Regression Based Gaze Estimation with Natural Head Movement

Yang Fu

A Thesis

In the Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

May 2015

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Yang Fu

Entitled:“Regression Based Gaze Estimation with Natural Head Movement”

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science

Complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

____ Chair
Dr. H. Rivaz

____ Examiner, External
Dr. R. Ganesan (MIE) To the Program

____ Examiner
Dr. C. Wang

____ Supervisor
Dr. Dr. W-P Zhu

____ Supervisor
Dr. D. Massicotte (UQTR)

Approved by: ____
Dr. W. E. Lynch, Chair
Department of Electrical and Computer Engineering

____________20_____ __________________________________
Dr. Amir Asif, Dean
Faculty of Engineering and Computer Science

ABSTRACT

Regression based gaze estimation with natural head movement

Yang Fu

This thesis presents a non-contact, video-based gaze tracking system using novel eye detection and gaze estimation techniques. The objective of the work is to develop a real-time gaze tracking system that is capable of estimating the gaze accurately under natural head movement. The system contains both hardware and software components. The hardware of the system is responsible for illuminating the scene and capturing facial images for further computer analysis, while the software implements the core technique of gaze tracking which consists of two main modules, i.e., eye detection subsystem and gaze estimation subsystem.

The proposed gaze tracking technique uses image plane features, namely, the inter-pupil vector (IPV) and the image center-inter pupil center vector (IC-IPCV) to improve gaze estimation precision under natural head movement. A support vector regression (SVR) based estimation method using image plane features along with traditional pupil center-cornea reflection (PC-CR) vector is also proposed to estimate the gaze.

The designed gaze tracking system can work in real-time and achieve an overall estimation accuracy of 0.84º with still head and 2.26º under natural head movement. By using the SVR method for off-line processing, the estimation accuracy with head movement can be improved to 1.12º while providing a tolerance of 10cm×8cm×5cm head movement.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my supervisors, Dr. Wei-Ping Zhu in the department and Dr. Daniel Massicotte from University of Quebec at Trois-Rivieres, who directed me through my study at Concordia University and the writing of the thesis. The completion and the innovation of the developed system and the thesis presentation cannot be realized without their insightful advice. Their dedication and enthusiasm to work also remained as an example for me and motivated me through my M.A.Sc study here.

I would also like to thank Regroupement stratégique en microsystème du Québec (ReSMiQ) for its partial financial support for this study.

I am also very grateful that Dmitry Rozhdestvenskiy, Dave Chu and Jeffrey Landry offered kind help during the development of the hardware platform.

Contents

List of Figures

List of Tables

List of Abbreviations

BGF	Boosted Gaussian filter
FOV	Field of view
GRNN	Generalized regression neural network
IAIB	Image acquisition and illumination board
IC-IPCV	Image center-inter pupil center vector
IPV	Inter pupil vector
IR	Infrared light
KKT	Karush-Kuhn-Tucker
LoG	Ling of gaze
LoS	Line of sight
MPE	maximum permissible exposure
PC-CR	Pupil center-cornea reflection
PoR	Point of regard
RBF	Radial basis function
RMSE	Root mean square error
SE	System exposure
SSE	Sum square error
SVR	Support vector regression

List of Symbols

M	The size of Gaussian filter
d	The bias of boosted Gaussian filter
σ_x	The standard deviation of Gaussian filter on x-axis
σ_y	The standard deviation of Gaussian filter on y-axis
r_c	The radius of circle
C	Circular filter
G	Gaussian filter
G_B	Boosted Gaussian filter
$\mathbf{V}_{PC-CR}$	PC-CR vector
$\mathbf{V}_{IPV}$	Inter pupil vector
$\mathbf{V}_{IC-IPCV}$	Image center-inter pupil center vector
$\mathbf{P}_l$	The coordinate of left-eye pupil center in the 2D image plan
$\mathbf{P}_{\mathrm{r}}$	The coordinate of right-eye pupil center in the 2D image plan
$\mathbf{P}_l^{est}$	The estimated coordinate of left-eye pupil center in the 2D image plan
$\mathbf{P}_{\mathrm{r}}^{est}$	The estimated coordinate of right-eye pupil center in the 2D image plan
$\mathbf{G}$	The coordinate of the glint in the 2D image plan
$\mathbf{C}_I$	The coordinate of the center point in the image
Sx	Horizontal component of PoR
Sy	Vertical component of PoR

x	Horizontal component of PC-CR vector
y	Vertical component of PC-CR vector
ς	Estimation error of polynomial regression
a_i	Coefficients of regression polynomial on Sx
b_i	Coefficients of regression polynomial on Sy
$\mathbf{T}_{Sx}$, $\mathbf{T}_{Sy}$	Training set
$\mathbf{A}$, $\mathbf{B}$	The regression polynomial coefficient array
$\mathbf{R}$	The input array of the regression function
$\mathbf{u}$	The function input of SVR
v	The observation of PoR
$\varphi(\mathbf{u})$	The mapping function
$f(\mathbf{u})$	The approximation function
$\mathbf{w}$	Coefficients of the approximation function
$r(\mathbf{u}, v)$	The residual
$E(r)$	Error function
ε	Small positive value
ξ	Slack variables
C	Weigh of slack variables
$Q(\omega, b, \xi, \xi^*)$	The optimization problem with slack variables
$Q(\omega, b, \xi, \xi^*, \gamma, \gamma^*, \eta, \eta^*)$	The unconstrained optimization problem with Lagrange multipliers

γ, γ^*	Lagrange multipliers
η, η^*	Lagrange multipliers
b	The bias of the approximation function
$K(u, v)$	Kernel function
e	The estimation error of the location of pupil center
ME_{Sx}	Mean error of the estimated Sx
ME_{Sy}	Mean error of the estimated Sy
σ_{Sx}	Standard deviation of of the estimated Sx
σ_{Sy}	Standard deviation of the estimated Sy
θ_x	The horizontal angular error
θ_y	The vertical angular error
t	Time

1. Introduction

1.1. Background

Eye gaze represents a person’s focus of attention on the object in space that the viewer looks at. It is often referred to as point of regard (PoR) that corresponds to a point on the surface of the object in the scene.

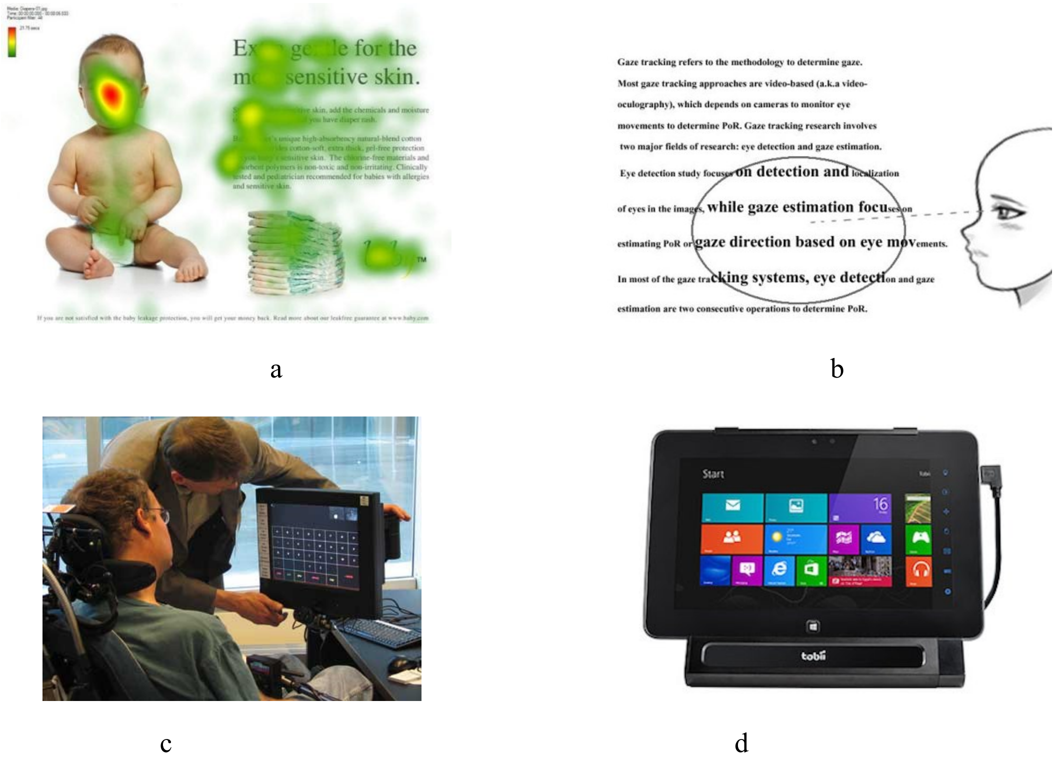
Gaze tracking refers to the process of determining the gaze direction or PoR with a computer software and/or hardware based device called gaze tracker. Usually, such device is an integration of one or several cameras and an illumination circuit. Although a gaze tracker has to work with a computer and proper supporting software to fulfill the tracking task, people are used to call the circuit with cameras as gaze tracker, since the software can be installed in any personal computer via an easy-to-use interface like USB. Most gaze tracking approaches are video-based, namely video-oculography, which rely on the camera(s) to monitor eye movements to determine the PoR. Gaze tracking research involves two major subjects: eye detection and gaze estimation. Eye detection focuses on the detection and localization of eyes in the images captured, while gaze estimation focuses on estimating the PoR or gaze direction based on the detected eye position and its movements. In most of the gaze tracking systems, eye detection and gaze estimation are two consecutive operations jointly conducted to determine the PoR.

Gaze tracking has found many applications in various fields. Most of these applications fall into two major categories: human behavior study and human-computer interaction. Human behavior study refers to researches on human behaviors based on gaze information. For example, Rayner made a study on reading as a specific example of cognitive processing, in which eye and gaze movement are used to find out how people use eyes to read [32]. Many applications along this study are based on human attention revealed by gaze [33,35,51,38], for example, gaze information could help make advertisements or web page design more effective by studying which part of them draws more attention from the viewers [51].

Heat map helps to reveal the viewer's degree of attention on different parts of an image. The heat map of an image is the integration of the viewer's focus over a period of time, and is represented by the color and intensity of marks on an extra layer of image over the original one. As seen from Figure 1.1(a), for example, extra marks were placed over the original image, in which higher intensity marks correspond to the parts that are more frequently viewed. Obviously, the viewers pay more attention to the baby's face than the text. As an useful reference, such technique could help publishers to study the interest of the viewers and improve the advertisements.

As another example of human behavior study, driving vigilance aims at determining a driver's level of vigilance based on the driver's face orientation, eyelid movement and gaze information [38]. Typically, a driving vigilance monitoring system alerts the driver once fatigue is detected.

Human-computer interaction applications take gaze as a controlling tool. For example, the gaze tracking system can be used to help disabled persons who for instance have language speaking and hand typing difficulties. In particular, the viewer's gaze could be displayed on a computer screen to replace mouse or used to type on a virtual screen keyboard [40]. Being aware of the viewer's gaze, the computer could react or adjust its displayed contents to serve the viewer. Such technique is called gaze contingent display [37]. Further more, the computer could enlarge the text in the area around the viewer's PoR to help with the reading as shown in Figure 1.1(b).



Gaze tracking refers to the methodology to determine gaze. Most gaze tracking approaches are video-based (a.k.a video-oculography), which depends on cameras to monitor eye movements to determine PoR. Gaze tracking research involves two major fields of research: eye detection and gaze estimation. Eye detection study focuses on detection and localization of eyes in the images, while gaze estimation focuses on estimating PoR or gaze direction based on eye movements. In most of the gaze tracking systems, eye detection and gaze estimation are two consecutive operations to determine PoR.

Figure 1.1 Applications of gaze tracking. (a) Heat map of an advertisement used to study the interest of the consumers. (b) Virtual magnifier. (c) Physically disabled people assistance. (d) Gaze controlled tablet.

Gaze based operating computer involves the least physical movement, thus remains the best communication means for people with cerebral palsy, ALS and high level spinal injuries [38]. Figure 1.2(c) is a gaze controlled computer used to assist ALS patient. The patient could stare at the pre-defined phrases to express his ideas or even use eye gaze to type on a virtual keyboard. Moreover, the patients could operate Windows system to fulfill more complicated tasks than simply typing. From Figure 1.1(d) for example, a commercial gaze tracker is attached to the bottom of a tablet to enable gaze control of Windows system. Such technique serves normal people as well, especially when they can not liberate their hands.

1.2. A typical gaze tracking system

A typical eye typing system is introduced here to explain the mechanism of gaze tracking. As shown in Figure 1.2, a person can use his/her eye gaze to type on a virtual keyboard.

Similar to the regular mouse pointing and clicking the gaze tracking system also needs to "point" and "click" but on the virtual keyboard on the screen. Two common solutions for clicking are blinking and staring. The advantage of blinking is its short response time but the gaze tracker loses the track of eyes in a moment and has to relocate the eyes afterwards. For the latter solution, on the other hand, the user has to spend 0.2-1 second staring at a key on the screen. Although it slows down the typing speed, it appears to be more practical and reliable to users.

In the process of pointing and staring, the computer estimates the user's line of gaze (LoG) or PoR by analyzing the eye region image. First, the facial image of the user is captured by a camera, from which the eye region image is extracted. A gaze estimation algorithm analyzes the acquired images and estimates the PoR. With the precisely estimated PoR, the computer or the eye typing software knows which key has been typed.

Figure 1.3 briefly shows the process that a gaze tracking system makes an estimation of the PoR based on the captured image. The tracking algorithm first locates eyes in the captured facial image to extract the eye region image. Then, some key features such as pupil center-cornea reflection (PC-CR) vector are extracted from the eye region image. The PC-CR vector is the difference between the coordinate of the pupil center and the cornea reflection in the eye image. The cornea reflection is the reflection of a fixed light source placed near the computer screen and is assumed to be static on the cornea surface. Thus, the PC-CR vector changes as eye ball moves, forming a feature highly related to the viewer's PoR. The relationship between the PC-CR vector and the PoR can be characterized by regression polynomials, based on which the software tracking system makes an estimation of PoR.

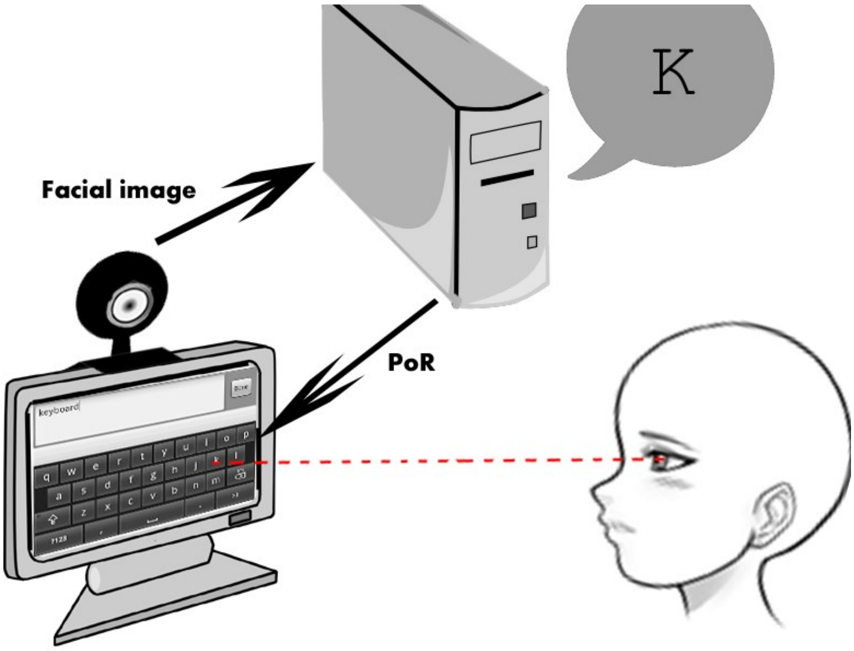


Figure 1.2 A typical eye typing system.

The major operations involved in the tracking system can be broadly classified into eye detection and gaze estimation. Eye detection, typically, includes eye localization in the facial image, eye tracking from frame to frame and feature extraction. Many works do not solely depend on eye features, but also use additional features like head pose, distance between eyes, facial landmarks, etc. The operation to acquire such features is called feature extraction, which is considered as a part of eye localization. Gaze estimation relies on eye features to estimate and track a person's gaze direction or point-of-regard.

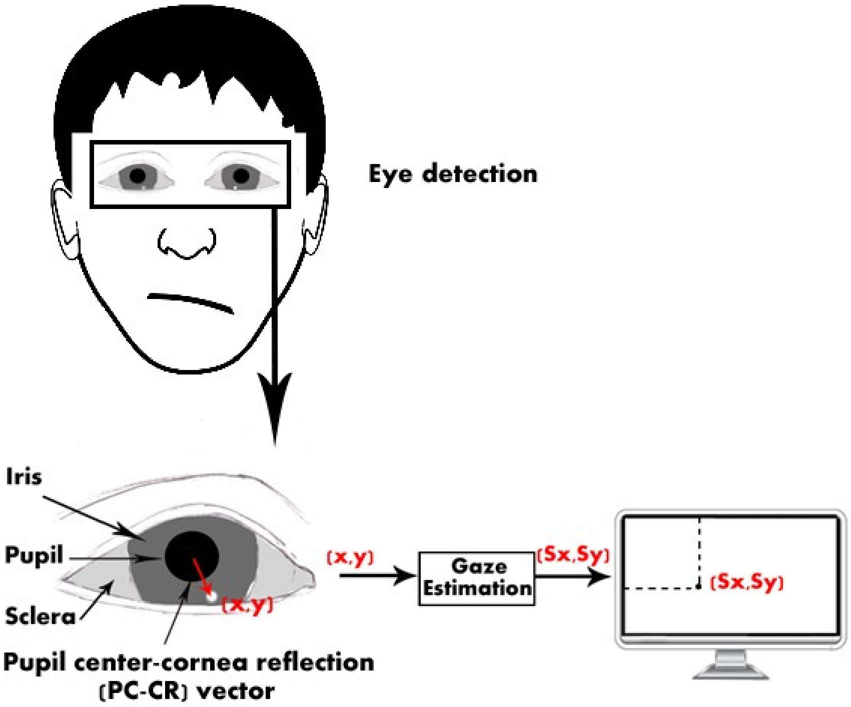


Figure 1.3 Eye detection and gaze estimation.

