

# Eye Movement Tracking for Diagnostic Systems



Xindian Long

*bimagicLab*  
*Center for Bioimage Informatics*  
*Dept. of Electrical & Computer Engineering*  
*Carnegie Mellon University*

# Eye Movement Tracking for Diagnostic Systems

Xindian Long

*Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy by the Department of Electrical & Computer Engineering, Carnegie Inst. of Tech., Carnegie Mellon University.*

## Thesis Committee Member

Prof. Ozan K. Tonguz (Advisor)  
*Department of Electrical & Computer Engineering  
Carnegie Mellon University*

Prof. Jelena Kovačević (Co-advisor)  
*Departments of Biomedical Engineering and  
Electrical & Computer Engineering  
Carnegie Mellon University*

Dr. Alex. Kiderman  
*Neuro Kinetics, Inc. Pittsburgh, PA*

Prof. Vijayakumar Bhagavatula  
*Department of Electrical & Computer Engineering  
Carnegie Mellon University*

# Acknowledgments

I would like to express my deep and sincere gratitude to my advisor, Prof. Ozan K Tonguz. It has been an honor to be his Ph.D. student. He is responsible for involving me in the eye tracking project in the first place. He showed me different ways to approach a research problem. He taught me how good research is done and the need to be persistent to accomplish a goal. I appreciate all his contributions of time, ideas to make my Ph.D. experience productive and stimulating.

A special thanks goes to Dr. Alex Kiderman, with whom I explored the ideas and discussed the implementation of the eye tracking system. He is always readily available to discuss a problem, test an idea, or comment on my papers. He helped me to more deeply understand the requirements and impacts of the project. He has provided for me an excellent example of a successful and dedicated scientist.

I am deeply grateful to my co-advisor, Prof. Jelena Kovačević, for guiding me through the research, for helping me complete the writing of this dissertation, and for her detailed and constructive comments. Jelena has been a friend and mentor. I am thankful for her constant encouragement and having my best interest at heart.

I would like to thank Prof. Vijayakumar Bhagavatula, who is in my thesis committee, for his support, the challenging questions and invaluable advice, and for being so flexible in time and reviewing the dissertation in a short notice.

I wish to extend my warmest thanks to my fellow PhD students and post doctor researchers working together, Paisarn Sonthikorn, Hsin-mu Tsai, Wantanee Viriyasitavat, Yi Zhang, Jiun-Ren Lin, Hsiao-Yu Fan, Mate Boban, Evsen Yanmaz, Nawaporn Wisitpongphan, Inci Özgünes, Gowri

Srinivasa, Ramu Bhagavatula , Pablo Hennings Yeomans, who have made my time at CMU enjoyable, and have been the source of friendship and good collaboration. Thank you for listening to my presentation, giving extremely helpful suggestions, fixing my car, and especially, for marking those images by hand, which is a very tedious task.

My sincere thanks are due to Lynn Philibi and Elaine Lawrence, who have been a tremendous help in the process of my getting back into the graduate program.

I owe my loving thanks to my family; my parents, Haotian Long and Shuyuan Cui, I am thankful for their unconditional support and encouragement to pursue my interests; my sister, Jiali, Sirui and their family, I am grateful for their loving support and understanding; My special gratitude is due to my husband, Peixing Sun, for listening to my complaints and frustrations, and for believing in me.

This work was funded by Neuro Kinetics, Inc., which is gratefully acknowledged.

# Contents

<b>I</b>	<b>Introduction</b>	<b>2</b>
<b>1</b>	<b>Background and Motivation</b>	<b>3</b>
1.1	Eye Dynamics, Related Medical Applications and Eye Tracking Systems . . . . .	3
1.2	Related Work and the Need for a New System . . . . .	6
<b>II</b>	<b>Calibration Method and Simulation System for Eye Tracking</b>	<b>10</b>
<b>2</b>	<b>The Calibration Method</b>	<b>11</b>
2.1	Background and Motivation . . . . .	11
2.1.1	Geometric Camera Models and Homogeneous Coordinates . . . . .	11
2.1.2	Modeling the Eye Movement . . . . .	14
2.2	Existing Calibration Methods . . . . .	17
2.2.1	The Calibration Method based on a One-Radius Model . . . . .	17
2.2.2	The Calibration Method based on a Two-Radius Model . . . . .	17
2.3	Proposed Method: Projective Geometry Based Calibration . . . . .	19
2.3.1	Estimating Projection Parameters from Known Eye Positions . . . . .	19
2.3.2	Calculating Eye Position During Eye Tracking . . . . .	27

---

2.4	Results . . . . .	29
2.5	Summary . . . . .	33
<b>3</b>	<b>The Simulation System</b>	<b>36</b>
3.1	Background and the Importance of Ground Truth . . . . .	36
3.2	Proposed System: A Novel Simulation System Generating Realistic Eye Images . .	37
3.2.1	Simulation of the Projective Geometry for the 3D Eye Movement . . . . .	37
3.2.2	Photometric Simulation . . . . .	42
3.2.3	Simulation of the Eyelid Movement . . . . .	46
3.3	Results . . . . .	46
3.4	Summary . . . . .	48
<b>III</b>	<b>Eye Movement Tracking</b>	<b>50</b>
<b>4</b>	<b>Horizontal and Vertical Eye Tracking</b>	<b>51</b>
4.1	Existing Online Eye Tracking Systems . . . . .	51
4.2	Proposed System: A High-Speed Online Eye Tracking System . . . . .	53
4.2.1	System Overview . . . . .	54
4.2.2	High-Speed Pupil Detection: The Dual-Mode Capturing Software and the Two-Step Processing Algorithm . . . . .	56
4.2.3	The Symmetric Mass Center Algorithm . . . . .	58
4.3	Results . . . . .	60
4.3.1	Speed of Horizontal and Vertical Position Measurement . . . . .	60
4.3.2	Accuracy of the Symmetric Mass Center Algorithm . . . . .	61

