust 3d Eye Gaze Tracking System Using Noise Reduction", *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, (2008), 189–196.

[14] Tofiq, K. and Shafagat, M.: "About a Method of Recognition of Race and Ethnicity of Individuals Based on Portrait Photographs", *Intelligent Control and Automation*, Vol.5, No.3, (2014), 120-125.

[15] Biedert, R., Dengel, A., Buscher, G. and Vartan, A.: "Reading and Estimating Gaze on Smart phones", *ETRA*, (2012), 385-388.

[16] Biedert, R., Dengel, A., Buscher, G. and Vartan, A.: "Reading and Estimating Gaze on Smart phones", *ETRA*, (2012), 385-388.

[17] Zhu, Z., and Ji. Q.: "Eye and Gaze Tracking for Interactive Graphic Display", *Machine Vision and Applications*, Vol. 15, (2004), 139–148.

[18] Barea, R., Boquete, L., Mazo, M., Lopez, E. and Bergasa, L.M.: "EOG Guidance of a Wheelchair Using Neural Networks", *International Conference on Pattern Recognition* (2000), 668–671.

[19] Tian, Y., Kanade T., and Cohn, J., "Dual-State Parametric Eye Tracking", *Proc. Int, l Conf. Face and Gesture Recognition*, (2000), pp. 110-115.

[20] Qiang, J., Zhiwei Z., and Peilin L.: "Real-time Nonintrusive Monitoring and Prediction of Driver Fatigue", *IEEE Trans. Vehicles Technology*, Vol. 53, No. 4, (2004), 1052 -1068.

[21] Takatsugu, H., Kenji, M., and Kazuya, T.: "Driver Inattention Detection based on Eye Gaze-Road Event Correlation", *International Journal of Robotics Research*, (2009), 774-801.

[22] Linda, S., Robert, J., and James T.: "Evaluation and Analysis of Eye Gaze Interaction", *NRL Report NRL/FR/5513–01-9990*, Naval Research Laboratory, Washington, DC, 2001.

[23] Young, L. R. and Sheena, D.: "Survey of Eye Movement Recording Methods", *Beh. Res. Methods Instrum.*, Vol. 7, No. 5, (1975), 397–429.

[24] Kass, M., Witkin, A. and Terzopoulos, D.: "Snakes: Active Contour Models", *International Journal of Computer Vision* (1987), 321-331.

[25] Lanitis, A., Taylor, C., Cootes, T.: "Automatic Interpretation and Coding of Face Images Using Flexible Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (997), Vol. 19, No. 7, 743-756.

[26] Lanitis, A., Taylor, C., Cootes, T.: "Recognising Human Faces Using Shape and Grey-Level Information", *Third International Conference on Automation, Robotics and Computer Vision*, (1994).

[27] Thai, H. L., Truong N. V.: "Face Alignment Using Active Shape Model and Support Vector Machine", *International Journal of Biometrics and Bioinformatics*, (2011), Vol. 4, No. 6, 224–234.

[28] Aditya, A. and Stephanie S.: "Active Shape Models for Effective Iris Segmentation", *SPIE 6202: Biometric Technology for Human Identification III,* 2006, 6202:H1–H10.

[29] Liu, N. and Lovell B. C.: "Hand Gesture Extraction by Active Shape Models", *Proc. Digital Image Computing: Techniques and Applications*, (2005), 59-64.

[30] Cootes, T., Taylor C., Hill, A. and Halsam J.: "The Use of Active Shape Models for Locating Structures in Medical Images", *Proceedings of th 13th International Conference on Information Processing in Medical Imaging*, (1993), 33-47.

[31] Timothy, F. C., Gareth, J. E., and Christopher, J. T.: "Active Appearance Models", *European Conference on Computer Vision*, (1998), 321-331.

[32] Drysdale, R.L. and Lee, D. T., "Generalized Voronoi Diagrams in the Plane", *Proc. 16th Annual Allerton Conference on Communications, Control and Computing*, (1978), 833–842.

[33] Belongie, S., Malik, J., and Puzicha, J.: "Shape Matching and Object Recognition Using Shape Contexts", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (2002), Vol. 24, 509-522.

[34] Berg, A., Berg, T., and Malik, J.: "Shape Matching and Object Recognition using Low Distortion Correspondences", *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, (2005).

[35] Lin, F, and Cohen, W. W.: "Power Iteration Clustering", *ICML*, (2012).

[36] Pereira, S. and Pun, T. "Robust Template Matching for Affine Resistant Image Watermarks", *IEEE Trans. Image Processing*, (2000), Vol. 9, 1123-1129.