In the rest of the thesis, "eye detection" and "gaze estimation" which are the two key components of the gaze tracking system will be discussed in Chapter 2 and Chapter 3, respectively.

1.3. Literature review

Current gaze tracking approaches can be categorized into video-based methods (video-oculography) [1,2,4,5,6] and non-video-based methods [42-44] such as electro-oculography and photo-oculography. Video oculography dominates current gaze tracking research because it is more reliable, accurate and user-friendly. The following review only covers video-based gaze tracking which is also the focus of this thesis.

1.3.1. Eye detection

Eye detection methods can be categorized based on the eye models used or the lightning conditions in the setup [3]. According to eye models, eye detection techniques can be further divided into shape-based [9,20,46] and appearance-based methods [48,24].

Shape-based methods rely on eye or facial features to locate eyes. Pupil, for example, is an important feature due to its elliptical shape. Hough transform is an effective tool to detect circular objects like iris and pupil [11], however, such a method is too computation demanding to be applied in real time. The center of elliptical objects can also be located with voting schemes[9,20]. By drawing the isophotes on the boundary of pupil, iris and sclera and calculating the derivative of the isophotes, the center of pupil can be determined [9]. The authors of [9] also used the voting result to compute a convolution with a Gaussian kernel in order to make the result unique. On the other hand, the isophote is very sensitive to image intensity and thus vulnerable to the changes of lightning conditions. As such, some researchers made voting directly by computing two-dimensional (2D) convolution of the gradient of the image and a circular filter [20]. By setting the diameter of the filter slightly bigger than the diameter of the iris, the point with maximum filter response corresponds to the center of iris. This method is effective but very sensitive to the size of iris, thus incapable of detecting eyes at different distances.

Appearance-based methods are also known as image template or holistic methods. They detect eyes directly based on its color/intensity distribution or filter response. In [48] a wedge-shaped filter was used to detect eye corners, while Gaussian filters and Gabor wavelets were used to detect the eye shape. Image template and holistic method can also be found in many other object recognition/tracking applications[47] in addition to eye detection.

Based on lighting conditions, eye detection could be categorized into passive light [9,11,20,46,48] and active light methods [2,5,6,7,21,22]. Methods using images captured under visible light are called passive light methods, where those with images captured under infrared (IR) light are called active light methods. It is noted that general eye detection studies normally use passive light methods [9,11,20,46,48], while most gaze-tracking-oriented eye detection schemes rely on active light methods[2,6,7,8].

The use of IR light is common because it is invisible and helpful to both eye localization and gaze estimation. The wavelength of IR light is around 780-880nm, which is invisible to human eyes but easily visible to many CCD cameras. Accordingly, the cameras can capture the cornea reflection of the light source without causing interference on human eyes. This is very important in gaze estimation because the cornea reflection is a key feature to help determine the gaze. It is also very helpful to eye localization because the IR light reflects off the retina making the entire pupil bright and evident.

Eye detection with IR light takes advantage of the bright-pupil effect. When an IR light source is placed close to the optical axis of the camera, the pupil appears to be bright in the captured image since most of the IR light reflects back to the camera. Such light is called on-axis light. When the IR light source is placed away from the optical axis of the camera, the pupil appears to be dark. Accordingly, such light is called off-axis light [49]. This effect is best exploited by differencing techniques which can be dated back to late 1980s. Tomono et al [21] used three cameras (CCD1, CCD2, CCD3) and two infrared light sources with different wavelengths to locate pupils and glints. By using polarized IR light and filters, different cameras receive the light from desired light source simultaneously, where CCD1 only captures dark pupil, CCD2 captures dark pupil and glint, and CCD3 captures bright pupil. The pupil is extracted using the difference of the images captured by CCD2 and CCD3, and the glint is extracted using the difference of CCD1 and CCD2 images.

An easier approach was proposed in a recent work [5], where only 1 camera and 2 light sources (with the same wavelength) were used. The camera was surrounded by 2 sets of IR LEDs, forming an inner ring and outer ring, respectively. The two rings shine alternatively, and thus generate both on-axis light and off-axis light, thereby creating bright and dark pupil images. In this method, the camera was synchronized with the LEDs by setting the image acquisition rate equal to LEDs alternating frequency. Each pair of adjacent images captured from the synchronized camera contain a bright pupil image and a dark one. Then the pupil is easily extracted by differencing the two images because the non-pupil regions are assumed to be constant when the frame rate is fast enough.

1.3.2. Gaze estimation

There are two major approaches for gaze estimation: geometry based method [1,6,8,12,13,25] and interpolation based method [2,4,7,14,26], which are also referred to as 3D and 2D methods, respectively.

The geometry based method relies on the geometric relationship between the line of sight and the target plane. This is achieved by estimating the 3D gaze direction and intersecting it with the target plane [6,12]. As mentioned before, the visual axis represents the true direction of gaze, but it is not directly measurable. Rather, it is estimated based on the optical axis which can be derived by calculating the 3D location of cornea and pupil centers. Although there is a 4-5 degree offset between the optic axis and the visual axis depending on different individuals, the offset is constant for each person. With estimated optical axis, the difference between the two axises can be estimated with a calibration process in which the user is asked to look at pre-defined calibration points [1].

The interpolation method uses the image captured by an infrared camera. In this method IR LEDs are often used for illumination and creating cornea reflections. The difference between pupil center and cornea reflection is called pupil center-cornea reflection (PC-CR) vector. The interpolation based methods seek the relationship between the point of regard (PoR) and the PC-CR vector with so-called regression polynomials.

Early research on regression based method dates back to 1974. For example, Merchant et al. [28] used a single light system under IR illumination to estimate the PoR. They assumed the relationship between PC-CR vector and PoR to be linear so that a simple linear regression can be used. However, this method is found to have evident approximation error when the PC-CR vector is large. A full second-order regression polynomial was proved to outperform the linear regression function [6] and, as a matter of fact, is adequate to describe the nature of eye movement. Although some other polynomials have been proposed [4], the full second order polynomial is still a popular and reliable regression function today. However, the accuracy of gaze estimation with PC-CR techniques decays as head moves away from the original position. Head movement on longitudinal direction alone could cause considerable estimation error. Sanchez et al. used an additional light source and inter-glint normalization to improve the estimation accuracy under longitudinal head movement [2]. But the solution to lateral and vertical head movement was not explained.

Aside from polynomial regression, the generalized regression neural network (GRNN) [8] and support vector regression (SVR) [7] are also applied to estimate the PoR. Zhu and Ji employed multiple features as the input of the GRNN, including the PC-CR vector, the ratio of the major and minor axes of the ellipse of the pupil, the ellipse orientation, and glint coordinate [8]. They have reported that considerable head movement was tolerated with this method. They have also allowed head movement with SVR by using the geometric information of the pupil center and the PC-CR vector as training input into SVR model [8]. With sufficient training samples, which are acquired by asking the user to look at different pre-defined screen coordinates from several head poses, this method could also tolerate moderate head movement. But both of the methods are computationally intensive, as the GRNN methods can achieve only a frame rate of up to 20 Hz, and the SVR method may not work in real time. Moreover, as both are essentially regression methods, they require geometric information of eyes, which adds up to the implementation complexity of the system.

1.4. Objective and contributions of the thesis

Although researchers have made great efforts to realize gaze tracking under free head movement scenario [1,2,7,8,10], the performance is still below expectation. This thesis focuses on gaze tracking with regression based method and tries to overcome the degradation caused by head movement. Different from previous studies [7,8], this work estimates the PoR without requiring geometric information of user's eyes and head while allowing free head movement. By using 2D image plane information instead of the geometric information, the computational complexity can be greatly reduced.

The objective of this research is to develop a real-time gaze tracking system backed up by duly designed hardware and software for applications in the free head movement scenario. First of all, a hardware platform consisting of a single light source with IR LEDs is designed. An illumination circuit is built to synchronize the camera and the IR LEDs. A novel IR LED control circuit based on computer sound card is proposed to enable computer control of the LEDs via software. In order to reduce the hardware cost, a commonly used simple commercial web-cam, rather than expensive industrial camera, is used to capture the images.

With regards to the software core of the tracking system, a new eye detection scheme and a gaze estimation algorithm are proposed. With polynomial regression, the system could run real time with a frame rate of 15 Hz, which corresponds to the maximum frame rate of our camera--30 Hz. The gaze estimation accuracy is further enhanced by using the proposed SVR method when the tracking system is applied for an off-line processing.

With the designed gaze tracking system, the user could move head naturally in front of the computer screen. Here, the term "natural head movement" refers to a moderate degree of head movement while keeping shoulder still. The system can achieve an angular accuracy of 0.8° with a tolerance of head movement volume up to 10cm×8cm×5cm (vertical×horizontal×longitudinal).

1.5. Organization of the thesis

The rest of the thesis is organized as follows:

Chapter 2: This chapter presents an overview of the proposed gaze tracking system that is composed of three main sub-systems, namely, image acquisition, eye detection and gaze estimation. The image acquisition module involves most of the hardware work while eye detection and gaze estimation implement gaze tracking algorithms on the software level. The design and hardware implementation of the image acquisition sub-system is discussed in detail in this chapter.

Chapter 3: This chapter deals with eye detection using the image differencing technique, in which a boosted Gaussian filter is proposed to estimate the pupil center and glint location precisely. Two new image features, i.e., the inter-pupil vector (IPV) and the image center-inter pupil center vector (IC-IPCV) are defined and used as additional information to combat head movement impact.

Chapter 4: This chapter presents a few regression based gaze estimation methods including the polynomial regression method and the newly proposed support vector regression (SVR) method. Binocular gaze estimation, as a method to enhance the estimation accuracy, is also discussed. The pros and cons of these methods are analyzed.

Chapter 5: The experimental results of the gaze tracking system are provided in this chapter. First, the accuracy of eye detection is measured under different circumstances. Then, the performance of gaze estimation is investigated under two scenarios: no head movement and natural head movement. It is shown that with sufficient training, the proposed method prevails the traditional one under natural head movement.

Chapter 6: This final chapter summarizes the research work of the thesis with some of the original contributions highlighted. The limitation of the present system and possible future work are also discussed.

2. System architecture

2.1. System overview

Figure 2.1 shows an overview of the gaze tracking system composed of three subsystems: image acquisition, eye detection and gaze estimation. Image acquisition subsystem is responsible for capturing user's facial image. The facial image is then passed on to the eye detection subsystem, in which both eyes will be located in the facial image. With the eye location information, eye region images are generated, based on which several features are extracted and passed on to the gaze estimation subsystem. Gaze estimation subsystem estimates the PoR with the regression model based on polynomial regression or SVR proposed in this work. Detailed description of the three subsystems are provided in the following sections.

2.2. Image acquisition

The image acquisition subsystem consists of an IR camera and an illumination circuit. They are integrated on one PCB which is called image acquisition and illumination board (IAIB), as demonstrated in Figure 2.2.

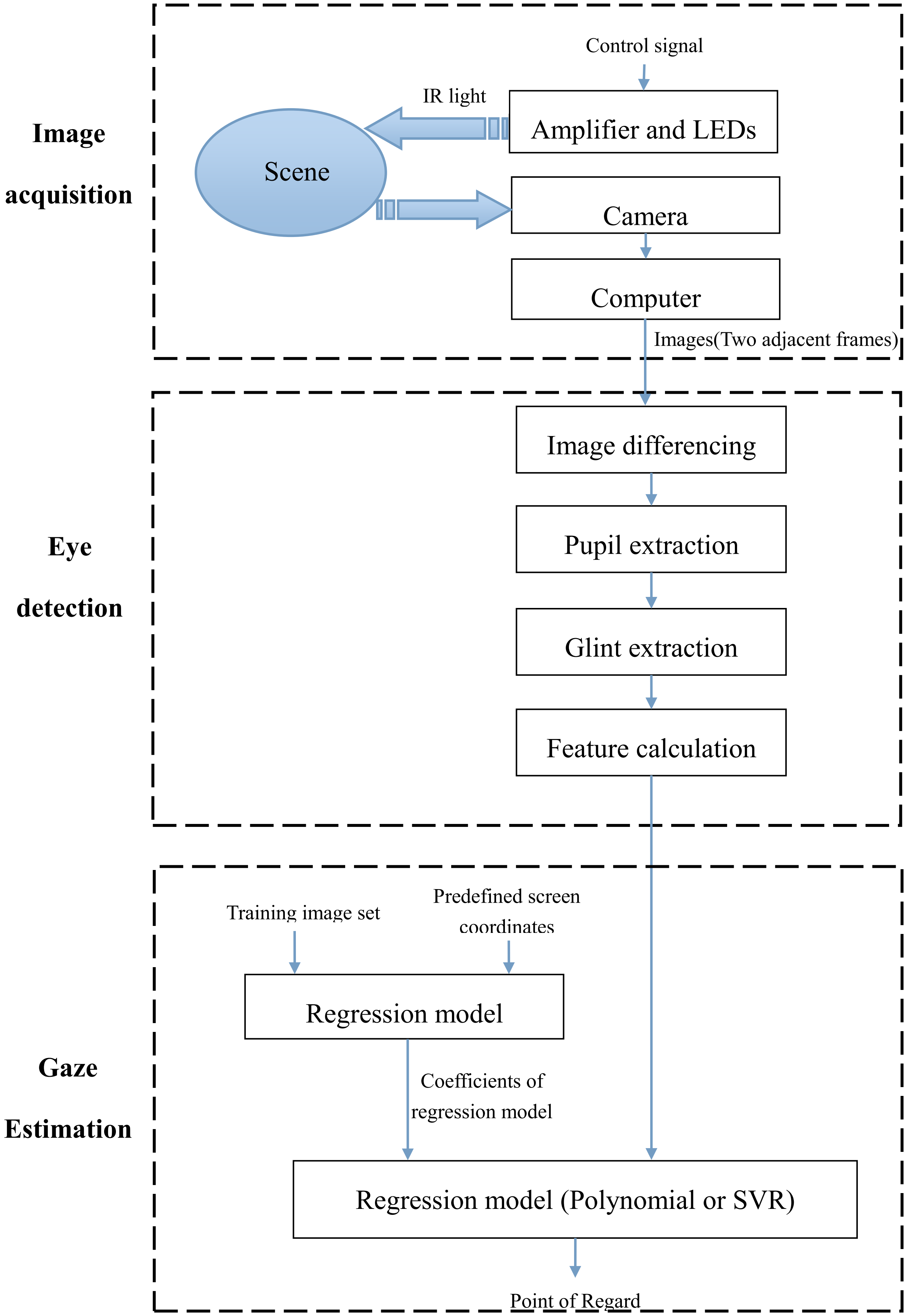


Figure 2.1 Gaze tracking system overview.

2.2.1. Image acquisition and illumination board

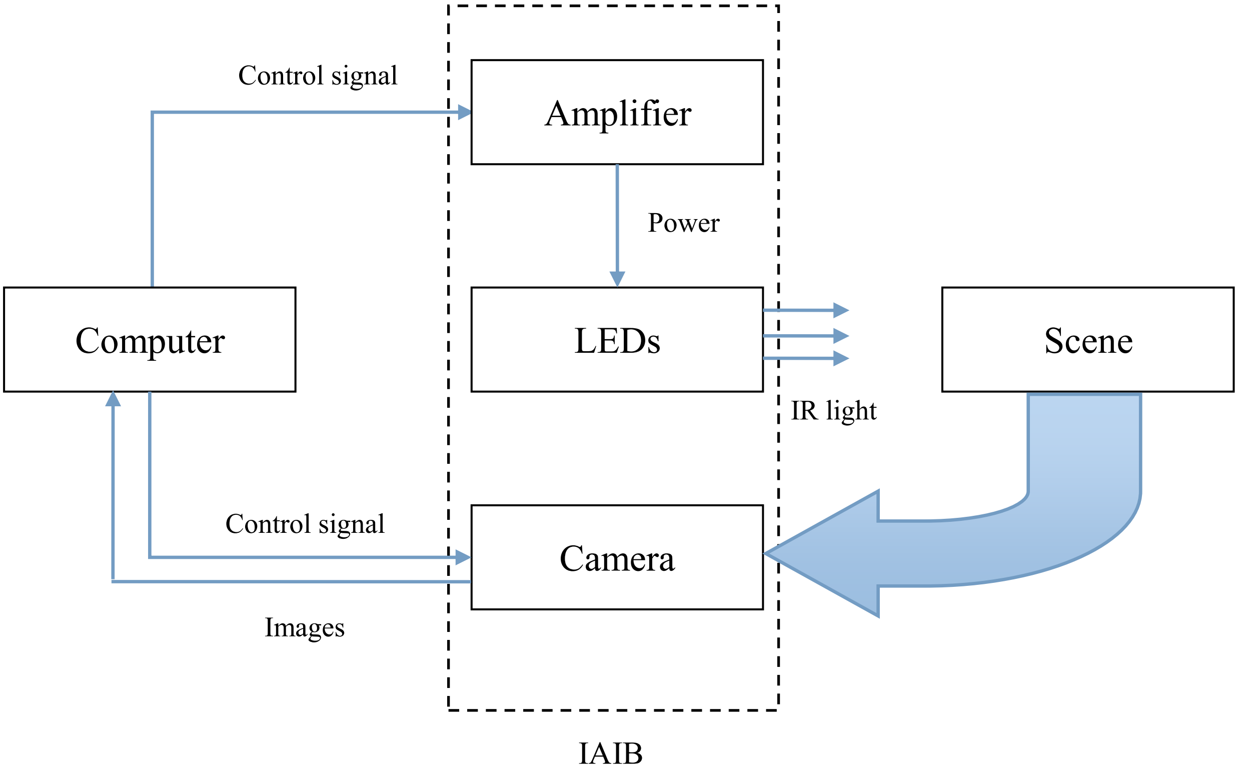


Figure 2.2 The architecture of the image acquisition and illumination board.