---

4.4	Summary . . . . .	63
<b>5</b>	<b>Torsional Eye Tracking</b>	<b>65</b>
5.1	Existing Torsional Eye Tracking Methods . . . . .	65
5.2	Proposed System: The New Torsional Eye Tracking Algorithm . . . . .	68
5.2.1	Iris Localization Algorithm . . . . .	70
5.2.2	The Robust Template-Matching Algorithm . . . . .	75
5.3	Results . . . . .	79
5.3.1	Different Geometric Compensation Methods . . . . .	79
5.3.2	Robust Template Matching Algorithm . . . . .	83
5.4	Summary . . . . .	87
<b>6</b>	<b>Pupil Monitoring and Application in DR Screening</b>	<b>89</b>
6.1	Motivation: Diabetic Retinopathy Diagnosis . . . . .	89
6.2	Existing Pupil Area Monitoring Methods . . . . .	92
6.3	Proposed System: Fast Online Pupil Monitoring System . . . . .	96
6.3.1	System Overview . . . . .	96
6.3.2	Online Pupil Area Measuring Based on Ellipse Estimation with Shape Constraint . . . . .	100
6.4	Results . . . . .	104
6.5	Application in DR Screening and Proof of Concept Test with Human Subjects . . .	108
6.5.1	Analysis of the Pupil Response . . . . .	109
6.5.2	Proof of Concept Test with Human Subjects . . . . .	112
6.6	Summary . . . . .	115

<b>IV</b>	<b>Conclusions and Future Work</b>	<b>118</b>
<b>7</b>	<b>Summary of Contributions and Future Work</b>	<b>119</b>
<b>V</b>	<b>Appendix</b>	<b>123</b>
<b>A</b>	<b>Homography Between Different Views of the Eye and Relation Between Shapes of Pupil (Iris) Boundaries</b>	<b>124</b>
A.1	Derivation of Homographies . . . . .	125
A.2	Mathematical Relation between the Shapes of Pupil Boundary . . . . .	126

# List of Figures

1.1	A typical diagnostic (screening) procedure using eye tracking result as a measurement	6
2.1	The pinhole camera model.	13
2.2	Projection of the eye onto the image plane and the coordinate systems: the eye coordinate system $(O, X_e Y_e Z_e)$ ( $O$ is chosen to be the horizontal rotation center of the eyeball), the head coordinate system $(O, X_h Y_h Z_h)$ , the camera coordinate system $(O_c, X_c Y_c Z_c)$ , and the image coordinate system $(O_i, X_i Y_i)$ .	15
2.3	Measurement error with different calibration methods. The star data points show the performance of Moore's method, the circle data points show the performance of Zhu's method, and the dot data points show the performance of our proposed method. The left column shows the cases when the eye only moves horizontally, i.e., $\phi = 0^\circ$ . The middle column shows the same plots as the left column at a finer scale in the $Y$ axis. The right column shows the cases when the eye only moves vertically, i.e., $\theta = 0^\circ$ . Note that the right column has the same scale in $Y$ axis with the middle column.	31

2.4 Measurement error with different calibration methods. The star data points show the performance of Moore’s method, the circle data points show the performance of Zhu’s method, and the dot data points show the performance of our proposed method. The left column shows the cases when the eye only moves horizontally, i.e.,  $\phi = 0^\circ$ . The right column shows the cases when the eye only moves vertically, i.e.,  $\theta = 0^\circ$ . Note that the right column has different scale in  $Y$  axis with the first column. . . . . 32

3.1 (a) Diagram of the eye structure, (b) A picture of the eye. . . . . 39

3.2 Phone’s model used to model the specular component of the cornea surface reflection. Specular surfaces usually reflect light into a lobe of directions around the specular direction, and the reflection intensity depends on the direction. . . . . 44

3.3 The eye model used to calculate the position of reflections in the image.  $R_{eye}, R_c, R_i$  are radius of the eye ball, the cornea sphere, and the iris plane. The specular direction and the light source direction are symmetric with respect to the surface normal. For each different eye position, the light source position and the camera position coordinates need to be recalculated (transformed into the eye coordinate system). . . . . 45

3.4 Deformation of the upper eyelid. The starting curve of the upper eyelid area and the original upper eyelid is hand-marked by the user.  $E_1, E_2$  are the beginning and ending columns of the eyelid deformation area.  $d$  is the amount of eyelid movement. A column line intersects the starting curve, the original upper eyelid and the new upper eyelid at  $S(x), M_o(x), M_n(x)$  lines, respectively. . . . . 47