[37] Land, M.F.: Structure of the Retina of the Principle Eyes of Jumping Spiders (Salticidae, Dendryphantinae) in Relation to Visual Optics", *J. exp. Biol.*, (1969), Vol. 51, 443–470.

[38] DiCarlo, J., Zoccolan, D. and Rust, N.: "How Does the Brain Solve Visual Object Recognition", *Neuron*, (2012), Vol. 73, 415-434.

[39] Field, D.J., Hayes, A., and Hess, R.F.: "Contour Integrations by the Human Visual System: Evidence for a Local "Association" Field", *Vision Res.*, (1993), Vol. 33, 173-193.

[40] Weiman, C., Chaikin, G.: "Logarithmic Spiral Grids for Image Processing and Display", *Computer Graphics and Image Processing*, (1979), Vol. 11:197–226.

[41] Acton, S. T., Bovik, A. C. and Crawford, M. M.: "Anisotropic Diffusion Pyramids for Image Segmentation", *IEEE Int. Conf. Image Processing*, Vol. 3, (1994), 478 -482.

[42] Ding, L. and Goshtasby, A.: "On the Canny Edge Detector", *Pattern Recognition*, Vol. 34, No. 3, (2001), 721–725.

[43] Felsberg, M.: "Low-Level Image Processing with the Structure Multivector", *Doctoral Dissertation*, Univ. of Kiel, Germany, 2002.

[44] Ballard, D. H.: "Generalizing the Hough Transform to Detect Arbitrary Patterns", *Pattern Recognition*, Vol. 13, No. 2, (1981), 111 -122.

[45] Kontsevich, L. L.: "Pairwise Comparison Technique: A Simple Solution for Depth Reconstruction", *Journal of Optical Society*, Vol. 10, No. 6, (1993), 1129 -1135.

[46] Martin, C. and Schovanec, L.: "Muscle Mechanics and Dynamics of Ocular Motion", *J. Math. Syst., Estimat. Control*, Vol. 8, (1998), 1 -15.

[47] Duke-Elder, W. S. "Textbook of Ophthalmology", Vol. 1, *The C.V. Mosby Co. St. Louis*, (1940), Chapter III.

[48] Carpenter, R. H. S.: "Movements of the Eye", Pion, London, (1977), 116-127.

[49] Milborrow, S. and Nicolls, F.: "Locating Facial Features with an Extended Active Shape Model", *International European Conference on Computer Vision*, (2008).

[50] Liu, N. and Lovell, B.C.: "Hand Gesture Extraction by Active Shape Models", *Proceeding Digital Image Computing: Techniques and Applications*, (2005), 59-64.

[51] Baumberg, A. and Hogg, D. "An Efficient Method for Contour Tracking Using Active Shape Models", *Proceeding Workshop Motion of Non-rigid and Articulated Objects*. Los Alamitos, California, (1994).

[52] Duta, N. and Sonka, M.: "Segmentation and Interpretation of MR Brain Images: An Improved Active Shape Model", *Medical Imaging, IEEE Transactions on*, Vol. 17, (1998), 1049 -1067.

[53] Yan, S.C., Liu, C., and Zhang, H.J.: "Face Alignment Using Texture Constrained Active Shape Models", *Image and Vision Computing*, (2003), 69-75.

[54] Aldibaja. M., and Suzuki, S.: "An Adaptive Active Shape Model for Eye Shape Detection," *Proceedings of the 2013 International Symposium on Advanced Science and Technology in Experimental Mechanics, Sendai*, 2013.

[55] Gower, J. C.: "Generalized Procrustes Analysis", (1975), 33–31.

[56] Bookstein, F. L., "Landmark Methods for Forms without Landmarks: Localizing Group Differences in Outline Shape", *Medical Image Analysis*, Vol. 1, No. 3, (1997), 225–244.

[57] Jackson, E. A.: "User's Guide to Principal Components", (1991), 1-25.8

[58] Cootes, T.F., Taylor, C.J. and Hill, A.: "The Use of Active Shape Models for Locating Structures in Medical Images", *Proceeding 13th International Conference on Information Processing in Medical Imaging*, (1993), 33-47.

[59] Marsaglia, G.: "Evaluating the Normal Distribution" *Journal of Statistical Software*, Vo. 11, No. 4, (2004), 1-11.

[60] Cootes, T.F., Taylor, C.J. and Lanitis, A.: "Active Shape Models: Evaluation of a Multi- Resolution Method for Improving Image Search", *Proceeding of The British Machine Vision Conference*, (1994), 327-336.