As illustrated in Figure 2.2, the IAIB consists of three major parts: amplifier, LEDs and camera. The amplifier and the LEDs are responsible for illuminating the scene. The amplifier or amplifier circuit drives two sets of LEDs according to the control signal generated by the computer. The signal to the amplifier controls the illumination pattern while the signal to camera controls the trigger signal which indicates the moment to capture image. Details of the illumination pattern will be described in chapter 3.

2.2.2. Amplifier and LEDs

There are several reasons that we use IR LEDs to illuminate the scene. First of all, IR light can be reflected by retina, which is an important feature we employ to locate the eyes. Secondly, IR light source is invisible to human eyes leaving no distraction to the user.

Moreover, by limiting the visible spectrum of the camera to the spectrum of IR light, the system is more robust against changes of environmental lighting.

However, IR LEDs are very power consuming and can not be driven by computer sound card directly. Therefore, an amplifier circuit as shown in Figure 2.3 is used to drive the LEDs. Two independent amplifiers are used since there are two sets of LEDs placed around the camera. Each consists of an voltage amplifier (uA741) and a transistor (2N2222). The LED we chose is EVERLIGHT HIR204C, which has a peak wavelength of 850nm. This diode works ideally with 1.45V voltage and 20mA current.

The safety of IR exposure is considered to ensure that the IR light does not hurt the retina of the viewer’s eyes. It is explained in Appendix B that our system is completely safe.

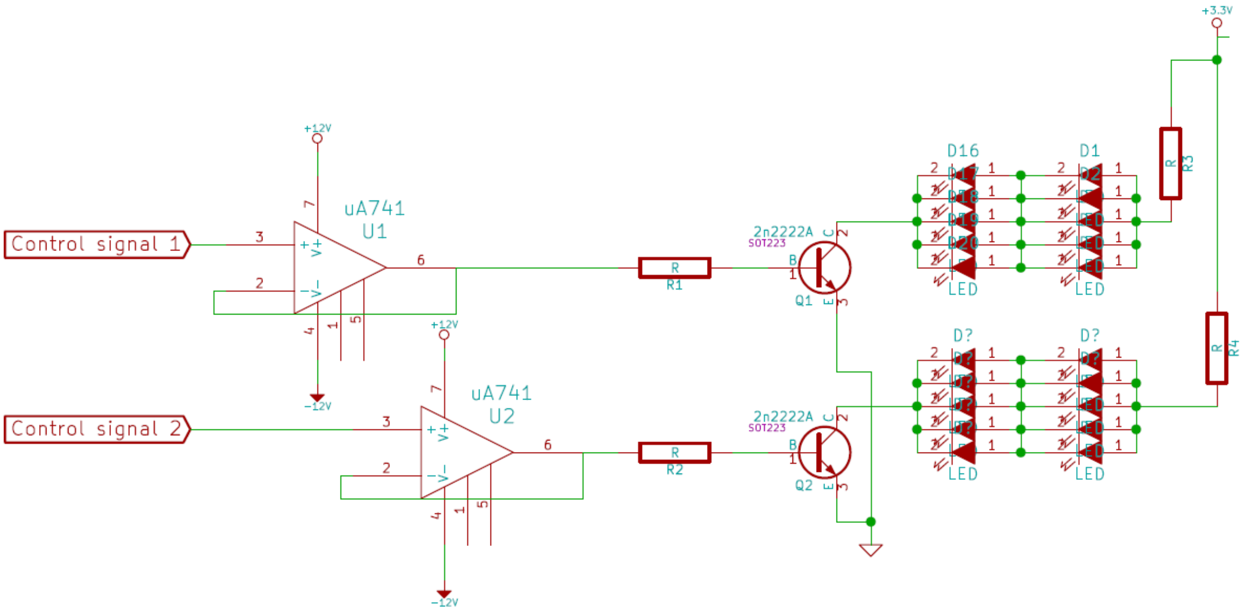


Figure 2.3 The schematic of the illumination circuit.

2.2.3. Camera

Industrial cameras are widely used for gaze tracking because they have better CCD sensors, higher image resolution and are more open to customization. However, industrial cameras are very expensive, usually over 500$. The camera used in this work is Microsoft Lifecam Studio, which is a normal webcam costing under 100$. It can work at a frame rate of up to 30Hz with a maximum resolution of 1280-by-720 pixels. Meanwhile, it has no IR light filter, thus can be used directly in this work.

Usually, lens with longer focal length is desired since long-length camera has a small field of view (FOV) and, consequently, better resolution of image details. For example, in the images captured by cameras with 6mm lens and 16mm lens, respectively, the corresponding eye region image could be 15-by-20 pixels or 60-by-80 pixels. Apparently, it is very difficult to analyze eye movement base on a poorly defined image with very few pixels. However, a larger focal length may not be a better solution. Since the FOV decrease as the focal length increases, a camera with small FOV loses track of eyes easily when the user moves away from the perfect position. Thus, a long focal length is favorable as long as the FOV is big enough to keep the viewer's face insight. The original lens of this camera is 6mm, which is too short to provide a good resolution. By trying several different lenses, it turned out that 16mm lens could offer a good balance between resolution and FOV.

2.3. Eye detection

The eye detection subsystem is responsible for localizing the eyes in the facial image and extracting features. There are four major steps in this subsystem, namely image differencing, pupil extraction, glint extraction and feature calculation.

Image differencing aims at segmenting the pupil in order to find the eyes. It is achieved by making subtraction with two adjacent frames in the captured images. Since the pupils appear to be dark in the odd frame but bright in the even frame, they appear to be the most accentuated part in the difference of the two frames. Ideally, the difference of the two frames contains two bright ellipses (segmented pupils) in a dark background. The pupil center is then located by finding the center of the ellipse. Using pupil center location, the eye region image could be extracted from the original image for further analysis.

The IR light reflects on cornea surface as well as retina. Since the cornea is like a spherical mirror, the reflection appears to be a small “dot”. But it is not ideally a one-pixel dot but a “bright circle” with a diameter of several pixels (depending on the camera). Therefore, it is more precise to describe its location using the center of the circle.

A similar approach as used to locate pupil center could be adopted to find glint center. With the location of pupil center and glint center, the pupil center-cornea reflection (PC-CR) vector can be calculated. In addition to the PC-CR vector used in traditional methods, the inter-pupil vector (IPV) and image center-inter pupil center vector (IC-IPCV) are also used in the proposed method as additional features.

2.4. Gaze estimation

Gaze is estimated using regression function, which takes previously extracted eye features as input and gives the estimated PoR as output.

Polynomial regression method is firstly introduced. The vertical and horizontal component of PoR are approximated independently by two second order polynomials. A training process is carried out in advance to determine the coefficients of the second order polynomial function. To do the training, the user is asked to look at several predefined screen coordinates. The images captured are then processed by the eye detection subsystem to extract the features, whose values are paired with the screen coordinates which are interpreted as the PoRs. Such pairs are regarded as the input-output relations of the regression function. With the entire training set, which is formed by hundreds of input-output pairs, the coefficients of the regression function can be determined through an optimization process.

The SVR based method relies on more features to allow head movement while giving a high tracking accuracy. The SVR method also needs a function training process to determine the function coefficients.

Details of polynomial method and SVR method shall be discussed in chapter 4.

3. Eye detection

The objective of the eye detection subsystem is to determine several key features that gaze estimation relies on. In order to detect these features, the pupil center and glint must be accurately located in the facial image. There are three major steps to locate pupil center and glint: image differencing, pupil extraction and glint extraction. Finally, three features, i.e., PC-CR vector, inter-pupil vector (IPV) and image center-inter pupil center vector (IC-IPCV) are calculated based on the locations of the pupil center and the glint.

3.1. Image differencing

The location of eyes is represented by the location of pupil centers. In order to find the pupil centers in the image, we try to segment the pupils in the entire image by employing bright-pupil effect.

3.1.1. Bright-pupil effect

IR light could reflect off the retina and if that reflection is captured by the camera, the pupil appears to be bright in the captured image, see Figure 3.1(a) for example. Such phenomenon is called bright-pupil effect. It appears when light source is placed close to the optical axis of the camera, since the light is coaxial with the optical path of the camera and most of the reflected light reaches the camera. On the contrary, when the light source is placed away from the camera, the reflected light could hardly reach the camera, causing a dark pupil in the captured image as shown in Figure 3.1(b). Accordingly, the two different light sources are called on-axis light and off-axis light.

The on-axis light and off-axis light are generated by putting two sets of LEDs around the camera. Figure 3.2(a) demonstrates the allocation of the LEDs around the camera on IAIB. The inner-ring LEDs and outer-ring LEDs are independent and work alternatively as shown in Figure 3.2(b) and Figure 3.2(c). Consequently, the dark pupil and bright pupil are witnessed in the situation of Figure 3.2(b) and Figure 3.2(c).

Figure 3.1 (a) Bright pupil. (b) Dark pupil.

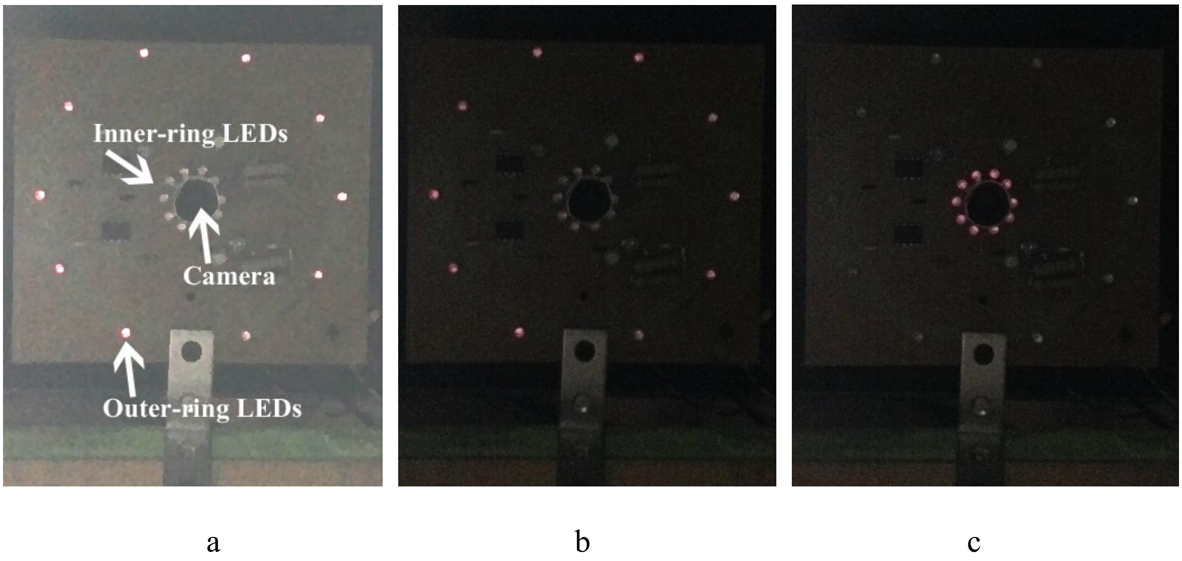


Figure 3.2 (a) Camera and illumination circuit. (b) The outer-ring LEDs. They appear to human eye to be dim red when they are on. (c) The inner-ring LEDs.

The peak wavelength of the LEDs is 850 nm, leaving most of its energy in the IR eye-insensitive spectrum and very little energy in the visible light spectrum. The "leaked" light makes these LEDs dim red to human eyes as shown in Figure 3.2(b),(c).

3.1.2. Pupil segmentation

The pupils are segmented by differencing the bright pupil and the dark pupil images. Figure 3.3 is the difference of Figure 3.2(b) and (c). It is obvious that the two pupils are the only evident areas regardless of the noise. Note that there is a bright region on the top-right corner of the image, which is the interference caused by the IR light leaked into the camera.

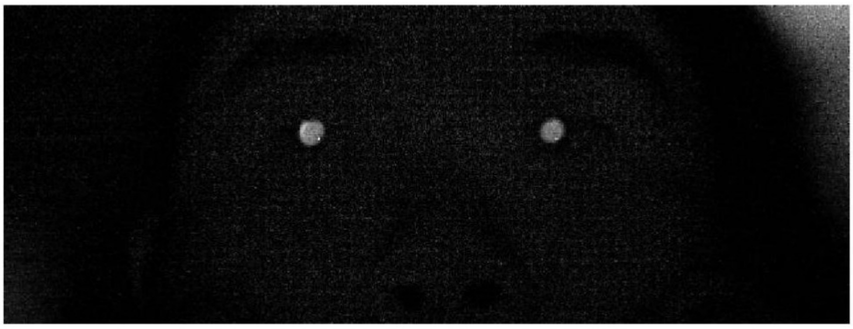


Figure 3.3 Segmented pupils: the difference of bright-pupil and dark-pupil images in Figure 3.2(b) and (c).

3.1.3. LEDs-camera synchronization

The differencing method relies on one assumption, that is, the time interval between the bright-pupil image and dark-pupil one is short enough to ensure that the change of the scene is negligible. In other words, the pupils in the two adjacent frames can not be perfectly segmented if they do not collide. This is the reason that we take the bright-pupil image and the dark-pupil one in adjacent frames. By setting the frame rate (image acquisition rate) at 30Hz, the time interval between two frames is 33.3ms, which is small enough to keep the scene almost invariant.

In order to guarantee the acquisition of bright and dark pupils in the adjacent frames, the LEDs and the cameras must be synchronized. Figure 3.4 shows the control signal from the computer to synchronize the LEDs and the camera. The control signal consists of two continuous signals to control the inner-ring LEDs and outer-ring LEDs, and one impulse signal to trigger the camera. When the LED-control signal is “1”, the LEDs are on, while the LEDs are off when it is “0”. According to Figure 3.4, the moment that the camera takes the first image corresponds to “outer-ring on”, therefore, the image captured is a dark-pupil image. Then the second image is taken when the pupil is bright. The system repeats this process to guarantee alternative occurrence of bright and dark pupils.

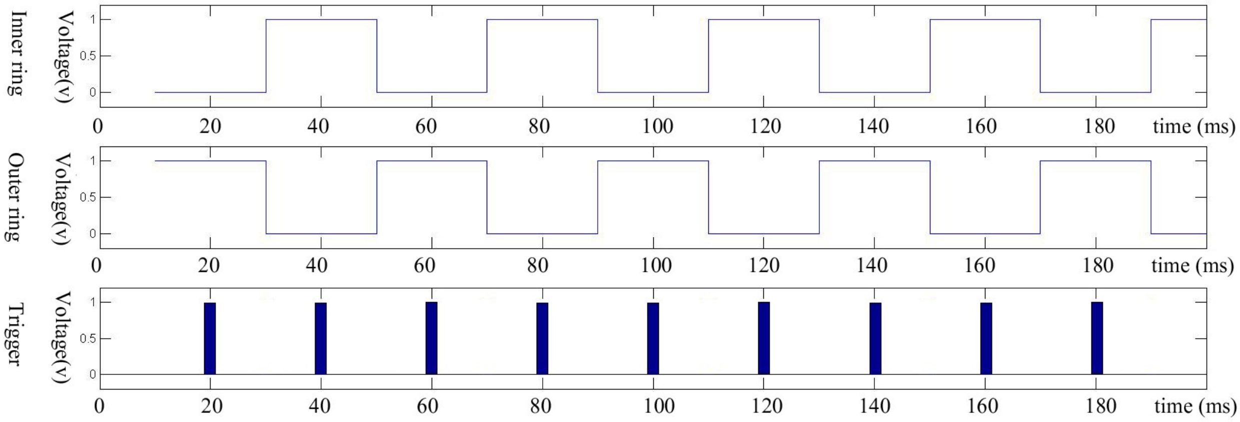


Figure 3.4 The control signal of the LEDs and the camera.

3.2. Pupil center localization

3.2.1. Pupil matching via boosted Gaussian filter

Template matching is a technique to detect a small part of an image which matches the template image or match filter [46]. By filtering the image using the template, a possible match can be obtained by finding a peak value in the filtered image. As an example, using Figure 3.3 as the original image and the left pupil as template image, the normalized filtered image is obtained as shown in Figure 3.5. Since the left pupil perfectly matches the template image, the peak value is found at the center of the left pupil. Although Figure 3.5 and Figure 3.3 appear to be similar, the intensity of the pixels around the pupils are quite different. In Figure 3.3, the intensity of the pupil-center is as high as its surroundings, while in Figure 3.5, the intensity of the pupil-center is the peak value in the local area.