3.5	Preprocessing of template images. The upper left image shows the template image captured. The upper right image shows the image with the cornea reflection filled and the pupil center marked. The lower left image shows the marked eyelid boundary. The boundary of the upper eyelid area is marked in blue and the boundary of the lower eyelid area is marked in red. The lower right image shows different regions segmented. . . . .	47
3.6	Images with different eyelid movement while the eye is in the same position . . . .	48
3.7	Images with different eye movement . . . . .	48
4.1	(a) The system components.  (b) A subject wearing the goggle with two cameras installed. The numbers in the picture indicate: (1) Two cameras installed on the goggle, with infrared pass filters installed before the camera lens. (2) Hot mirrors reflecting eye images to cameras. (3) 1394 Fireware cable connecting the cameras to the desktop computer system. . .	55
4.2	The upper row shows a full pixel density resolution grey-level image (the partial image defined by the ROI window), with the small threshold image inlayed on the right location. The lower row shows, from left to right, the downsampled grey-level image, the threshold image of the downsampled image, the grey-level image of full pixel density resolution in the tiny trace window, and the small threshold image in the tiny trace window. . . . .	57
4.3	Locating the center for an elliptical pupil area that is partially occluded. The algorithm finds the maximum parallelogram $S_1E_1E_3S_3$ , and calculates the mass center of the area enclosed by $\widehat{S_1}\widehat{S_3}\widehat{E_3}\widehat{E_1}$ . . . . .	58

4.4	Processing rate with different ROI size. By implementing the two-step processing algorithm, the processing rate is about 2-3 times faster than locating the pupil center directly at full pixel density resolution. . . . .	60
4.5	Measurement results for different algorithms when the pupil area is partially occluded:  (a) shows the eye images and highlights un-occluded part of the pupil boundary in red;  (b) plots the measurement results for different algorithms in terms of percentage of the pupil boundary being occluded. . . . .	63
5.1	Intermediate results for iris localization . . . . .	71
5.2	(a) Boundary of iris area localized in pink color;  (b) The iris template position in red color;  (c) Top image: the unwrapped iris template, bottom image: the mask identifying real iris pixels. . . . .	74
5.3	(a) The reference and test template captured for Zhu’s template matching algorithm.  (b) The reference and test template captured for our proposed robust template matching algorithm. . . . .	77
5.4	Cost function for different hypothetical torsion movement. . . . .	78
5.5	Distance function of each segment for different hypothetical torsion movement. . .	79
5.6	Torsion testing results using different geometric compensation method . . . . .	80
5.7	The reference image and the edge points detected on the pupil and iris boundary . .	81

5.8	Mapping of edge points on the pupil and iris boundary in the reference image to the test image by results from three different calibration algorithms. . . . .	82
5.9	Results when the algorithms employ a full searching range in $[-20^\circ, 20^\circ]$ . . . . .	84
5.10	Results when the algorithms employ a fixed inter-frame searching range with the last frame torsion position known . . . . .	85
5.11	Results when the algorithms employ a fixed or adaptive inter-frame searching range	86
6.1	A scene as it might be viewed by a person with normal vision (a) and with DR (b). Both pictures are from <a href="http://www.nei.nih.gov/photo/sims/">http://www.nei.nih.gov/photo/sims/</a> . . . . .	89
6.2	System components and the block diagram. . . . .	97
6.3	The LCD screen that presents the visual stimulus and the eye image capturing device, including the goggle frame, the cameras connected to the desktop computer by the Firewire cables, the LEDs used for illumination, the hot mirrors reflecting the images to the cameras, and the lens holders that can hold two thin rim trial lenses. . . . .	98
6.4	System components and the software diagram . . . . .	98
6.5	Examples of stimulus pattern, the circular object for the central vision and the annular object for the peripheral vision. . . . .	99
6.6	Curvature calculation in point c in an edge segment. It is based on the average gradient angle of point b and f, and curve length between them. . . . .	102

6.7 Flow of the algorithm and intermediate results:

- (a) The input grey-level image.
- (b) The result of threshold and blob analysis.
- (c) Output of the Canny edge detector.
- (d) The result of curve segmentation based on discontinuity of curvature.
- (e) The result of the iterative ellipse fitting process.
- (f) The final ellipse estimated shown in the input image. . . . . 104

6.8 Three iterations in the iterative ellipse fitting procedure. Red segments are those already chosen from previous iterations; the yellow segment is the one currently being processed. In the title of the left column, the new cost when the current segment is added is shown, and the decision whether to accept the segment or to discard is displayed as well. In the right column, the blue curve shows the ellipse estimated from currently chosen segments. . . . . 105

6.9 Eye images and the estimated elliptical pupil boundary. . . . . 106

6.10 The average measurement error and the 95% confidence interval for each bin of images with the occlusion level in  $(x - 5\%, x + 5\%]$ . . . . . 108

6.11 Tools to remove blinks. The black plots are the eye response traces, and the red dots represent the blinks detected. . . . . 110

6.12 In the pupil size trace plot, the solid red plot is the chosen part of the eye response trace for analysis, the dashed red part is the blink removed previously, and the yellow plot represents the on/off time of the central stimulus. . . . . 111

6.13 The average response for subject D0029, and the fitted second-order polynomial plots and equations. . . . . 116

6.14 Feature parameters for each individual subject who participated in the test. Light blue: normal subjects; Green: diabetic patients with no DR; Yellow: diabetic patients with mild DR; Purple: diabetic patients with moderate DR; Red: diabetic patients with severe DR. . . . . 117

# List of Tables

2.1	Intrinsic parameters of a camera . . . . .	13
2.2	Eye position used for calibration . . . . .	17
2.3	Measurement error when the camera rotation offset is $[10^\circ, 10^\circ, 10^\circ]$ , and $\alpha = 0.2$ . . . . .	34
4.1	System configuration . . . . .	54
4.2	Dimensions of the ROI window and the corresponding normalized image size . . . . .	61
4.3	Overall performance of different algorithms . . . . .	62
5.1	Horizontal and vertical eye position for testing images. . . . .	80
5.2	Average measurement error, 95% confidence interval and number of outliers for different geometric compensation methods . . . . .	81
5.3	Overall performance of different algorithms . . . . .	87
6.1	Ellipse parameters . . . . .	104
6.2	The 3D eye positions for which images are synthesized using the simulation system. . . . .	107
6.3	Distribution of the subjects participated in the test and included in the final data analysis . . . . .	114