[61] Sukno, F., Ordas, S., Butakoff, C. and Frangi. A.F.: "Active Shape Models with Invariant Optimal Features IOF-ASMs", *Proceeding of AVBP*, (2005), 365-375.

[62] Ginneken, B.V., Alejandro, F. F., Joes, J. S., Bart, M. H. and Max, A. V.: "A Non-Linear Gray-Level Appearance Model Improves Active Shape Model Segmentation", *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 2001, 205-212.

[63] Mahoor, M. H., Abdel-Mottaleb, M. and Ansari, N. A.: "Improved Active Shape Model for Facial Feature Extraction in Color Images", *Journal of Multimedia*, Vol. 1, No. 4, (2006), 21-28.

[64] Yunshu, F., Zhonghua, F. and Zhang Y.: "Facial Feature Extraction Based on a Modified Active Shape Model", *Application Research Of Computers*, Vol. 11, (2006), 255-257.

[65] Baochang, Z., Wen, G., Shiguang S. and Wei W.: "Constraint Shape Model Using Edge Constraint and Gabor Wavelet Based Search", *Proceeding of International conference on Audio- and Video-based Biometric Person Authentication*, (2003), 52-61.

[66] Rogers, M. and Graham, J.: "Robust Active Shape Model Search," *Proceeding European Conference on Computer Vision*, (2002), 517-530.

[67] Smith, L. I.: "A Tutorial on PCA", 2002: Cornell University, *Available online at http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf.*

[68] Grauman, K. and Leibe, B.: "Visual Object Recognition", (2011).

[69] Hirano, Y.: "Industry and Object Recognition: Applications, Applied Research and Challenges", *Lecture Notes in Computer Science*, Spinger, (2006), 49-64.

[70] Loutfi, A., Broxvall, M., Coradeschi, S. and Karlsson, L.: "Object Recognition: A New Application for Smelling Robots", *Robotics and Autonomous Systems*, Vol. 52, (2005), 272-289.

[71] Turk, M. and Pentland, A.: "Face Recognition Using Eigenfaces", *Proceeding IEEE Conference on Computer Vision and Pattern Recognition*, (1991), 586-59.

[72] Khac, C.N. and Park, H.: "Face Detection Using Variance Based Haar-Like Feature and SVM", *World Academy of Science, Engineering and Technology*, Vol. 60, (2009).

[73] Araujo, H. and Dias, J. M.: "An Introduction to the Log-polar Mapping", *IEEE Second Workshop on Cybernetic Vision*, (1996), 139-144.

[74] Abadpour, A. and Kasaei, S.: "An Efficient PCA-Based Color Transfer Method", *Journal of Visual Communication and Image Representation*, Vol. 18, (2007), 15–34.

[75] Giachetti, A., Campani, M. and Torre, V.: "The Use of Optical Flow for Road Navigation", *IEEE Transactions on Robotics and Automation*, Vol. 14, (1998), 34-48.

[76] Stasse, O., Kuniyoshi, Y. and Cheng, G.: "Development of a Biologically Inspired Real-time Visual Attention System", *Proceedings in Biologically Motivated Computer Vision*, Vol. 1811, (2000), 150-159.

[77] Bernardino, A. and Santos-Victor, J.: "A Binocular Stereo Algorithm for Log-polar Foveated Systems", *Proceedings in Biologically Motivated Computer Vision*, Vol. 25, (2002), 127-136.

[78] Traver, V. J. and Pla, F.: "The Log-polar Image Representation in Pattern Recognition Tasks", *Pattern Recognition and Image Analysis, Proceedings*. Vol. 2652, (2003), 1032-1040.

[79] Zokai, S. and Wolberg, G.: "Image Registration Using Log-polar Mappings for Recovery of Large-scale Similarity and Projective Transformations," *IEEE Transactions on Image Processing*, Vol. 14, (2005), 1422-1434.

[80] Tunley, H. and Young, D.: "First Order Optical Flow from Log-polar Sampled Images", *Proceeding of 3rd European Conference on Computer Vision*, Stockolm, Sweden, (1994), 132–137.

[81] Dias, J. A., Paredes, C. and Batista, J.: "Optical Normal Flow Estimation on Log-Polar Images. A Solution for Real-time Binocular Vision," *Real-Time Imaging*, Vol. 3, No. 3, (1997), 213–228.

[82] Stegmann, M. B. and Gomez, D. D.: "A Brief Introduction to Statistical Shape Analysis", *Technical Report*, Technical University of Denmark, 2002.

[83] Cruse, H.: "An Application of the Cross-correlation Coefficient to Pattern Recognition of Honey Bees", *Kybernetik*, Vol. 15, (1974), 73–84.