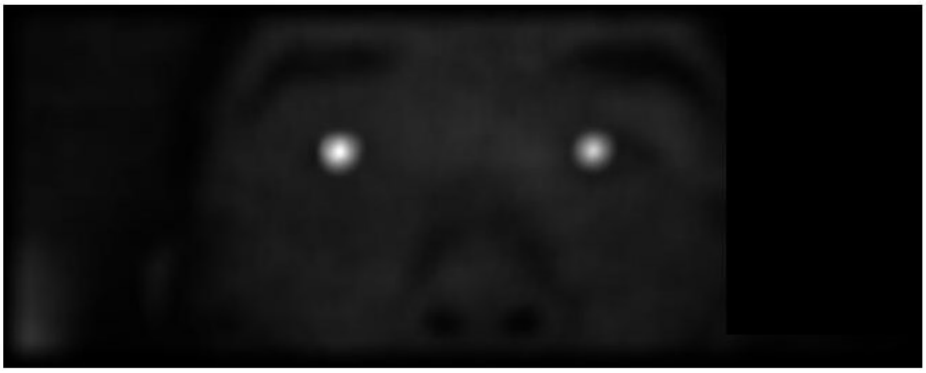


Figure 3.5 The normalized filtered image of Figure 3.3 with

left pupil image as template image.

However, the shape of the accentuated pupil changes every frame and the perfect template image is not predictable for the next frame. The solution is to use an universal template image or match filter. Existing filters include ideal circular filter [20] and Gaussian filter [48]. A ideal circular filter has a circular contour on the 2D spatial plane that is similar to the accentuated pupil in Figure 3.3, which is a $(2M+1)\times(2M+1)$ filter defined in (3.1), in which m and n are integers ranging from -M to M and r_c is the radius of the circle. A Gaussian filter is defined in (3.2) where σ_m and σ_n are the standard deviation used to control the width or the slope of the curve. In this work we use a new boosted Gaussian filter (BGF) defined in (3.3) where d is the bias which is a negative value.

$$C(m,n)=\begin{cases}1 & m^2+n^2\le r_c\\ 0 & m^2+n^2 \succ\end{cases} \tag{3.1}$$

$$G(m,n)=\exp\left(-\left(\frac{m^2}{2\sigma_m^2}+\frac{n^2}{2\sigma_n^2}\right)\right) \tag{3.2}$$

$$G_B(m,n)=\exp\left(-\left(\frac{m^2}{2\sigma_m^2}+\frac{n^2}{2\sigma_n^2}\right)\right)+d \tag{3.3}$$

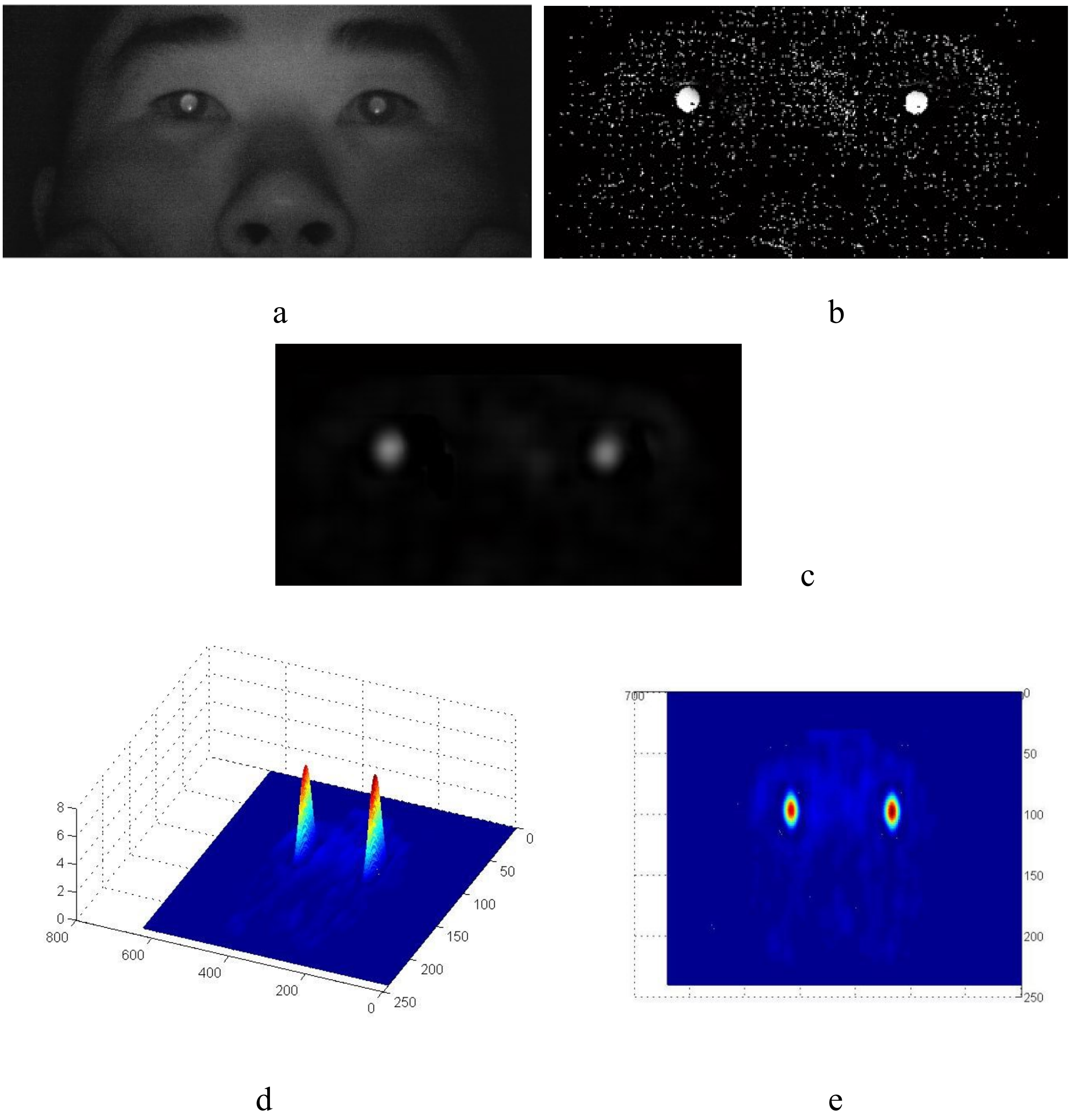


Figure 3.6 (a) Original image. (b) Bright pupil image. (c) 2D gray scale view of the filtered result. (d) 3D surface view of the filtered image. (e) 2D intensity view of the filtered result.

Noticeably, although these filters are essentially low pass filters, they are not used to smooth the image but used as matching filter to detect the pupil. An example using the proposed BGF to find the pupil center is demonstrated in Figure 3.6. The original bright pupil image and accentuated pupil image are presented in Figures 3.6(a) and (b), followed by the filtered image shown in the gray scale view, 3D surface view and 2D intensity view, respectively, in Figures 3.6 (c)-(e). It is obvious that in Figure 3.6(d) there are two peaks at two pupil centers, which means that the proposed BGF can clearly detect the pupil center.

3.2.2. The merit of the boosted Gaussian filter

Here we compare the three aforementioned filters fo the detection of pupil center. Although the ideal circular filter appears to be more similar to the pupil, it is very vulnerable to the change of the size of the pupil. The peak value is not unique if the size of the filter is not close to the size of the pupil. As a matter of fact, the size of pupil varies with individuals and even for the same individual it changes as the user moves close/away to the computer screen. For example, by using a pupil image which is 20% bigger than the normal one, the three filters perform differently. The filtered images are demonstrated in Figure 3.7, in which the peak is unique when using BGF and Gaussian filter while it is not unique when the ideal circular filter is used.

Additionally, BGF is robuster than the Gaussian filter and the ideal circular filter against interference, especially the glint which is usually much brighter than the pupil. The glint affects the filtering result as the estimated pupil center is shifted towards the glint. Such effect is obvious when using an ideal circular filter. The negative bias in BGF can combat the impact of glint, making BGF more reliable. Since the true pupil center can hardly be determined, the comparison of pupil center localization accuracy can be done by comparing the estimated PoR. Our experiment shows that the estimated PoR based on the features detected by BGF method prevails that with the Gaussian filter by 0.13^0 and that with the ideal circular filter by 0.36^0.

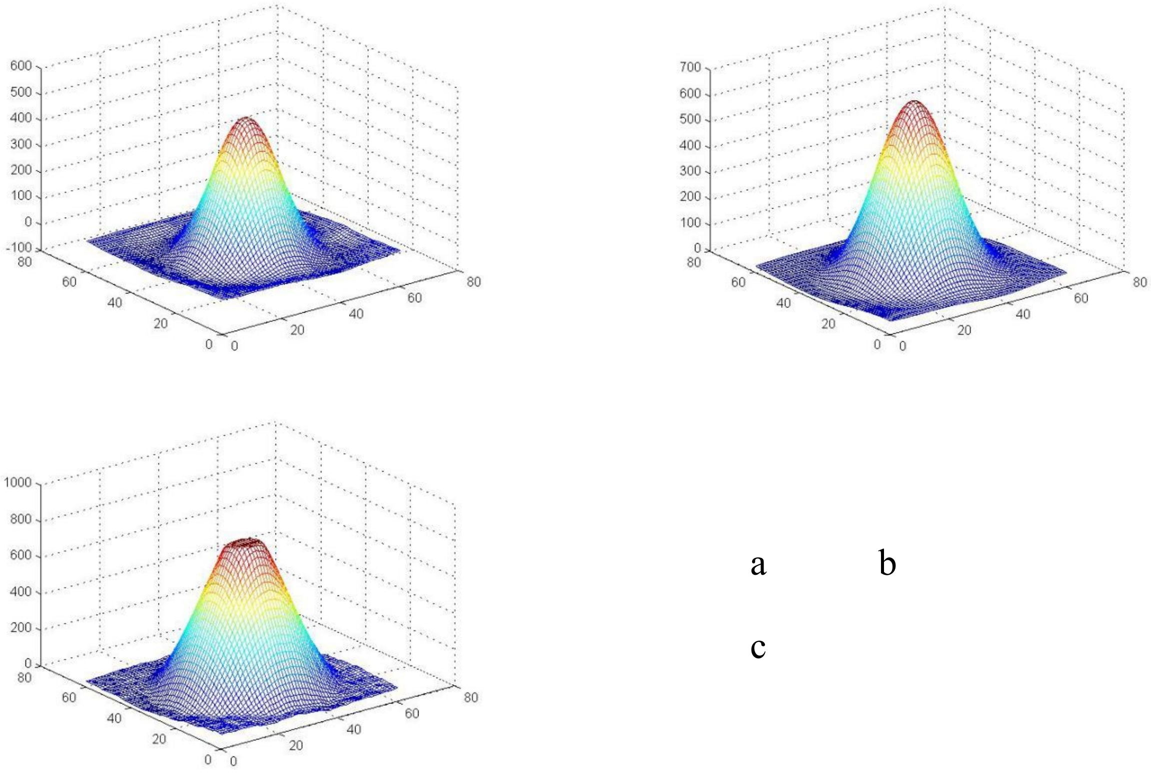


Figure 3.7 (a) 3D mesh view of the peak generated with BGF.

(b) 3D mesh view of the peak generated with Gaussian filter.

(c) 3D mesh view of the peak generated with ideal circular filter.

3.3. Glint localization

In order to acquire the PC-CR vector, the location of the cornea reflection or the glint center must be determined. It is noticed that the glint is not ideally a “dot” but a “circle”, therefore, the center of the circle is assumed to be the location of the glint.

The glint can be searched around the pupil instead of over the entire image to reduce the computational cost. However, the definition of the extracted image around the pupil is very low. In order to express the PC-CR vector accurately, the pupil image is first interpolated. Then the glint is detected based on the interpolated image.

3.3.1. Image interpolation

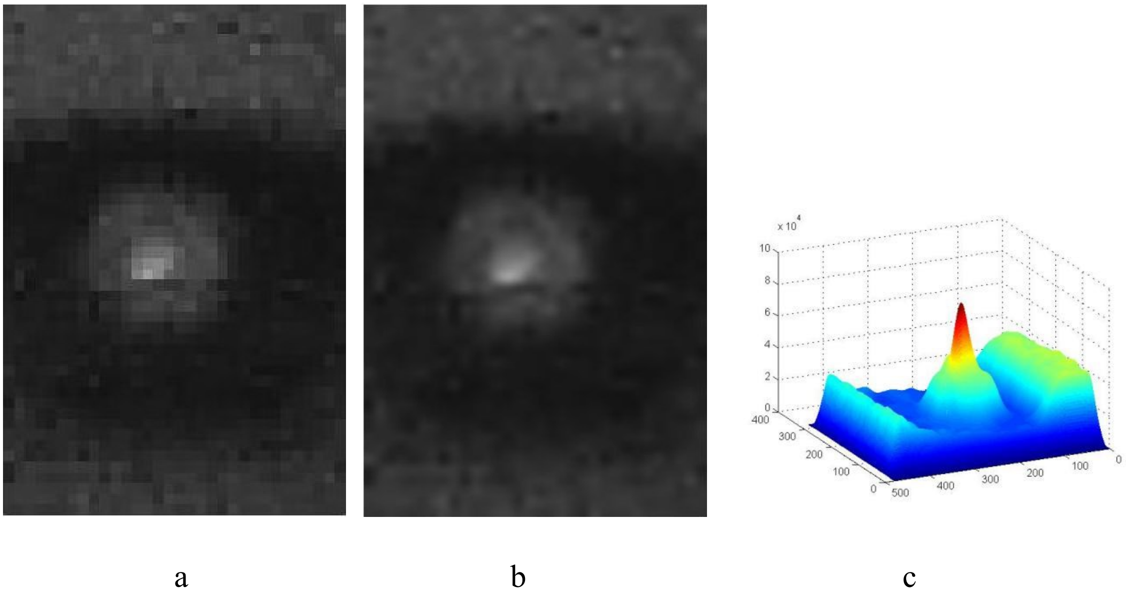


Figure 3.8 (a) Eye region image with resolution of 48-by-32 pixels. (b) Interpolated eye region image with resolution of 480-by-320 pixels. Note that both of the images are zoomed here. (c) The mesh view of the filtered result, where the pixel with maximum intensity corresponds to the glint center.

The extracted eye region image usually has very limited pixels. This is obvious in Figure 3.8(a) where an eye region image extracted from the facial image has a resolution of only 48-by-32 pixels. Under such a resolution, the location of the glint center and PC-CR vector can hardly be accurately determined since there are too few quantization levels. Our solution is re-sizing the image using bi-linear interpolation.

For example, in the original eye region image shown in Figure 3.8 (a), the PC-CR vector is (-1,1) (in pixels). The interpolated image in Figure 3.8(b) has 480-by-320 pixels instead of the original 48-by-32 pixels, the PC-CR vector becomes (-7,13). In the scale of the original image, such PC-CR vector equals (-0.7,1.3). Obviously, (-0.7,1.3) is closer to the true value than (-1,1).

3.3.2. Glint center detection

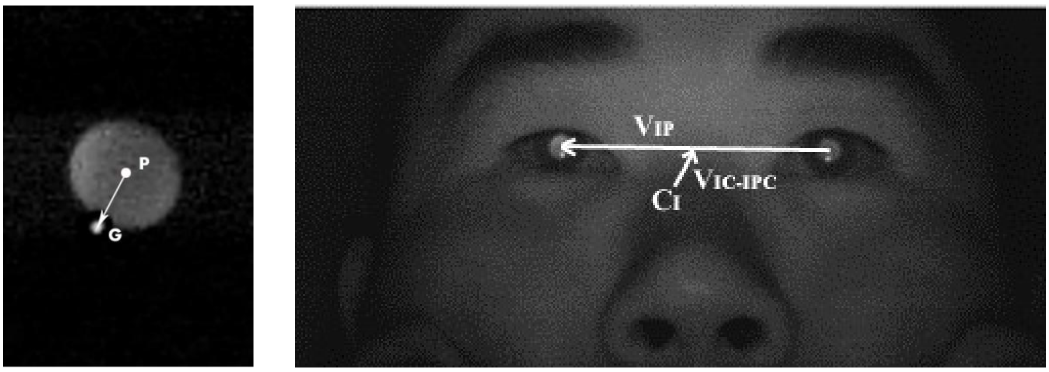
Even though the glint is brighter than the pupil and very likely it happens to be the brightest point in the image, we can not determine the glint center by simply searching for the maximum intensity value. Sometime, although very rare, there will be impulse noise with high intensity which could be mistaken as the glint. However, with BGF, the selection of glint center does not solely depend on the intensity of a single pixel but the intensity of its surroundings as well. For example, Figure 3.8(c) shows the mesh view of the filtered image of Figure 3.8(b), where the pixel with the maximum intensity corresponds to the glint center. We have found this selection very robust.