6.4 Color of the data bar and its represented prior knowledge of the subject with regard  
to clinical retinopathy. . . . . 114

# **Part I**

## **Introduction**

# Chapter 1

## Background and Motivation

### 1.1 Eye Dynamics, Related Medical Applications and Eye Tracking Systems

Eye dynamics, including 3D spatial movement (horizontal, vertical and torsional), pupil dilation and extraction have been used by clinicians to diagnose or screen a number of diseases including balance disorder, diabetic retinopathy (DR), strabismus, cerebral palsy, multiple sclerosis, etc. [1–3]. Eye movement is also investigated by researchers to study human ophthalmological, vestibular, and neuro-otologic systems [4, 5]. The essential component of a diagnostic or screening system is a reliable eye movement tracking system.

It is well known that abnormal eye movement can provide an indication of neurological, ophthalmological, and other medical problems. Among these, vestibular (balance) related problems are the most widely investigated in both research and clinical studies. Abnormal eye movement can also indicate diseases such as multiple sclerosis, Parkinson’s disease, diabetic retinopathy, etc.

Balance is a combined function of three systems: the vestibular system, vision, and the central neurological system. The vestibular system accomplishes two tasks. First, it contributes to an individual's sense of motion and spatial orientation with respect to the force of gravity. There are three perpendicular semicircular canals in the inner ear. Each canal has hair cells that can detect changes in fluid displacement and send information to the central neural system about acceleration and head tilt. Second, when head and body are in motion, the vestibular system controls eye movements so that the eye can capture steady and in-focus images. This is the vestibular-ocular reflex (VOR). The vestibule has a direct influence on eye motion via the VOR. By measuring the presence, absence, and degree of eye movement provoked by various stimuli of body motion, eye tracking systems can indirectly measure the vestibular function. Tracking and analyzing the eye movement can provide valuable information on the diagnosis of problems related to the vestibular system [6].

The positional eye movement has six degrees of freedom: three translations in the socket and three rotations. The translations are usually small and are generally ignored. The three rotations are the horizontal, vertical, and torsional eye movement. Torsional movement refers to the rotation of the eye around the axis along the line of sight.

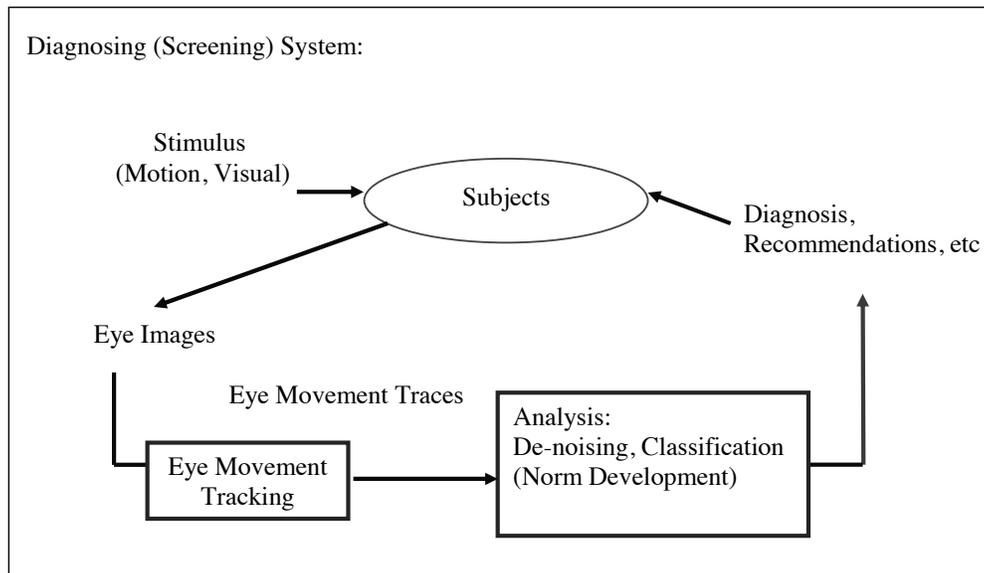
The eye is never completely at rest, even when it is fixed on a target. Fixation eye movement refers to this eye "fluctuation" when it tries to stabilize the retina over a stationary object of interest; it is involuntary and related to brain stem activity. Saccade is the rapid jerky eye movement when the eye tries to focus on a new object. The speed of eye movement can approach  $700^\circ/sec$  for large saccades. In most cases, the eye reaches the new position quickly and without noticeable oscillation or correction. Undershooting the target with jumps larger than  $20^\circ$  degree is common and overshooting is rare. There is also a latency of about 200 milliseconds between each target jump

and the induced saccade [7]. Abnormal saccades, which can be identified by traits including the velocity, accuracy, and latency of eye movement, can be used to assess neurological pathogenicities. Nystagmus characterizes the involuntary rhythmic eye movement when the eye moves quickly in one direction (fast phase) and then slowly in the other (slow phase). Nystagmus can be horizontal, vertical, or torsional. Most eye disorders associated with nystagmus can easily be diagnosed by tracking the visual pattern.