[84] Stefano, L. D. and Mattoccia, S.: "Fast Template Matching Using Bounded Partial Correlation", *Machine Vison Applications*, Vol. 13, No. 4, (2003), 213-221.

[85] Zitova, B. and Flusser, J.: "Image Registration Methods: A Survey," *Image and Vision Computing*, Vol. 21, (2003), 977-1000.

[86] Tsai, D. and Lin, C.: "Fast Normalized Cross Correlation for Defect Detection". *Pattern Recognition Letters*, Vol. 24, No. 15, (2003), 2625–2631.

[87] Berberidis, K. and Karybali, I.: "A New Efficient Cross-Correlation Based Image Registration Technique with Improved Performance", *11th European Signal Processing Conference*, (2002).

[88] Schatzman, J. C.: "Accuracy of the Discrete Fourier Transform and the Fast Fourier Transform", *SIAM Journal on Scientific Computing*, Vol. 17, No. 5, (1996), 1150 -1166.

[89] Proenca, H., Filipe, S., Santos, R., Oliveira, J. and Alexandre, L. A.: "The UBIRIS.V2: A Database of Visible Wavelength Iris Images Captured on-the-Move and at A-distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 32, (2010), 1529–1535.

[90] [Online]. http://iris.di.ubi.pt/

[91] Medsker, L., and Jain L.: "Recurrent Neural Networks Design and Applications", *CRC International Series on Computational Intelligent* (2000).

[92] McCulloch ,W.S. and Pitts,W.: "A Logical Calculus of Ideas Immanent in Nervous Activity," *Bull. Mathematical Bio physics,* Vol. 5, (1943), 115-133.

[93] Minsky, M. and Papert, S.: "Perceptrons: An Introduction to Computational Geometry" *MIT Press, Cambridge, Mass.,* (1969).

[94] Hopfield, J.J.: "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *in Roc. Nat'l Academy of Sciences,* USA .Vol, 79, (1982) 2554-2558.

[95] Werbos, P.: "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," *PhD thesis, Dept. of Applied Mathematics, Harvard University, Cambridge, Mass.,* (1974).

[96] Rumelhart,D.E. and McClelland,J.L.:"Parallel Distributed processing: Exploration in the Microstructure of Cognition, " *MIT Press, Cambridge, Mass.,* (1986).

[97] Anderson ,J.A. and . Rosenfeld, E.: "Neuro computing: Foundations of Research," *MIT Press, Cambridge, Mass.,* (1988).

[98] Rojas,R.: " Neural Networks:A Systematic Introduction. " *Springer,* (2005).

[99] Vaga ,T.: " Proting from chaos: using chaos theory for market timing, stock selection, and option valuation, " *McGraw-Hill*, (1994).

[100] Wilamowski, B.M.: "Neural networks and fuzzy systems," *Chapters 124.1–124.8. The Electronic Handbook. CRC Press, Boca Raton, FL,* (1996). 1893–1914.

[101] Eeldman,. J., Fanty,M.A. and Goddard, N.H.: "Computing with Structured Neural Networks," *Computer*, Vol. 21, No. 3, Mar. (1988), 91-103.

[102] Polikar , R. , Udpa ,L., Udpa ,S. and Honavar ,V.: "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, Vol. 31, no. 4, (2001) 497 -508

[103] Rumelhart, D., Hinton, G., and Williams, R.: "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* Vol. 1. (1986).

[104] Fu, L.: "Neural Networks in Computer Intelligence, " *McGraw Hill,* (1994), 169-185.

[105] Barea, R. , Boquete,L. , Mazo,M. ,and López M.: "EOG guidance of a wheelchair using neural networks", *International Conference on Pattern Recognition,* (2009), 668–671.

[106] S. Haykin.: Communication Systems, 2nd version. New York: Wiley, 1994.

[107] Giles, C,Kuhn, G. and WilHams, R.: " Dynamic Recurrent Neural Networks: Theory and Applications, " *IEEE Transactions on Neural Networks,* Vol. 5, No. 2, 153-155.

[108] Janacek, G., and Swift, L.: "Time Series Forecasting, Simulation,"*Applications, New York: E. Horwood*, (1993), P. 11.

[109] Williams ,R. J. and Zipser, D.: "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, " Neural network/ Computation, Vol. 1. (1989), 270 - 280.

[110] Mandic ,D. and . Chamers, J.: "Recurrent Neural Networks for Prediction," *Chichester, New York, John Wiley,* (2001).

[111] Kolen,J. and, . Kremer, S.: "A Field Guide to Dynamical Recurrent Networks", *Institute of Electrical and Electronics Engineers, Inc., New York, USA.*

{END}