3.4. Features extraction

The most important feature is $\mathbf{V}_{\text{PC-CR}}$, the PC-CR vector which is the difference between the pupil center and cornea reflection (glint). Figure 3.9(a) shows the PC-CR vector in an eye region image, in which $\mathbf{P}$ stands for the pupil center and $\mathbf{G}$ stands for the glint center. The definition of $\mathbf{V}_{\text{PC-CR}}$ is given by

$$\mathbf{V}_{\text{PC-CR}} \triangleq \mathbf{G} - \mathbf{P} \tag{3.4}$$

It is well known that when people look at different things, both head movement and eye movement are more likely to happen. These movements involve rotation and translation in general. In a head-restrained situation, the PoR is only related to the rotation of eyeball. Thus, the PC-CR vector revealing the rotation of eyeball can effectively help to estimate the PoR. But in a free head movement situation, the rotation and translation of eyes and head must be considered. Therefore, using the PC-CR vector alone is not sufficient and additional information on the head movement would be needed.



a b

Figure 3.9 (a) $\mathbf{V}_{\text{PC-CR}}$:the PC-CR vector. (b) $\mathbf{V}_{\text{IP}}$: inter-pupil vector (IPV). $\mathbf{V}_{\text{IC-IPC}}$: image center-inter pupil center vector (IC-IPCV). $\mathbf{C}_{\text{I}}$: image center point.

Two additional head movement related features are discovered in the image plane, one is the inter pupil vector (IPV) $\mathbf{V}_{\text{IP}}$, which is the difference between two pupil center points, as seen in Figure 3.9(b). Here $\mathbf{V}_{\text{IP}}$ is defined in (3.5), where $\mathbf{P}_{\text{l}}$ and $\mathbf{P}_{\text{r}}$ stand for the coordinate vectors of the left and right pupil centers, respectively. The L1 norm of $\mathbf{V}_{\text{IP}}$ is also known as inter-pupil distance, which is used to normalize the PC-CR vector. Another feature is the image center-inter pupil center vector (IC-IPCV) $\mathbf{V}_{\text{IC-IPC}}$, which is the vector from the image center to the center of two pupils, as seen in Figure 3.9(b). The $\mathbf{V}_{\text{IC-IPC}}$ is defined in (3.6) where $\mathbf{C}_{\text{I}}$ stands for the image center point, which is (120,320) in this work since the size of the facial image is 240-by-640. The $\mathbf{V}_{\mathrm{IP}}$ is more relevant to head rotation while $\mathbf{V}_{\mathrm{IC\text{-}IPC}}$ is more relevant to head translation. Details regarding these two features will be discussed in chapter 4.

$$\mathbf{V}_{\mathrm{IP}} \triangleq \mathbf{P}_{\mathrm{l}} - \mathbf{P}_{\mathrm{r}} \tag{3.5}$$

$$\mathbf{V}_{\mathrm{IC\text{-}IPC}} \triangleq \frac{1}{2}(\mathbf{P}_{\mathrm{l}} - \mathbf{P}_{\mathrm{r}}) - \mathbf{C}_{\mathrm{I}} \tag{3.6}$$

It is shown in [50] that a gaze estimation system becomes robuster with the normalized PC-CR vector which is obtained by dividing the PC-CR vector by a normalization factor such as glint distance, pupil distance and inter-glint distance. Glint distance is the euclidean distance between two glints in different eyes. Pupil distance is the Euclidean distance between two pupil centers from both eyes. Some systems employ multiple light sources to create several glints on the cornea surface. Then the normalization factor, namely inter-glint distance, in such systems can be the distance between the two glints in one eye. Using normalization could help improve the accuracy of the estimation and reduce the quantization problem introduced in section 3.3.1.

4. Regression based gaze tracking

Two regression methods are introduced in this chapter, the polynomial regression and the support vector regression. Before using them to estimate the PoR, the coefficients of the regression models must be determined via a training process based on the known eye features and PoR.

4.1. Polynomial regression based gaze tracking

The PC-CR based polynomial regression method assumes that the cornea surface is a perfect mirror and the cornea reflection or glint is stationary. Therefore, the PC-CR vector, which is the difference between the cornea reflection and pupil center, reveals the eye movement. For example in Figure 4.1, the PC-CR vector, which is represented by an white arrow, is acquired when the user looks at the upper-left and upper-right corners of the screen. It is obvious that the vertical component of the PC-CR vector is a strong indicator of the vertical movement of PoR.

The correspondence between the PC-CR vector and the position of PoR was firstly assumed to be linear in very early works [28]. By using the vertical and horizontal components of the PC-CR vector to linearly approximate the vertical and horizontal components of the PoR separately, the estimation is only accurate around the center of the screen but has inevitable error on the rest of the screen.

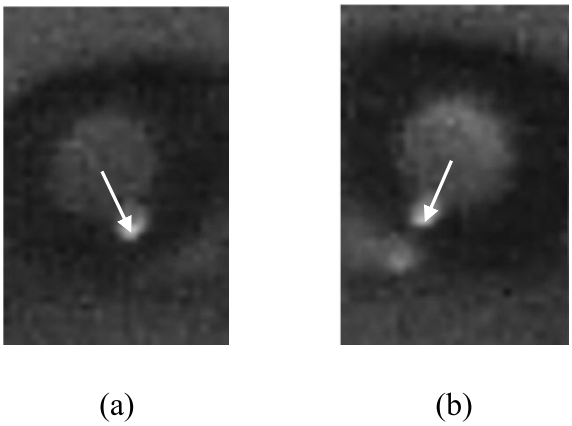


Figure 4.1 (a) The pupil image when user looks at the upper-left corner of the screen.

(b) The pupil image when user looks at the upper-right corner of the screen.

Latter works found that the second order regression function could approximate such correspondence more accurately [6]. A full second order polynomial as given in (4.1) is the most commonly used regression polynomial in the state-of-the-art works [2,6,7,26].

$$\begin{aligned} s_x &= a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \\ s_y &= b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2 \end{aligned} \tag{4.1}$$

In the above equation, (x, y) represents the PC-CR vector and (S_x, S_y) represents the estimated PoR. $(S_x, S_y) = (0,0)$ corresponds to coordinate of the pixel on the upper-left corner of the screen, $(S_x, S_y) = (1280,1024)$ corresponds to that on the lower-right corner of the screen.

Another polynomial (4.2) is used in a recent work [3] and in industries such as LC technology. It is similar to (4.1) but discards several entries that have minor influence on the regression result.

$$\begin{aligned} s_x &= a_0 + a_1 x + a_2 y + a_3 xy \\ s_y &= b_0 + b_1 x + b_2 y + b_3 y^2 \end{aligned} \tag{4.2}$$

A study on regression polynomials [4] claims (4.3) to be the optimal regression polynomial after comparing 400,000 models which differ in the number of terms and polynomial orders.

$$
\begin{aligned}
s_x &= a_0 + a_1 x + a_2 x^3 + a_3 y^2 \\
s_y &= b_0 + b_1 x + b_2 y + b_3 x^2 y
\end{aligned}
\tag{4.3}
$$

According to our study, however, it can not be concluded which polynomial is the universal optimal regression function for gaze estimation since the system setup could differ in different scenarios.

4.1.1. Regression function training

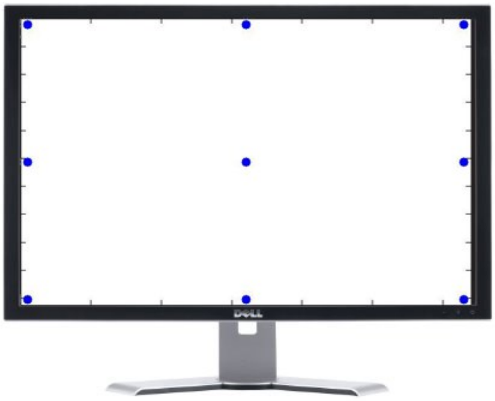
The training of a regression function aims at determining its coefficients.To this end, a training set is needed which is composed of a large number of pre-known input-output pairs of the training function.

The training process in this thesis follows two commonly used steps. First, in order to acquire the training set, the user is asked to look at several predefined dots or calibration points on the screen. The coordinate of these dots is assumed to be the true PoR, which is used as the output of the training function. And the PC-CR vector is extracted as the input of the training function. The most common allocation of these dots is the 3-by-3 grid shown in Figure 4.2, which has been adopted by many works[10,14,17,25] and commercial products such as Tobii Eyex. There are also other test grids such as 3-by-4 [1], 4-by-4 [2,4,11] and 4+4 [2,4]. But what they have in common is that some dots are placed close to the edges and corners of the screen and the rest are distributed evenly on the screen.

In this thesis work the user is asked to look at 9 dots on the screen. The dots are displayed one by one, each lasts for 3 seconds. Since the the image acquisition rate is 30Hz, we have 15 extracted PC-CR vectors per second (each PC-CR vector is extracted every 2 frames) and totally 45 PC-CR vectors from each dot in the 3-second period. Each PC-CR vector is paired with the corresponding screen coordinate forming one entry of the entire training set. All the entries derived in the whole training process forms the entire training set for Sx and Sy as defined in (4.4) and (4.5), in which M is the size of the training set.

$$\mathbf{T}_{Sx} = \left\{(x_i, y_i, Sx_i) \middle| i = 1,2,3,...M\right\} \tag{4.4}$$

$$\mathbf{T}_{Sy} = \left\{(x_i, y_i, Sy_i) \middle| i = 1,2,3,...M\right\} \tag{4.5}$$



c

Figure 4.2 The 3-by-3 calibration grid on the screen.

By using the screen coordinates of the dots as observed response and PC-CR vectors as inputs, the coefficients of the regression function are derived through multiple linear regression. According to the multiple linear regression theory [26], the coefficient vectors $\mathbf{A} = \left\{a_i \middle| i = 1,2,...,5\right\}$ and $\mathbf{B} = \left\{b_i \middle| i = 1,2,...,5\right\}$ in (4.1) are determined by minimizing the sum of squared errors (SSE) defined by (4.6) and (4.7), in which ζ is the estimation error and $\hat{S}x$ and $\hat{S}y$ is the estimated horizontal and vertical component of PoR.

$$SSE_{Sx} = \sum_i \hat{\zeta}^2_{Sx_i} = \sum_i (Sx_i - \hat{S}x_i)^2 \tag{4.6}$$

$$SSE_{Sy} = \sum_i \hat{\zeta}^2_{Sy_i} = \sum_i (Sy_i - \hat{S}y_i)^2 \tag{4.7}$$

It has been shown in [26] that the optimal coefficients $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are given by (4.8) and (4.9), respectively, in which $\mathbf{R} = \left\{1, x, y, xy, x^2, y^2\right\}$, $\mathbf{S}_x = \left\{Sx_i \middle| i = 1,2,3,...,M\right\}$ and

$\mathbf{S}_y = \{ Sy_i | i = 1,2,3,...,M \}$.

$$\hat{\mathbf{A}} = \left(\mathbf{R}^T\mathbf{R}\right)^{\text{-1}} \mathbf{R}^T\mathbf{S}_{\text{x}} \tag{4.8}$$

$$\hat{\mathbf{B}} = \left(\mathbf{R}^T\mathbf{R}\right)^{\text{-1}} \mathbf{R}^T\mathbf{S}_{\text{y}} \tag{4.9}$$

Figure 4.3 visualizes an estimated full second order regression function, in which the x and y axes are the vertical and horizontal component of PC-CR vector and the z axis is the Sx and Sy. From the figure we can find out that the plane is very similar to a first order plane, while small nonlinearity is noticeable on the edges of the plane. This explains that linear function could offer a rough estimation while second order function are more accurate.

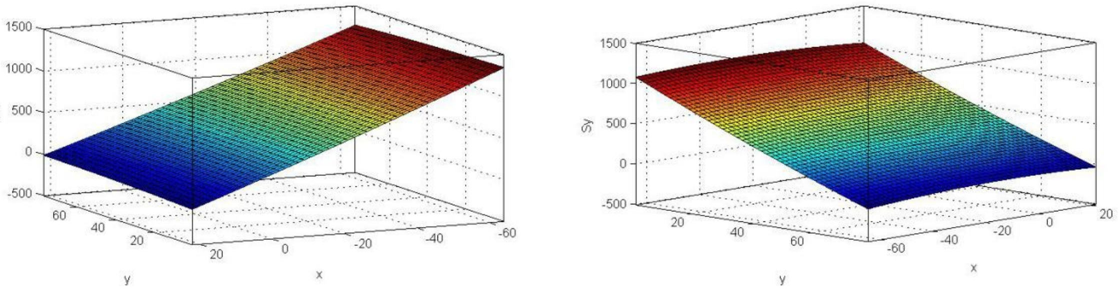


Figure 4.3 The surface of the second order regression function.

4.2. Support vector regression based gaze tracking

Support vector regression (SVR) can be used to approximate highly non-linear functions accurately by using kernel functions and thus considered as an efficient tool for function approximation [27].

4.2.1. General SVR theory

First of all, we use $(\mathbf{u}_i, v_i)$ $(i = 1,...,M)$ as input-output pairs, where $\mathbf{u}_i$ is the ith

M-dimensional input vector, v_i is the ith scalar output and M is the number of training data. The approximation function $f(\mathbf{u})$ is used to approximate v and is written in the linear form as given in (4.10), where ϕ is the mapping function used to map the original M-dimensional input vector $\mathbf{u}_i$ to a l-dimensional feature space vector $\phi(\mathbf{u}_i)$, $\mathbf{w}$ is the weight vector and b is the bias term. Noticeably, $\phi(\mathbf{u})$ and is an abstract term with no practical sense and l does not need to be specified, while $\phi^T(\mathbf{u})\phi(\mathbf{u}_i)$ can be specifically expressed and is referred to as kernel function, which will be quite obvious by the end of this section.

$$f(\mathbf{u}) = \mathbf{w}^T\phi(\mathbf{u}) + b \tag{4.10}$$

Similar to linear regression in which the square error function is used to measure the error, SVR relies on a piecewise linear function shown in Figure 4.4 to measure the approximation error. The error function $E(r)$ is given in (4.11), in which ε is a small positive value and r, given in (4.12) is the residual defined by the difference between the true value of v and the its estimate. From (4.11) and Figure 4.4 we can find that the error is not equal to the residual, and the error is 0 when the residual is small enough.

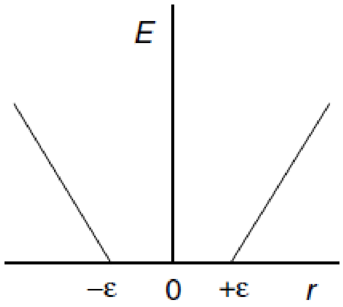
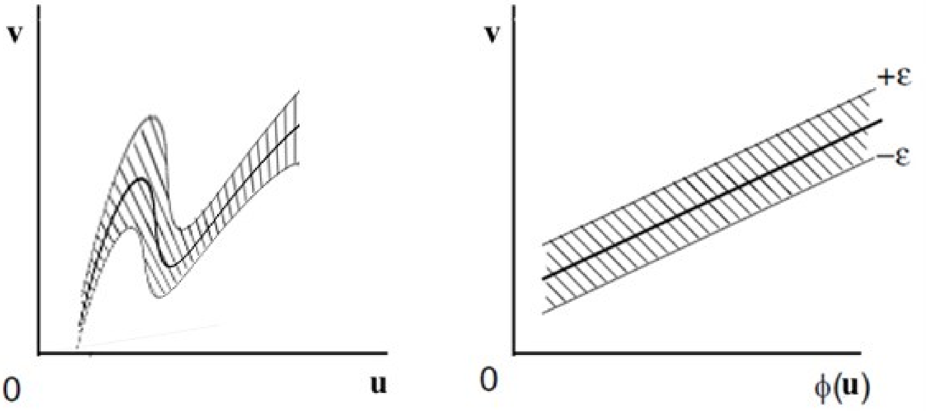


Figure 4.4 Piecewise linear error function .

$$E(r) \triangleq \begin{cases} 0 & for|r| \le \varepsilon, \\ |r| - \varepsilon & otherwise, \end{cases} \tag{4.11}$$

$$r(\mathbf{u}, v) \triangleq v - f(\mathbf{u}) \tag{4.12}$$

According (4.11) and (4.12), an ideal estimate of v satisfies $|r(\mathbf{u}, v)| \le \varepsilon$ so that the error $E(r)$ is 0. Thus, the training data which could lead to an ideal estimation lies in a band with width ε. This band is referred to as ε-insensitive zone. Figure 4.5 illustrates the ε-insensitive zone in the original space and the feature space. Figure 4.5(a) shows the approximation function in the original space that is abstractly depicted as a curve while the function in the feature space given by (4.10) is linear. Such transformation will benefit further analysis to find the optimal $\mathbf{w}$.



(a) (b)

Figure 4.5 (a) The ε-insensitive zone in the original space.

(b) The ε-insensitive zone in the feature space.