Pupil constriction and dilation is another type of dynamics that has been used in the medical field for diagnostic purposes. For people who might have a disease on the retina of the eye, recording and analyzing the pupil response under designated light stimulus can provide precious information on the retina function, which may not be easily available by other conventional examination methods. Monitoring pupil response can help to diagnose several diseases such as glaucoma, or diabetic retinopathy [3].

An eye tracking system typically uses electrical and computing devices to track the eye movement. It can provide a quantitative, consistent, and repeatable record of eye movements. It helps distinguish small deviations such as high frequency, low amplitude movements, and helps in early detection of diseases, or diagnosis of low level, chronic diseases.

Figure 1.1 shows the diagram of a typical diagnostic (screening) process where eye tracking result is used as a measurement. Subjects are given a certain stimulus, usually motion (in vestibular tests) or visual (in diabetic retinopathy tests) stimulus in the test. Concurrently, the eye images are captured by video cameras and processed by eye tracking software. As the output of the eye tracking software, eye movement traces are given and saved for analysis. The analysis software usually performs de-noising first to obtain a cleaner signal. A norm indicating either normal or different risk of sickness is developed based on tests among the group of population of interest.



**Figure 1.1:** A typical diagnostic (screening) procedure using eye tracking result as a measurement

Computationally speaking, this is a classification problem. The output of the analysis software usually would be an index associated with the norm. Based on the analysis results, a physician (M.D.) will make a diagnosis or give recommendations to the patient. If necessary, another test may be arranged for further investigation.

The research in this PhD dissertation concentrates on eye tracking systems, because an accurate and robust measurement of eye movement is essential for many applications. The analysis and norm development is of great interest and would be an important direction for future research.

## 1.2 Related Work and the Need for a New System

To capture rapid eye movement, a high temporal resolution is sometimes required. Online processing is essential because it may be important for the experimenter (e.g., the physician) to have immediate control over the output quality [8]. In addition, the experimenter monitoring the test may decide to do further tests on-the-fly based on the observed test results.

The digital video based eye tracking system is non-invasive compared to other methods including scleral search coil systems [6, 9, 10]. Head-mounted eye tracking systems are more accurate than non-head-mounted systems, electro-oculography (EOG) systems [11], cornea reflection systems [12, 13], or flying-spot laser based systems [14, 15]. The temporal resolution of the video based eye tracking systems used to be limited by the camera speed and computation power. With the availability of ever increasing computation power, the development of digital camera and image processing technology, it is now possible to exploit the full potential of video based eye tracking systems. In this research, we investigate digital image based eye tracking systems and study three aspects: tracking the horizontal and vertical movement, monitoring the pupil size change, and tracking the torsional eye movement.

Current video image based eye tracking systems either run online at a low speed, do the processing off-line, or use dedicated hardware to reach high online processing rates; they are usually not robust to interference and scenarios including eyelid drops, illumination variations, eccentric eye locations, etc. In this PhD dissertation, we report an accurate, robust, and online eye movement tracking system by applying image processing techniques on captured digital video images.

To capture dynamics during high-speed eye movement, e.g., during saccadic eye movement, a high frame rate is needed, and often an online system is preferred. Therefore, the system required is one that can capture and process images at high frame rate, and at the same time, still be able to handle interference arising from eyelid drops, reflections, and blinks, etc. Considering the development cost and time, a tracking system running on a general computer system would be preferred. Therefore, we developed an accurate, robust, and high-speed horizontal and vertical eye tracking system on a general computer system by using commercially available digital cameras.

Torsional eye position measurement is very important for the diagnosis of balance related prob-

lems and research in the vestibular system. Most systems available in the market and reported in the literature are not robust enough for clinical use, especially in situations when the eye is not in central position or the eye is partially occluded by eyelids. Therefore, we developed a torsional eye tracking system robust to artifacts and various testing scenarios.

DR is one of the potential complications of diabetes that may cause blindness. DR can cause vascular damage on the retina before the patient develops more severe symptoms. Patients with vascular damage may have a different pupil contraction and dilation pattern under some light stimulus. It is possible to develop a screening method if we can measure the pupil size change under designated light stimulus. Therefore, in this thesis we have also developed an accurate and online pupil size monitoring system for the screening of DR.

Eye position in space can be described by its horizontal, vertical, and torsional components. Discovering the 3D eye position from 2D images requires the recovering of image projection parameters through calibration. Existing calibration methods either use a simplified model for eye movement or a simplified model for the imaging process to achieve a closed-form or linear solution to the problem. These simplifications can result in large error in eye position measurement. In this PhD dissertation, we report a novel calibration method that can accurately measure the 3D eye position in a wider range than other systems. The method uses a more accurate two-radius rotation model for eye movement and the perspective projection model for the imaging process. The parameters of the elliptical pupil boundary from multiple views of the eye are used as the constraints to estimate the projection parameters.

Given the importance of eye tracking systems, it is still difficult to evaluate and compare the performance of these systems in terms of their accuracy, robustness, and capability to handle artifacts including occlusion and changes in illumination. The difficulty lies in the lack of ground truth

information. To provide data sets with ground truth available, we developed a simulation system that generates realistic eye video images with known eye position.

## **Part II**

# **Calibration Method and Simulation System for Eye Tracking**

# Chapter 2

## The Calibration Method

Eye position in space can be described by its horizontal, vertical, and torsional components. However, video images only provide  $2D$  information of the eye position. Camera projection parameters are needed to fully recover the  $3D$  position of the eye from images. Calibration is the procedure for finding the camera projection parameters, and is especially important to recover the torsional component, in which geometrical distortion of the iris pattern needs to be compensated before calculating the torsional component. In this chapter, we present a novel calibration method that can accurately measure the  $3D$  eye position in a wider range than other systems. We start with the basics of camera models and existing eye tracking calibration methods.

### 2.1 Background and Motivation

#### 2.1.1 Geometric Camera Models and Homogeneous Coordinates

The imaging procedure is typically modeled by the pinhole model as shown in Figure 2.1. A point  $P$  with the coordinate vector  $[x, y, z]^T$  in the camera based coordinate system  $(O, ijk)$  is projected

onto the image plane  $\Pi$  at point  $[x', y']$  in the image coordinate system, and we have:

$$\begin{cases} x' = f \frac{x}{z} \\ y' = f \frac{y}{z} \end{cases}, \quad (2.1)$$

where  $f$  is the distance from the camera center (the pinhole) to the image plane. One can see that the position of the projected point on the image is related to  $z$ , the distance of the point in the 3D space to the camera plane  $(O, ij)$ . This model is called the perspective projection model. Let  $m = -\frac{f}{z}$ ; when the scene depth (variation of  $z$ ) is small relative to the distance to the camera,  $m$  can be taken as a constant. This model is called the weak perspective projection model. We have:

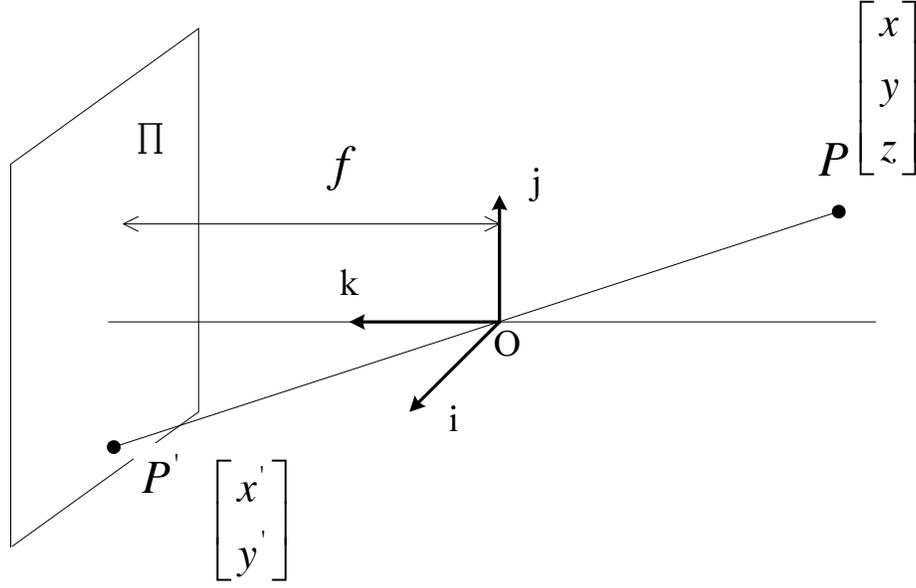
$$\begin{cases} x' = -mx \\ y' = -my \end{cases}. \quad (2.2)$$

Furthermore, when we normalize the image coordinates so that  $m = -1$ , this is the orthographic projection model, and we have:

$$\begin{cases} x' = x \\ y' = y \end{cases}. \quad (2.3)$$

However, (2.1)-(2.3) are only valid when all distances and coordinates are measured in the camera's coordinate system, and the origin of the image coordinate system is at the principal point, where the camera's optical axis (-the  $k$  axis in Figure 2.1) intersects with the image plane. In reality, the image origin may not be at the principal point, and the object may be measured in a world coordinate system, which is different from the camera coordinate system. Typically the world frame and the camera frame may be related by rotation, translation, or difference on scales, etc.

Table 2.1 lists the intrinsic parameters of cameras, which relate a real camera coordinate system to the idealized one described in Figure 2.1 and (2.1). It is known that a point on the image with



**Figure 2.1:** The pinhole camera model.

$\alpha$	magnification in the $x$ direction: it is related with $f$ and pixel size
$\beta$	magnification in the $y$ direction: it is related with $f$ and pixel size
$\theta$	skew: angle between the two axes of the image, sometimes it is not strictly $90^\circ$
$u_0, v_0$	principal point: the intersection point of the camera optical axis and the image plane

**Table 2.1:** Intrinsic parameters of a camera

coordinate vector  $[u, v]^T$  is related with the object point with camera coordinate vector  $[x^c, y^c, z^c]^T$  according to the following equation [16]:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^c} \mathcal{K} \begin{bmatrix} x^c \\ y^c \\ z^c \end{bmatrix}, \quad \text{where } \mathcal{K} = \begin{bmatrix} \alpha & -\alpha \cot \theta & u_0 \\ 0 & \frac{\beta}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

We define the *homogeneous coordinate vector* of a 3D point  $[x, y, z]^T$  as  $\mathbf{P} = [x, y, z, 1]^T$ , and the homogeneous coordinate vector of a 2D point on an image as  $\mathbf{p} = [u, v, 1]^T$ . One benefit of homogeneous coordinates is that we can describe rotation and translation as one linear transforma-

tion. For example, if there is a translation and rotation between the world coordinate system and the camera coordinate system, one has:

$$\mathbf{P}^c \simeq \begin{bmatrix} \mathcal{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{P}^w, \quad \text{where } \mathbf{P}^c = \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix}, \mathbf{P}^w = \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix} \quad (2.5)$$

In (2.5),  $\mathcal{R}$  is the  $3 \times 3$  rotation matrix, and  $\mathbf{t}$  is the  $3 \times 1$  translation vector; they are usually referred as the extrinsic parameters of the camera, which relate the camera coordinate system with the world coordinate system.