The choice $f(\mathbf{u})$ that satisfies $r(\mathbf{u}, v) \le \varepsilon$ is infinite. What we want to achieve is to find a solution with the maximum generalization ability. The training data that satisfies $r(\mathbf{u}, v) = \pm\varepsilon$ is farthest from the hyperplane $r(\mathbf{u}, v) = 0$. The distance from the hyperplane is called margin. Clearly, if we maximize the margin, we could have the best chance to have unknown data fallen into the ε-insensitive zone. Based on (4.10), the distance of the point

$(\mathbf{u},v)$ from the hyperplane $r(\mathbf{u},v)=0$ is $|r(\mathbf{u},v)|/\|\mathbf{w}^*\|$, where $\mathbf{w}^*$ is given by (4.13).

$$\mathbf{w}^* = (1,-\mathbf{w}^T)^T \tag{4.13}$$

The farthest point satisfies $|r(\mathbf{u},v)|/\|\mathbf{w}^*\| = \varepsilon$. Since $\|\mathbf{w}^*\|^2 = \|\mathbf{w}\|^2 + 1$, the minimum of $\|\mathbf{w}\|^2$ leads to the maximum value of ε, thus giving us the maximum margin. Therefore, we have such an optimization problem as given below.

$$\text{Minimize} \qquad \frac{1}{2}\|\mathbf{w}\|^2 \tag{4.14}$$

$$\text{Subject to} \qquad r(\mathbf{u},v) \le \varepsilon \tag{4.15}$$

Substitute (4.10) into (4.14)(4.15), we have

$$\text{Minimize} \qquad \frac{1}{2}\|\mathbf{w}\|^2 \tag{4.16}$$

$$\text{Subject to} \qquad \begin{matrix} v_i - \mathbf{w}^T\phi(\mathbf{u}_i) - b \le \varepsilon \\ \mathbf{w}^T\phi(\mathbf{u}_i) + b - v_i \le \varepsilon \end{matrix} \quad \text{for i=1,...,M} \tag{4.17}$$

In case some data fall outside of ε-insensitive zone, two independent non-negative slack variables ξ_i , ξ_i^* are introduced by

$$\xi_i = \begin{cases} 0 & \textit{for } D(\mathbf{u}_i,v_i) - \varepsilon \le 0, \\ \varepsilon + r(\mathbf{u}_i,v_i) & \textit{otherwise}, \end{cases} \tag{4.18}$$

$$\xi_i^* = \begin{cases} 0 & \textit{for } D(\mathbf{u}_i,v_i) + \varepsilon \ge 0, \\ -\varepsilon - r(\mathbf{u}_i,v_i) & \textit{otherwise}, \end{cases} \tag{4.19}$$

Taking slack variables into consideration, the optimization problem becomes

$$\text{Minimize} \qquad Q(\mathbf{w},b,\xi,\xi^*) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}(\xi_i + \xi_i^*) \tag{4.20}$$

$$\text{Subject to} \qquad \begin{matrix} v_i - \mathbf{w}^T\phi(\mathbf{u}_i) - b \le \varepsilon + \xi_i \\ \mathbf{w}^T\phi(\mathbf{u}_i) + b - v_i \le \varepsilon + \xi_i^* \\ \xi_i \ge 0, \quad \xi_i^* \ge 0 \end{matrix} \quad \text{for i=1,...,M} \tag{4.21}$$

where C is the weight of the slack variables determining the trade-off between the tolerance of the ε-insensitive zone and the estimation error of the training data.

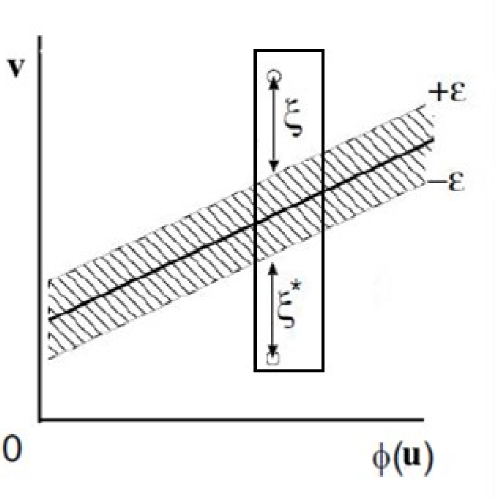


Figure 4.6 The slack variables.

To solve the optimization problem of (4.20), (4.21), Lagrange multipliers $\gamma_i, \gamma_i^*, \eta_i$ and η_i^* are introduced to transform the original constrained optimization problem into the following unconstrained optimization problem.

$$\text{Minimize} \quad Q(\mathbf{w}, b, \xi, \xi^*, \gamma, \gamma^*, \eta, \eta^*) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}(\xi_i + \xi_i^*)$$
$$- \sum_{i=1}^{M}\gamma_i(\varepsilon + \xi_i - v_i + \mathbf{w}^T\phi(\mathbf{u}_i) + b)$$
$$- \sum_{i=1}^{M}\gamma_i^*(\varepsilon + \xi_i + v_i - \mathbf{w}^T\phi(\mathbf{u}_i) - b) \qquad (4.22)$$
$$- \sum_{i=1}^{M}(\eta_i\xi_i + \eta_i^*\xi_i^*)$$

The saddle point of (4.22) can be found by letting the partial derivative with respect to $\mathbf{w}, b, \xi, \xi^*$ equal to zero. At the saddle point, the $\mathbf{w}$ satisfies

$$\mathbf{w} = \sum_{i=1}^{M}(\gamma_i - \gamma_i^*)\phi(\mathbf{u}) \qquad (4.23)$$

By substituting (4.23) into (4.10) an update of $f(\mathbf{u})$ is given by (4.24) with known $\gamma, \gamma^*, \eta, \eta^*$ at the saddle point.

$$f(\mathbf{u}) = \sum_{i=i}^{M} (\gamma_i - \gamma_i^*) \phi^T(\mathbf{u}_i) \phi(\mathbf{u}) + b \tag{4.24}$$

The problem characterized by (4.22) is further solved by applying Karush-Kuhn-Tucker (KKT) conditions on the dual problem, giving us the value of b as (4.25) and (4.26). Note that (4.25) and (4.26) give us multiple b's, the final b, which should be a constant, is the average of the b's.

$$b = v_i - \mathbf{w}^T \phi(u_i) - \varepsilon - \frac{\gamma_i}{C} \quad for\, \gamma_i > 0 \tag{4.25}$$

$$b = v_i - \mathbf{w}^T \phi(u_i) + \varepsilon + \frac{\gamma_i^*}{C} \quad for\, \gamma_i^* > 0 \tag{4.26}$$

The training data $\mathbf{u}_i$ with $0 < \gamma_i \le C$ or $0 < \gamma_i^* \le C$ are called support vectors because they contribute to the construction of the function (4.24), while those with $\gamma_i = \gamma_i^* = 0$ are not support vectors and have no influence on the approximation function. The approximation function (4.24) can also be rewritten as

$$f(\mathbf{u}) = \sum_{i=i}^{M} (\gamma_i - \gamma_i^*) K(\mathbf{u}, \mathbf{u}_i) + b \tag{4.27}$$

in which $K(\mathbf{u}, \mathbf{u}_i) = \phi^T(\mathbf{u}) \phi(\mathbf{u}_i)$ is called kernel. Commonly used kernels include linear kernel, polynomial kernel, Gaussian radial basis function (RBF) kernel, etc. The most commonly used kernel in SVR is the RBF kernel as given by

$$K(\mathbf{u}_i, \mathbf{u}) = \exp\left(-\frac{\|\mathbf{u}_i - \mathbf{u}\|^2}{2\sigma^2} \right) \tag{4.28}$$

The intuitive understanding of SVR is “similarity and voting”. From (4.27), we can see that the RBF kernel is a metric of similarity between two input variables, if they are the same, then their similarity is 1, otherwise it is small or close to 0. Therefore, the approximation is made by summing the “weighed similarities” contributed by all the support vectors.

4.2.2. SVR function training

The model training is similar to that of the polynomial regression, while the construction of the training data is different. In addition to the PC-CR vector and the screen coordinate, the training sample in the SVR method also includes the IPV and IC-IPCV vectors. According to (4.26), $\mathbf{u} = (\mathbf{V}_{\mathbf{PC\text{-}CR}}, \mathbf{V}_{\mathbf{IP}}, \mathbf{V}_{\mathbf{IC\text{-}IPC}})$ is the training input, $v_x = S_x$ and $v_y = S_y$ are the output. The model training is essentially a process of solving an optimization problem characterized by (4.22).

4.2.3. SVR based gaze estimation

As mentioned before, the SVR regression function input is $\mathbf{u} = (\mathbf{V}_{\mathbf{PC\text{-}CR}}, \mathbf{V}_{\mathbf{IP}}, \mathbf{V}_{\mathbf{IC\text{-}IPC}})$ instead of $\mathbf{u} = \mathbf{V}_{\mathbf{PC\text{-}CR}}$. The use of IPV and IC-IPCV vectors is based on the assumption that they reveal the head movement and are related to the PoR. In order to show that the additional features are eligible references to help estimate PoR, these features are recorded during a calibration process and are presented in Figure 4.7, which tells us how these features change as the user looks at different screen coordinates. The horizontal axis of the figure is the frame index which is divided into 9 periods, each corresponding to one of the 9 pre-set dots on the screen. The first period corresponds to the top-left dot in Figure 4.2, and the following ones correspond to the lower-right, lower-left, upper-right, left, right, top and bottom one, respectively. The combined vector $\mathbf{u} = (\mathbf{V}_{\mathbf{PC\text{-}CR}}, \mathbf{V}_{\mathbf{IP}}, \mathbf{V}_{\mathbf{IC\text{-}IPC}})$ is expressed by six scalars which are the vertical and horizontal components of the PC-CR, IPV and IC-IPCV vectors. The scaled magnitudes of the six scalars are shown as the vertical axis of the figure. Note that the variables are scaled for inspection purpose.

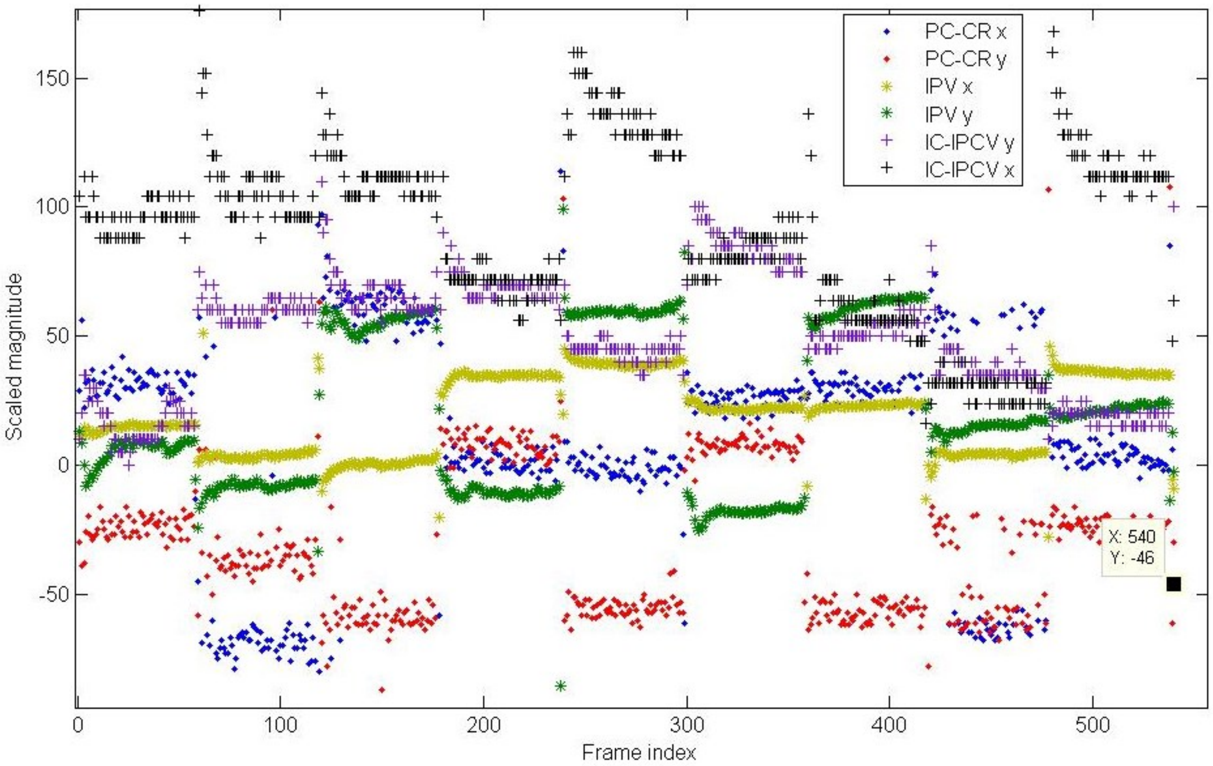


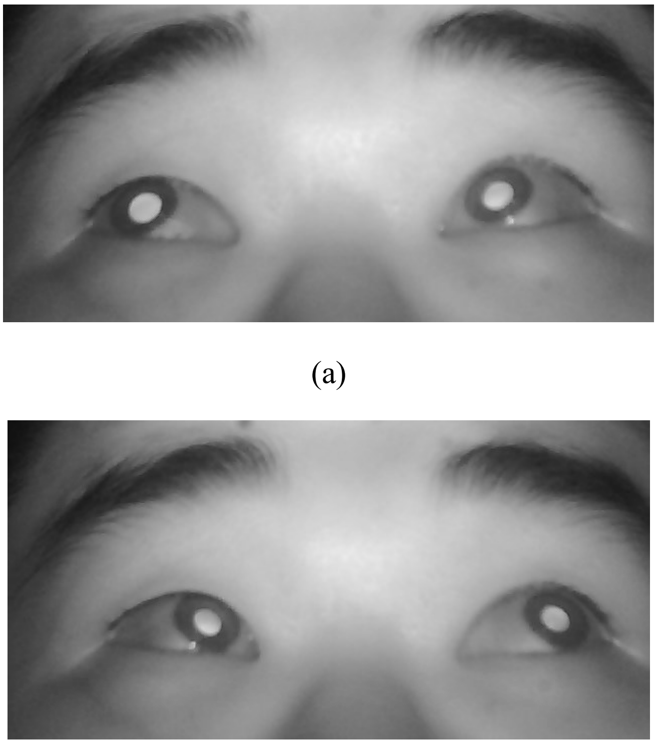
Figure 4.7 $\mathbf{V}_{\mathbf{PC\text{-}CR}}, \mathbf{V}_{\mathbf{IP}}, \mathbf{V}_{\mathbf{IC\text{-}IPC}}$ change as the viewer looks at different screen coordinates in 9 different periods.

We can discover from Figure 4.7 that PC-CR vector is very stable in each period and very distinctive between different periods. IPV and IC-IPCV are also distinctive between different periods but are less stable in each period. Precisely speaking, the IPV and IC-IPCV vectors in each period start with a level and end with a different level or converge to a stable level after a short while. The cause of such phenomenon is that, naturally, people are used to turning their head to a comfortable position and rotate eye ball in the meanwhile when there is any change of attention. Three activities are involved in such process: eye rotation, head rotation and head translation. Eye rotation, obviously is much faster than head rotation and translation, thus the PC-CR vector which depends on eye rotation converges almost in no time. Correspondingly, IPV and IC-IPCV which are more relevant to head rotation and head translation converge slowly. Even though, the correlation between the additional features and the PoR implies that they can help the estimation in natural head movement scenario by offering additional information on head pose.

4.3. Binocular gaze estimation

The gaze estimation method discussed in the previous section is based on one eye, which is referred to as monocular gaze estimation. In contrast, binocular gaze estimation requires the estimation on both eyes, which is achieved by averaging the estimated PoRs of both eyes. As compared to monocular gaze estimation, binocular method is more computational consuming, causing almost twice the computational time. However, the binocular gaze estimation is reported to be more accurate[2], especially when the light source(s) is (are) unsymmetrically placed.

Another important contribution of binocular gaze estimation is combating the "blind region" effect. When the angle between the camera optic axis and line of gaze is big enough the glint is no longer visible on the cornea surface. Since the camera in our system is placed near the bottom of the screen, the blind spots are the upper-left and upper-right corner of the screen. Figure 4.8 illustrates the eye region image when the viewer looks at the blind region. In Figure 4.8(a), we can find that the glint in the left eye is invisible when the viewer looks at the upper-left corner of the screen. Similarly, in Figure 4.8(b), the glint in the right eye disappears when the viewer looks at the upper-right corner of the screen. Therefore, sometimes the estimation of PoRs over the entire screen can not be achieved with monocular estimation. However, binocular estimation can deal with such occasion by relying on the other eye where the glint is visible. Noticeably, the blind region effect does not necessarily exist in every gaze tracking system. The presence of such effect depends on the location of the camera, degree of head movement, the viewer-screen distance and the viewer’s eye size.



(a)

(b)

Figure 4.8 (a) When the viewer looks at the upper-left corner of the screen, the glint in the left eye is invisible. (b) When the viewer looks at the upper-right corner of the screen, the glint in the right eye is invisible.

5. System implementation and testing

5.1. System implementation

The entire gaze tracking system consists of both hardware and software. The hardware includes the illumination and image acquisition board (IAIB) and the computer. The images captured by IIAB is imported into the computer using MATLAB image acquisition tool box. Real-time processing is achieved with MATLAB which allows us to display a moving cursor on the screen to represent the estimated PoR. Although MATLAB is not the ideal tool to maximize the processing rate, real time estimation can be realized at a frame rate of 30Hz, which is the maximum frame rate of the camera we use.

The computer used in the system is powered by an Intel Xeon E3-1230 V2 processor which has 4 physical cores, 8 virtual cores with a clock speed of 3.3 GHz. The RAM of the computer is a single slot 8GB DDR3 1600.

The screen used in the experiment is a 19-inch DELL 1907FP. The resolution is 1280-by-1024. The size of the display area is 376-by-301mm with an image pitch of 0.294mm.

The operation system of the computer is Windows 7 ultimate 64 bits. The version of MATLAB is R2013a(8.1.0.164).

5.2. Eye detection performance

In many eye localization works [29], the estimation accuracy is defined by the following normalized error,

$$e = \frac{\max\left(\left\|\mathbf{P}_r - \hat{\mathbf{P}}_r\right\|, \left\|\mathbf{P}_l - \hat{\mathbf{P}}_l\right\|\right)}{\left\|\mathbf{P}_r - \mathbf{P}_l\right\|} \tag{5.1}$$

where $\hat{\mathbf{P}}_l$ and $\hat{\mathbf{P}}_r$ are the estimated pupil center locations of the left and right pupils. The normalization error is a ratio of maximum estimation error of both eyes over the inter-pupil distance[29]. There are three classes of normalized errors, i.e. e<0.05, e<0.1, e<0.25. An error of 0.05 corresponds to a displacement as large as the pupil radius while that of 0.1 is as large as the iris radius and that of 0.25 is as large as half of the eye width. Gaze-tracking-oriented eye localization requires very accurate location of the pupil center since an accurate estimation of eye features depends on accurate location of the pupil center. Empirically, the magnitude PC-CR vector ranges between 0 to 1.5 pupil-diameter, thus an estimation of pupil center location with $e = 0.05$ is completely unacceptable. Therefore, in our experiment, an estimation of pupil center location with $e \le 0.01$ is regarded as a correct detection. And the eye detection rate is defined as the ratio of correction detections over the total detections.

The system is tested in 3 lighting conditions, i.e., dark indoor environment, normal indoor environment and outdoor environment with sufficient sunlight, while 3 different viewer-screen distances are tested. BGF method is used to find the pupil centers in a total of 1020 frames of images.

According to the tested eye detection rate given by Table 5.1, the eye detection rate decreases as the visible light interference increases. The cause of such phenomenon is the absence of visible light filter on the camera. When the system is exposed to intensive visible light, the contrast of the accentuated pupil decreases significantly causing more detection failures.

The accuracy rises as the viewer-screen distance increases since the desired bright-pupil effect is more evident on large distance. Therefore, the accentuated pupil image has good contrast and remain robust against noise. Within a distance of 400mm, the intensity of bright pupil drops dramatically. We also find that the intensity varies between individuals, some people's pupils are still bright around 350mm, while some others' go dark around 400mm.

Eye detection rate (e<0.005)			
Viewer-screen distance	Dark indoor	Normal indoor	Outdoor
400mm	95.1%	96.9%	72.5%
500mm	98.2%	97.5%	80.9%
600mm	**98.3%**	**98.0%**	**85.0%**

Table 5.1 Eye detection rate using BGF filter.

A comparison of three filtering methods for eye localization is given in Table 5.2. The performance metric is chosen to be the angular error of the estimated PoR since a better estimation of PoR indicates a better estimation of pupil center. Another reason for making this indirect comparison is made indirectly is that the true pupil center location is unknown, unless we use additional method to determine it.

According to Table 5.2, the minimum error comes from BGF while the performance of Gaussian filter is very close. The worst result comes from the ideal circular filter mainly because its selection of pupil center is not always unique. By looking into the frames with obvious poorly estimated PoR, we found that the difference between BGF and Gaussian filter is caused by the frames in which the glint lies on the boundary of the pupil.

RMS Angular error (degree)	
BGF	0.81
Gaussian filter	0.89
Ideal circular filter	1.71

Table 5.2 Angular error of estimated PoR derived based on different eye localization methods.

Another concern is the influence of rapid scene changes including fast user movement and background changes like people moving behind the user. Figure 5.1 shows an accentuated bright pupil image acquired during fast head movement, in which the pupils are not ideally segmented.Consequently, the pupil center might not be correctly located due to heavy interference. As a matter of fact, experiment shows that natural head movement involved in daily use of computer degrade the eye detection rate by around 0.7% at a distance of 600m in the normal indoor environment.

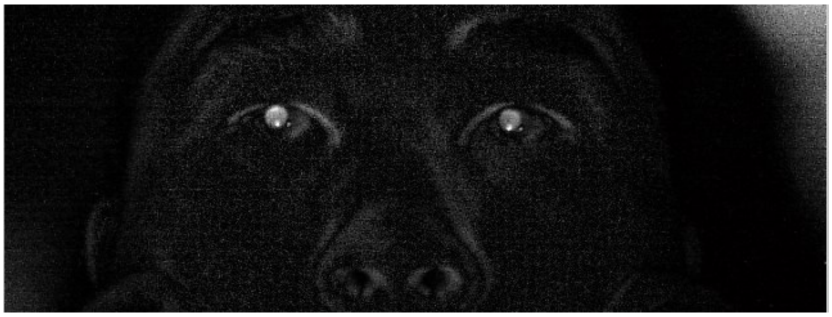


Figure 5.1 The accentuated pupil image interfered by fast head movement.

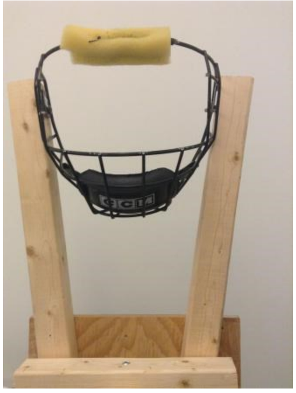


Figure 5.2 The headrest to restrict head movement.

5.3. Gaze estimation without head movement

It is very common in polynomial regression works to use a bite bar or head rest to restrict head motion. Our system is also tested with fixed head pose using a headrest shown in Figure 5.2. The head movement is restricted by asking the user to put his/her chin on the chin guard and forehead on the foam.

The common evaluation metric of gaze estimation accuracy is the mean error, standard deviation and root mean square (RMS) angular error of the estimated PoR. The mean error of Sx and Sy are given by (5.2) and (5.3), where N is the size of the testing set. The standard deviation σ_{Sx} and σ_{Sy} are given by (5.4) and (5.5), in which μ_{Sx} and μ_{Sy} are the mean value of Sx and Sy around a calibration point. The horizontal and vertical RMS angular error θ_x and θ_y are given by (5.6) and (5.7), where d is the distance between the user and the screen. The angular error reveals the estimation accuracy most directly and remains as the universal specification for performance evaluation.

$$ME_{Sx} = \frac{1}{N}\sum_{i=1}^{N}\left|Sx_i - \hat{S}x_i\right| \tag{5.2}$$

$$ME_{Sy} = \frac{1}{N}\sum_{i=1}^{N}\left|Sy_i - \hat{S}y_i\right| \tag{5.3}$$

$$\sigma_{Sx} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{S}x_i - \mu_{Sx}\right)^2} \tag{5.4}$$

$$\sigma_{Sy} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{S}y_i - \mu_{Sy}\right)^2} \tag{5.5}$$

$$\theta_x = \arctan\left(\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Sx_i - \hat{S}x_i\right)^2}}{d}\right) \tag{5.6}$$

$$\theta_y = \arctan\left(\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Sy_i - \hat{S}y_i\right)^2}}{d}\right) \tag{5.7}$$

In order to compare the performance of different methods, the same recorded image sequences with 405 and 2127 frames are used as the training and testing set. The training set is acquired using a 3-by-3 calibration grid. The testing set is acquired in a similar way as the training process, while a 5-by-5 grid as shown in Figure 5.3 is used instead of the 3-by-3 one shown in Figure 4.2. Thus, the testing set includes more samples corresponding to more screen points.

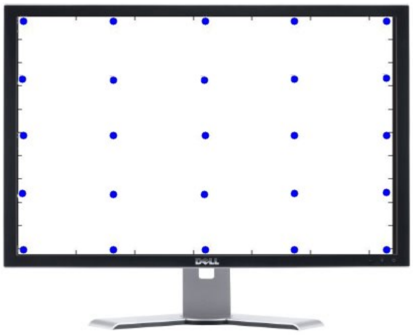


Figure 5.3 The 5-by-5 testing grid on the screen.

The performance of different regression methods is presented in Table 5.3, where $R_i(i=1,2,3)$ represents the regression function (4.1)-(4.3), M and B refers to monocular and binocular estimations, respectively. Since the IPV and IC-IPCV vectors are not available in such situation, the SVR function is trained and tested with the PC-CR vector alone.

Method	Mean error (mm)		σ (mm)		RMS angular error (degree)	
	Vertical	Horizontal	Vertical	Horizontal		
R_1,M	5.38	**5.51**	**5.97**	**7.25**	**0.77**	**0.87**
R_2,M	**5.19**	5.67	6.03	8.08	0.76	0.94
R_3,M	5.60	6.10	6.34	8.27	0.81	0.98
R_1,B	**3.16**	**3.44**	**5.13**	**6.09**	**0.57**	**0.67**
R_2,B	3.80	3.77	5.41	6.54	0.63	0.72
R_3,B	3.93	4.01	5.43	6.73	0.64	0.75
SVR,B	4.82	5.38	6.61	6.61	0.78	0.81

Table 5.3 The performance of tested methods using 3-by-3 training grid and 5-by-5 testing grid.

According to Table 5.3, the accuracy of binocular method is higher than the monocular method regardless of the regression function. As to three polynomial functions, R_1 which is the full second order polynomial (4.1) outperforms R_2 and R_3 while the accuracy of R_2 and R_3 are very close. The SVR method shows no advantage comparing to polynomial regression methods under such circumstance.

Another experiment is carried out by using a 5-by-5 training grid and a training set with 1125 frames of images to see whether an advanced training set with more calibration points and training samples improves the estimation. According to the result listed in Table 5.4, the new training set has improved the estimation of polynomial method slightly as evidenced by decreasing the angular error by around 0.03°. Meanwhile, the SVR based estimation has been improved considerably by around 0.1°. It can be concluded that more training data can improve the SVR method but their contribution is negligible for polynomial functions. For our system, the best method under the head restrained situation is the binocular method using full second order polynomial.

Method	Mean error (mm)		σ (mm)		RMS angular error (degree)	
	Vertical	Horizontal	Vertical	Horizontal		
R_1, M	5.16	**5.50**	**5.70**	**6.98**	**0.73**	**0.85**
R_2, M	**4.96**	5.56	5.76	7.56	0.72	0.89
R_3, M	5.37	5.97	6.06	7.74	0.77	0.93
R_1, B	**3.03**	**3.38**	**4.90**	**5.7**	**0.55**	**0.63**
R_2, B	3.66	3.69	5.17	6.12	0.60	0.68
R_3, B	3.77	3.93	5.19	6.3	0.61	0.70
SVR, B	3.87	4.30	5.36	6.18	0.63	0.71

Table 5.4 The performance of tested methods using 5-by-5 training grid and 5-by-5 testing grid.

5.4. Gaze estimation with head movement

In the following experiments, the user is allowed to move his/her head freely without using a headrest while keeping his/her shoulder still. In such a situation, the rotation of eyes, the translation and rotation of head are expected when eye gaze changes.

An experiment is firstly conducted with the 3-by-3 training grid and 5-by-5 testing grid.

From Table 5.5 we can notice that the performance of polynomial and SVR methods with head movement is worse than that with fixed head, in particular, the accuracy of polynomial methods decreases around 1° while the the degradation of SVR method is more evident which is round 1.2°.

Another experiment is conducted using the 5-by-5 training grid. From the result demonstrated in Table 5.6, the improvement on polynomial regression method is negligibly around 0.05° while it is significant on SVR method with a boost of approximately 1.1°. As a conclusion, the number of calibration points is an important factor for SVR method. More calibration points are desirable as long as the training process does not bother the user's experience.

Method	Mean error (mm)		σ (mm)		RMS angular error (degree)	
	Vertical	Horizontal	Vertical	Horizontal		
R_1, M	**14.9**	**16.2**	**15.7**	**18.3**	**2.07**	**2.33**
R_2, M	16.4	17.9	17.3	20.1	2.28	2.57
R_3, M	15.9	17.3	16.8	19.5	2.21	2.48
R_1, B	**10.8**	**11.7**	**11.4**	**13.3**	**1.50**	**1.69**
R_2, B	12.1	13.1	12.8	14.7	1.68	1.88
R_3, B	12.7	13.8	13.4	15.6	1.76	1.98
SVR, B	14.0	15.2	14.7	17.1	1.94	2.18

Table 5.5 The performance of tested methods with head movement using 3-by-3 training grid and 5-by-5 testing grid.

Method	Mean error (mm)		σ (mm)		RMS angular error (degree)	
	Vertical	Horizontal	Vertical	Horizontal		
R_1, M	**14.9**	**17.2**	**14.3**	**17.0**	**1.97**	**2.31**
R_2, M	16.3	18.9	15.7	18.7	2.17	2.54
R_3, M	15.8	18.3	15.3	18.1	2.10	2.46
R_1, B	**10.8**	12.7	**10.4**	**12.3**	1.43	**1.67**
R_2, B	12.1	**12.5**	11.6	13.8	**1.42**	**1.67**
R_3, B	12.7	14.0	12.2	14.5	1.68	1.87
SVR, B	**6.30**	**7.71**	**5.64**	**6.63**	**0.87**	**0.97**

Table 5.6 The performance of tested methods with head movement using 5-by-5 training grid and 5-by-5 testing grid.

5.5. Multi-user test

Multi-user test is conducted in order to see how the performance of the proposed methods varies among different individuals since factors like eye size, ethnics, looking habit may differ between individuals. The system was tested by 6 people, of which 2 are Caucasians, 2 Asians, 1 Arab and 1 Indian. All of them are tested without wearing eye glasses. Note that, for simplicity, the RMS angular error in the table is based on Euclidean distance instead of the vertical and horizontal distance. The polynomial regression method used is the binocular estimation with full second-order function.

User	Angular error (degree) No head movement		Angular error (degree) Free head movement	
	Polynomial	SVR	Polynomial	SVR
1	0.84	0.91	2.26	1.12
2	0.93	1.07	2.97	2.11
3	1.22	1.35	2.45	1.37
4	0.85	0.93	2.97	2.15
5	0.91	0.98	3.53	3.11
6	1.05	1.18	3.00	2.60

Table 5.7 Multiuser test results.

The test result given in Table 5.7 shows that the performance between different users is different. With SVR method, the difference between users under free head movement scenario is greater. This is due to the limitation of the method. As we all know, one can look at a point naturally with both head movement and eye movement , or uncomfortably with more head (eye) movement and less eye (head) movement. In case that the user choose to move his/her head and eye unnaturally, the estimation error would increase since the regression function is trained under the natural coordination of eye and head movements. One solution to this problem is to train the function under different head poses. In other words, during the calibration process the user has to move his/her head while keeping his/her gaze on the same dot so that the training is adapted to the user's viewing habit. However, such solution increases the size of the support vectors dramatically and greatly increases the computational cost.

5.6. Comparison with state-of-the-art works

Among many recent contributions on gaze estimation, only regression based works that allow free head movement are compared here. The performance of this work in the following table is obtained by binocular SVR method with free head movement, while we choose to neglect the simple polynomial regression method since its performance is only acceptable when the user's head is restrained, thus remaining very unpractical.

Related works	Angular error (degree)
This work	1.12
Zhu *et al* [1]	0.77
Sesma-Sanchez *et al* [2]	< 1
Zhu *et al* [7]	1.5
Lu *et al* [28]	1.0-1.5
Martinez *et al* [17]	2

Table 5.8 Comparison with other state-of-the-art works

Table 5.8 shows a comparison of the proposed system with the regression based state-of-the-art work. The best result comes from [1], in which the authors used polynomial regression method with a geometric information based compensation function to combat the head movement impact. The authors of [2] also used polynomial regression method but limited the freedom of the head movement to only one dimension to achieve an accuracy of below 1°. As a matter of fact, it is very difficult to achieve such accuracy solely depending on polynomial regression when 3-dimensional head translation and rotation is allowed. The authors of [7], [28] and [17] all used SVR as the regression method while relevance vector regression is also used in [17]. In [7], the PC-CR vector and the 3D coordinate of the eyes are used as SVR function input while the PC-CR vector and the so-called local-binary-pattern texture feature are used as input in [28]. Although both of these methods can achieve a similar accuracy to ours, their approaches are more complicated. The tracking system of [7] requires additional calculation of 3D coordinate of the eyes, while that of [28] has a feature dimension of 531, which makes the optimization process to approximate the function rather slow and difficult. Different from other works, the tracking system of [17] does not depend on IR illumination but works in natural light environment. Without using the PC-CR vector, the authors use histograms of the gradients (HOG) of the eye image as the regression function input and choose the SVR and relevance vector regression as regression function. Since the features such as HOG acquired under natural light is far less sensitive than those acquired under IR light, it is very difficult to accurately estimate PoR with natural light.

As a conclusion, our system is comparable to the regression based state-of-the-art works in tracking performance while it only uses a simple webcam and easy-to-calculate image plane features instead of complex geometric calculation, thus reducing the implementation cost to a large extent.

6. Conclusions

6.1. Summary

This thesis work has been focused on estimating human gaze with video oculography with a goal to develop a real-time gaze tracking system that consists of hardware and software.

The first chapter introduced the fundamentals of gaze tracking techniques and reviewed relevant literatures. It has been pointed out that eye detection and gaze estimation are the major tasks in gaze tracking.

Chapter 2 focused on eye detection. The flowchart of the entire system was presented in the beginning of the chapter. The four major steps to detect eyes and calculate eye features were then explained following the design of the flowchart. A novel boosted Gaussian filter based voting scheme is proposed to locate pupil center and glint center. Additional features, i.e. IPV and IC-IPCV vectors are calculated to help the proposed gaze estimation method.

In chapter 3, two different gaze estimation methods, i.e. polynomial regression method and SVR method were studied. Different regression polynomials have been reviewed and compared in order to select the optimal one for our system. SVR which is an effective tool to approximate nonlinear functions is chosen to estimate the PoR based on image plane eye features. With additional head pose related features, SVR method is robuster than traditional polynomial regression methods in free head movement scenario. Binocular gaze estimation technique is also introduced in our system to make the estimation more accurate and reliable.