The notation  $\simeq$  in (2.5) represents *projective equality*. When one has  $A \simeq B$ , it means that there exists a scalar value  $c$ , such that  $A = cB$ . One intuition is that if two homogeneous coordinates are projective equal to each other, they represent the same physical point. With projective equality, (2.4) and (2.5) can be rewritten as:

$$\mathbf{p} \simeq \mathcal{K} \begin{bmatrix} \mathcal{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{P}^w, \quad \text{in which } \mathbf{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \quad (2.6)$$

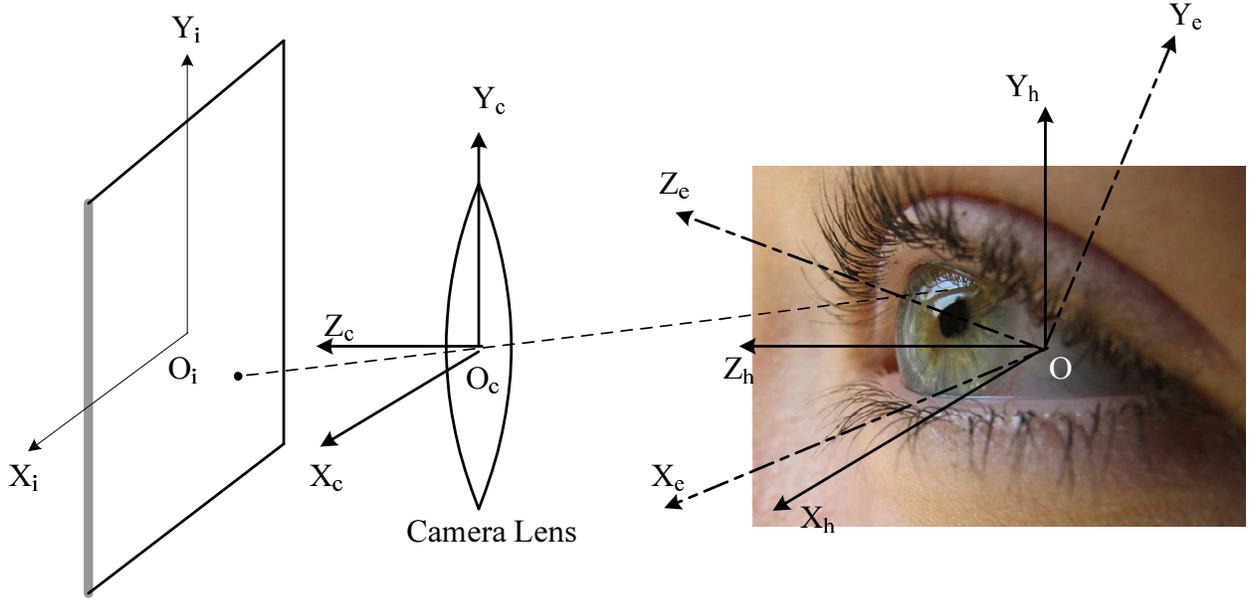
### 2.1.2 Modeling the Eye Movement

In eye movement tracking, the eye is typically modeled as a perfect sphere and a solid object [17–20], i.e., there is no deformation of the eye ball. It is also commonly assumed that:

- The iris and pupil are approximated as a plane intersecting the sphere;
- The eye movement consists mainly of pure rotation around the sphere center, with no or very small translation components;

- The visual axis of the eye can be approximated by the optical axis .

In some scenarios [18], it is assumed that eye rotates around the same center, while in other cases, it is modeled such that the vertical rotation center is distinct from the horizontal rotation center [19–21].



**Figure 2.2:** Projection of the eye onto the image plane and the coordinate systems: the eye coordinate system  $(O, X_e Y_e Z_e)$  ( $O$  is chosen to be the horizontal rotation center of the eyeball), the head coordinate system  $(O, X_h Y_h Z_h)$ , the camera coordinate system  $(O_c, X_c Y_c Z_c)$ , and the image coordinate system  $(O_i, X_i Y_i)$ .

Figure 2.2 shows the relative position of the eye and the camera in the system, and four coordinate systems of interest: the eye coordinate system, the head coordinate system, the camera coordinate system, and the image coordinate system. For a camera installed on a wear-on goggle, it can be assumed that there is no relative motion between the head and the camera during tests. The offset between the head coordinate system and the camera coordinate system is characterized by a rotation  $\hat{R}(\theta_c, \phi_c, \psi_c)$  and a translation  $\hat{s}$ . Therefore, the homogeneous coordinate vector  $P^h = [x^h, y^h, z^h, 1]^T$  in the head frame, the corresponding homogeneous coordinate vector in the

camera frame  $P^c = [x^c, y^c, z^c, 1]^T$ , and the homogeneous coordinate vector in the image frame  $\mathbf{p}^i = [u, v, 1]^T$  are related by:

$$\begin{aligned} \mathbf{P}^c &\simeq [\hat{R}(\theta_c) \hat{R}(\phi_c) \hat{R}(\psi_c), \hat{s}] \mathbf{P}^h \\ \mathbf{p}^i &\simeq \mathcal{K} \mathbf{P}^c \end{aligned} \tag{2.7}$$

Matrix  $\mathcal{K}$  represents the intrinsic camera parameters described in (2.4).  $\hat{R}(\theta_c), \hat{R}(\phi_c), \hat{R}(\psi_c)$  represent the horizontal, vertical and torsional rotation between the head frame and the camera frame, respectively. Note that  $\mathcal{K}, \hat{R}(\theta_c), \hat{R}(\phi_c), \hat{R}(\psi_c)$ , and  $\hat{s}$  are fixed if there is no change to the camera and there is no relative motion between the head and the camera installed on the wear-on goggle, i.e., these values do not change with different eye positions.