Finally in chapter 4, the experimental setups and test results are discussed. The hardware which is mainly composed of the camera and the illumination circuit is introduced at the beginning. Then, the eye detection subsystem is tested, showing that the best estimation result of pupil center was achieved by the BGF method in the dark indoor environment. Latter, the test result of gaze estimation with headrest is presented. The SVR method also works with a headrest but does not have advantage over the traditional methods. In the test with natural head movement, the accuracy of traditional methods dropped to 1.42° from 0.55° vertically and 1.67° from 0.83° horizontally, however, the accuracy of SVR only decreased slightly to 0.87° vertically and 0.97° horizontally from 0.63° and 0.71°, respectively. Additionally, the system was tested on multiple users and was proved to be reliable.

It can be concluded that the traditional polynomial regression method performs well without head movement but degrades significantly when head movement is allowed. In a head restrained situation, the polynomial method is the best choice since it outperforms the SVR method both in accuracy and computational complexity. In a free head movement situation, the SVR method is more accurate but requires sufficient training samples and calibration points. Nevertheless, the estimation with head movement is worse than that with still head for both methods.

Moreover, it should be noted that the proposed method does not rely on geometric information of pupils or head. It offers satisfactory gaze estimation accuracy with low implementation cost by using only image plane information under the free head movement scenario.

6.2. Original contributions

This work is inspired by several existing studies. The image differencing technique used to make pupil distinctive is based on the method proposed by Morimoto et al [5]. The classic PC-CR based polynomial regression methods were contributions from Merchant et al. [2,4,6,28]. Making gaze estimation with SVR using geometric features was originally proposed by Zhu et al [7]. Our new system has been developed based on the available state-of-the-art works. Yet it has employed the following innovative approaches which are proposed during this study.

- A boosted Gaussian filter based filter matching scheme is proposed to locate pupil center and glint center.
- Precise PC-CR vector estimation is realized via eye region image interpolation.
- SVR along with image plane features is used to estimate PoR more accurately and reliably in a natural head movement scenario.
- An illumination and image acquisition circuit board with synchronization signal controlled by computer sound card is developed.

6.3. Future work

One short coming of this system is that the SVR method could not work in truly real time due to the heavy computational load, especially when the training set is large. Our computer, powered by Intel Xeon-E3 1230 V2, can process 4 frames per second under the circumstances that the size of the training set is 600 frames. The problem can not be solved by solely upgrading the computer. One solution is to program with C++ instead of using Matlab since C++ is much more efficient.

Another improvement that could be done is the camera. The maximum frame rate of our camera is 30 Hz, thus the maximum estimation rate is 15 Hz since each estimation is made with 2 frames. Moreover, the captured images are often contaminated by insufficient exposure, causing many dark horizontal bands which block part of the image. Instead, we can use industrial camera to increase the image acquisition rate and improve the image quality.

The dual-rings allocation of LEDs is not the optimal scheme to create bright and dark pupils in the image. The existence of outer ring LEDs makes it difficult to place the camera close to the screen, thus incurring an easy loss of the glint in the captured image. It should also be noticed that most modern commercial products such as Gazepoint, Mirametrix and myGaze avoid to use ring-like LEDs allocation but favor the two-sided allocation, in which some LEDs are placed closely to the camera and the others are placed remotely on the two sides of it.

References

[1] Z. Zhu and Q. Ji, “Novel eye gaze tracking techniques under natural head movement,” Biomedical Engineering, IEEE Transactions on, vol.54, no.12, pp.2246-2260, Dec. 2007.

[2] L. Sesma-Sanchez, A. Villanueva and R. Cabeza, “Gaze estimation interpolation methods based on binocular data,” Biomedical Engineering, IEEE Transactions on , vol.59, no.8, pp.2235-2243, Aug. 2012.

[3] D.W. Hansen and Q. Ji, “In the eye of the beholder: A survey of models for eyes and gaze,” Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.32, no.3, pp.478-500, March 2010.

[4] J. Cerrolaza, A. Villanueva, and R. Cabeza, “Taxonomic study of polynomial regressions applied to the calibration of video-oculographic systems,” in Proc. Symp. Eye Tracking Res. Appl., ACM, 2008, pp. 259–266.

[5] C. Morimoto, D. Koons, A. Amir, M. Flickner, “Pupil detection and tracking using multiple light sources,” Image Vis. Comput. 18 (4) (2000) 331–336.

[6] C. Morimoto and M. Mimica, “Eye gaze tracking techniques for interactive applications,” Computer Vision and Image Understanding 98 (2005) 4–24.

[7] Z. Zhu, Q. Ji and K.P. Bennett, “Nonlinear eye gaze mapping function estimation via support vector regression,” Pattern Recognition, 2006. ICPR 2006. 18th International Conference on , vol.1, no., pp.1132-1135, 0-0 0.

[8] Q. Ji and Z. Zhu, “Eye and gaze tracking for interactive graphic display,” Proc. Second Int’l Symp. Smart Graphics, pp. 79-85, 2002.

[9] R. Valenti and T. Gevers, “Accurate eye center location through invariant isocentric

patterns," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.34, no.9, pp.1785-1798, Sept. 2012.

[10] K.P. White Jr., T.E. Hutchinson, and J.M. Carley, "Spatially Dynamic Calibration of an Eye-Tracking System," IEEE Trans. Systems, Man, and Cybernetics, vol. 23, no. 4, pp. 1162-1168, July/ Aug. 1993.

[11] D. Young, H. Tunley, and R. Samuels, "Specialised hough transform and active contour methods for real-time eye tracking," Technical Report 386, School of Cognitive and Computing Sciences, Univ. of Sussex, 1995.

[12] C. Hennessey, B. Noureddin, and P. Lawrence, "A single camera eye-gaze tracking system with free head motion," Proc. Symp. Eye Tracking Research and Applications, pp. 87-94, 2006.

[13] C.A. Hennessey and P.D. Lawrence, "Improving the accuracy and reliability of remote system-calibration-free eye-gaze tracking," Biomedical Engineering, IEEE Transactions on , vol.56, no.7, pp.1891-1900, July 2009.

[14] J. Chi, C. Zhang, Y. Yan, Y. Liu and H. Zhang, "Eye gaze calculation based on nonlinear polynomial and generalized regression neural network," Natural Computation, 2009. ICNC '09. Fifth International Conference on , vol.3, no., pp.617-623, 14-16 Aug. 2009.

[15] D. Beymer, D, M. Flickner, "Eye gaze tracking using an active stereo head," Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on , vol.2, no., pp. II- 451-8 vol.2, 18-20 June 2003.

[16] J. Canny, "A computational approach to edge detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.PAMI-8, no.6, pp.679-698, Nov. 1986.

[17] F. Martinez, A. Carbone and E. Pissaloux, “Gaze estimation using local features and non-linear regression,” Image Processing (ICIP), 2012 19th IEEE International Conference on , vol., no., pp.1961,1964, Sept. 30 2012-Oct. 3 2012.

[18] D. Tweed and T. Vilis, “Geometric Relations of Eye Position and Velocity Vectors during Saccades,” Vision Research, vol. 30, no. 1, pp. 111-127, 1990.

[19] J. Merchant, R. Morrissette, and J. Porterfield, “Remote Measurements of Eye Direction Allowing Subject Motion over One Cubic Foot of Space,” IEEE Trans. Biomedical Eng., vol. 21, no. 4, pp. 309-317, July 1974.

[20] T. D’Orazio, M. Leo, G. Cicirelli, and A. Distante, “An Algorithm for Real Time Eye Detection in Face Images,” Proc. 17th Int’l Conf. Pattern Recognition, vol. 3, no. 0, pp. 278-281, 2004.

[21] A. Tomono, M. Iida, Y. Kobayashi, “A TV camera system which extracts feature points for non-contact eye movement detection,” Proceedings of the SPIE Optics, Illumination, and Image Sensing for Machine Vision IV, 1989, vol. 1194, pp. 2–12.

[22] Y. Ebisawa, S. Satoh, “Effectiveness of pupil area detection technique using two light sources and image difference method,” in A.Y.J. Szeto, R.M. Rangayan (Eds.), Proceedings of the 15th Annual International Conferemce of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, 1993, pp. 1268–1269.

[23] A. Fitzgibbon, M. Pilu, and R. Fisher, “Direct least square fitting of ellipses,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 476-480, May 1999.

[24] Z. Zhu and Q. Ji, “Robust Real-Time Eye Detection and Tracking under Variable

Lighting Conditions and Various Face Orientations," Computer Vision and Image Understanding, vol. 98, no. 1,special issue on eye detection and tracking, pp. 124-154, 2005.

[25] E.D. Guestrin and M. Eizenman, "General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections," IEEE Trans. Biomedical Eng., vol. 53, no. 6, pp. 1124-1133, June 2006.

[26] A.C. Rencher and W.F. Christensen, "Chapter 10, Multivariate regression – Section 10.1, Introduction," Methods of Multivariate Analysis, Wiley Series in Probability and Statistics 709 (3rd ed.), John Wiley & Sons, p. 19, ISBN 9781118391679.

[27] S. Abe. Support Vector Machines for Pattern Classification. Springer-Verlag, New York,2005.

[28] Hu-chuan Lu, Chao Wang and Yen-wei Chen, "Gaze tracking by Binocular Vision and LBP features," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on , vol., no., pp.1,4, 8-11 Dec. 2008.

[29] O. Jesorsky, K.J. Kirchbergand, and R. Frischholz, "Robust Face Detection Using the Hausdorff Distance," Proc. Third Int'l Conf. Audio and Video-Based Biometric Person Authentication, pp. 90-95, 1992.

[30] K. Hyoki, M. Shigeta, N. Tsuno, Y. Kawamuro, and T. Kinoshita, "Quantitative Electro-Oculography and Electroencephalography as Indices of Alertness," Electroencephalography and Clinical Neurophysiology, vol. 106, pp. 213-219, 1998.

[31] L.H. Yu, and E. Eizenman, "A new methodology for determining point-of-gaze in head-mounted eye tracking systems," Biomedical Engineering, IEEE Transactions on ,

vol.51, no.10, pp.1765,1773, Oct. 2004.

[32] Keith Rayner, “Eye movements in reading and information processing: 20 years of research”, Psychological Bulletin, pp.372-422, 1998.

[33] G. Anders, “Pilot’s Attention Allocation during Approach and Landing—Eye- and Head-Tracking Research,” Proc. 11th Int’l Symp. Aviation Psychology, 2001.

[34] J.H. Goldberg and A.M. Wichansky, Eye Tracking in Usability Evaluation: A Practitioner’s Guide, pp. 493-516. Elsevier Science, 2003.

[35] H. Knight and R. Simmons, “Estimating human interest and attention via gaze analysis,” Robotics and Automation (ICRA), 2013 IEEE International Conference on , vol., no., pp.4350,4355, 6-10 May 2013.

[36] H.K.Su and Y.K. Min, “Head-mounted binocular gaze tracker as a human-robot interfacing device,” RO-MAN, 2013 IEEE , vol., no., pp.374,375, 26-29 Aug. 2013.

[37] L.C. Loschky And G.W. McConkie, “User performance with gaze contingent multiresolutional displays”, Proc. Eye Tracking Research and Application Symposium (ETRA’00), pp 97-103, ACM, 2000.

[38] Q. Ji and X. Yang, “Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance,” Real-Time Imaging, vol. 8, no. 5, pp. 357-377, 2002.

[39] L.J.G. Vazquez, M.A. Minor and A.J.H. Sossa, “Low cost human computer interface voluntary eye movement as communication system for disabled people with limited movements,” Health Care Exchanges (PAHCE), 2011 Pan American , vol., no., pp.165,170, March 28 2011-April 1 2011.

[40] D.J. Ward and D.J.C. MacKay, “Fast Hands-Free Writing by Gaze Direction,” Nature,

vol. 418, no. 6900, p. 838, 2002.

[41] A. Duchowski, Eye Tracking Methodology: Theory and Practice. Springer-Verlag, 2003.

[42] C. Topal, S. Gunal, O. Kocdeviren, A. Dogan and O.N. Gerek, “A Low-Computational Approach on Gaze Estimation With Eye Touch System,” Cybernetics, IEEE Transactions on , vol.44, no.2, pp.228,239, Feb. 2014.

[43] C. Topal, O. N. Gerek, and A. Dogan, “A head-mounted sensor-based eye tracking device: Eye touch system,” in Proc. ACM Symp. Eye Tracking Res. Appl., Savannah, GA, USA, 2008, pp. 87–90.

[44] C. Topal, A. Dogan, and O. N. Gerek, “A Wearable head-mounted sensor-based apparatus for eye tracking applications,” in Proc. IEEE Conf. Virtual Environ, Human-Comput. Interfaces Meas. Syst., Istanbul, Turkey, Jul. 2008, pp. 14–16.

[45] David A. Atchison, George Smith, “Optics of the human eye”, Butterworth-Heinemann, 2000.

[46] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009.

[47] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.4, pp.509,522, Apr 2002.

[48] Saad A. Sirohey and Azriel Rosenfeld, “Eye detection in a face image using linear and nonlinear filters,” Pattern Recognition, Volume 34, Issue 7, 2001, Pages 1367-1391.

[49] J.S. Agustin, A. Villanueva, and R. Cabeza, “Pupil Brightness Variation as a Function of

Gaze Direction," Proc. 2006 Symp. Eye Tracking Research and Applications, pp. 49-49, 2006.

[50] S. W. Shih and J. Liu, "A novel approach to 3-D gaze tracking using stereo cameras," IEEE Trans. Syst. Man, Cybern. Part-B, vol. 34, no. 1, pp. 234–245, Feb. 2004.

[51] G.L. Lohse, "Consumer Eye Movement Patterns on Yellow Pages Advertising", Journal of Advertising, 26(1), pp.61-73,1997.

Appendix A. Anatomy of human eye

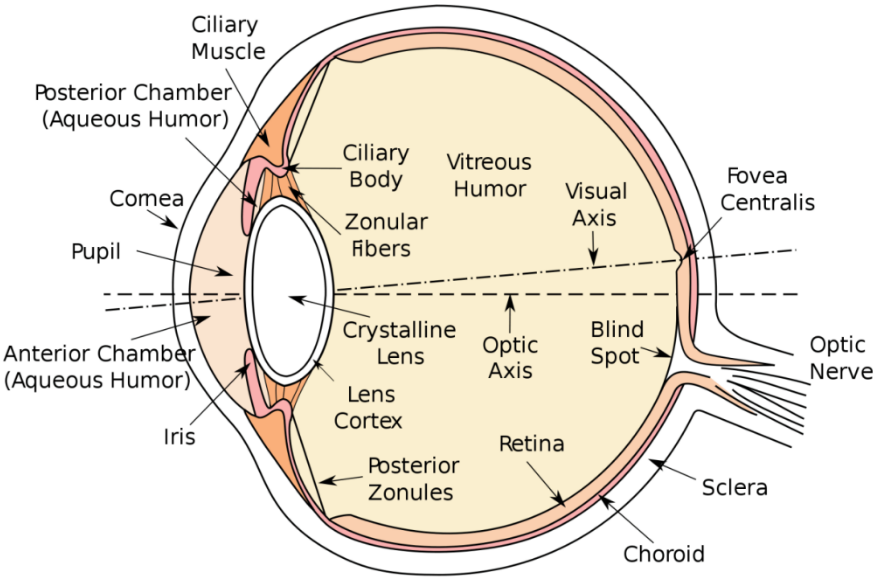


Figure A.1 Overhead diagram of the right eye. By Sathiyamoorthy.

The cornea is the outmost layer of the eye that covers the iris, pupil and anterior chamber, which is filled with transparent fluid called aqueous humor [45]. The pupil is a hole located in the center of the eye and is surrounded by iris, it appears to be black as most of the light going through is absorbed by the retina. The fovea centralis is located on the retina and is responsible for sharp central vision. The vast eyeball area between the lens and retina is filled with vitreous humor.

The visual axis, also known as line of sight (LoS), is defined as a line connecting the center of retina and fovea. The optic axis, also known as line of gaze (LoG), is the line connecting pupil center, cornea center and the eyeball center. They do not collide with each other, the angle between them is called angle kappa, which is approximately 4°. Since the fovea is the most important and sensitive area on retina to sense light, the visual axis, rather than optic axis, determines the true direction of gaze.

Reflections on cornea and lens are known as Purkinje images[41]. There are totally four reflections, the first reflection is from surface of the cornea and is the most evident one, it is commonly referred to as "glint".

Appendix B. IR LED safety

According The European Standard EN 60825-1:2007, long time eye exposure to IR light might hurt the retina. The standard regulates the maximum permissible exposure (MPE) by total energy in Joule over a period of time.

There are different equations to calculate the MPE depending on the continuous exposure time. The MPE, on condition that the exposure lasts from 10 seconds to 8.3 hours, is given by

$$MPE(t) = 3.5 \times 10^{-3} t^{0.75} C \quad for\, 10 < t < 3 \times 10^{4} \tag{*}$$

where t is the duration of the exposure in seconds and C is the coefficient depending on the IR light wavelength and the angle of acceptance. By applying the most strict restriction on the angle of acceptance, the C is 1.9953 on condition that the wavelength is 850nm. Therefore, using (*), the MPE over a period of 8 hours (28800 seconds) is 15.44J.

The amount of IR light that the user's retina receives is regulated by system exposure (SE), which is given by

$$SE(t) = P\psi t \tag{\#}$$

where P is the total radiant intensity of LED diodes in W/Sr, ψ is the square radian of pupil in Sr and t is also the duration in seconds. On condition that the radiant intensity of a single diode is 30mW/Sr, the total radiant intensity of our system is 300mW/Sr with totally 10 LED diodes. Allowing for the fact that the common pupil diameter is 7mm, the square radian ψ equals $1.069 \times 10^{-4} Sr$ at a 600mm user-screen distance. According to (#), the SE over 8 hours is 0.922J, which is far less than the MPE.