The motion of the eyeball relative to the head is *the movement* that we are trying to recover from captured images. Assuming that for a point on the eye, e.g., the pupil center, the homogeneous coordinate vector in the eye frame is  $\mathbf{P}^e = [x^e, y^e, z^e, 1]^T$ , then its corresponding vector in the head frame  $\mathbf{P}^h$  is given by:

$$\mathbf{P}^h = [R(\theta, \phi, \psi), s] \mathbf{P}^e, \tag{2.8}$$

in which  $R(\theta, \phi, \psi), s$  represent the rotation and translation of the eyeball respectively. The translation component is usually very small and negligible in tests [18]. However, as we will see in the following section, distinct rotation centers for vertical and horizontal eye movement can effectively produce a nonzero  $s$  vector, i.e., a small translation component.

If the distance from the iris plane to the horizontal rotation center is  $r_p$ , the homogeneous coordinate vector of the pupil center in the eye frame is always  $[0, 0, r_p, 1]^T$  regardless of the eye position. However, the coordinate vector of the pupil center in the head frame varies with different eye positions, and so is its coordinate vector in the camera frame and its projection on the image.

## 2.2 Existing Calibration Methods

### 2.2.1 The Calibration Method based on a One-Radius Model

A calibration model was first proposed in [18] to compensate for geometric distortion in calculating torsional movement. In addition to the assumptions mentioned in Section 2.1.2, the paper assumes that the horizontal and vertical movement has the same rotation center, and there is no translation component. In addition, orthographic projection is assumed in the model, although in the analysis, they do point out that the error can reach around  $0.5^\circ$  for large eye angles. Camera intrinsic matrix  $\mathcal{K}$  is assumed to be identity matrix as well.

These assumptions simplify the projection procedure and an analytical solution can be derived given five known eye positions and the pupil center in the corresponding images. Table 2.2 shows the five known eye positions used. Typically, one can choose  $\theta = 10^\circ$ ,  $\phi = 10^\circ$ . However, these assumptions can result in measurements with large error when the eye is in eccentric positions.

torsional position	0	0	0	0	0
horizontal position	0	$\theta$	$-\theta$	0	0
vertical position	0	0	0	$\phi$	$-\phi$

**Table 2.2:** Eye position used for calibration

### 2.2.2 The Calibration Method based on a Two-Radius Model

It has been shown, however, that the kinematics of the eye cannot be modeled by pure rotations around the same rotation center [21]; a better approximation of the eye movement can be achieved by shifting the vertical rotation center along the torsional rotation axis ( $Z_e$ )[19, 20]. Experimen-

tally, it is shown in [19] that by using the one-radius model, horizontal eye movements, in absolute values, are uniformly larger than the actual eye movements, and vertical movements uniformly smaller; the resulting error in eye position measurement can be up to  $1.5^\circ$ , while medical specialists using the eye tracking devices demand an accuracy at  $0.1^\circ$ . Therefore, a two-radius model is used in [19, 20] in which the vertical rotation center ( $C_v$ ) is distinct from the horizontal rotation center ( $C_h$ ); there is a shift between them along the optical axis ( $Z_e$ ) of the eye ball. The transformation from the eye frame to the head frame can then be written as:

$$[R(\theta, \phi, \psi), s] = R(\theta)R_tR(\phi)R_{-t}R(\psi), \quad (2.9)$$

in which  $R(\theta)$ ,  $R(\phi)$  and  $R(\psi)$  describe the horizontal, vertical, and torsional rotation (in Fick sequence), respectively, while  $R_t, R_{-t}$  characterize a forth and back translation along the  $Z_e$  axis before and after the vertical rotation correspondingly. The combination of these three transformations  $R_tR(\phi)R_{-t}$  is equivalent to the vertical rotation around the vertical center ( $C_v$ ).

By using the two-radius model, it is shown in [19, 20] that the error in horizontal and vertical eye position measurement can be decreased to  $0.1 - 0.2^\circ$  in certain scenarios. However, in both papers it is assumed that the horizontal and vertical rotational offset angles ( $\hat{R}(\phi_c), \hat{R}(\theta_c)$ ) between the head frame and the camera frame are negligible. Although it is shown in [18] that the error caused by the horizontal and vertical offsets is small compared with the torsional offset  $\hat{R}(\psi_c)$ , the error can still be up to  $1^\circ$ . Furthermore, in [19], orthographic projection model is again assumed and it can produce an error up to  $0.5^\circ$ . In [20], although the perspective projection is used, it is assumed that certain parameters, e.g., the distance from the lens to the CCD chip and the distance from the lens to the center of the pupil, are known, for which it is not easy to obtain an accurate measurement.

In summary, while there exist several calibration methods for eye position measurement, each of them has significant drawbacks. We therefore developed a calibration method to address these issues and to achieve a more accurate measurement for eccentric eye positions.

## 2.3 Proposed Method: Projective Geometry Based Calibration

The new algorithm we propose here is based on the two-radius model introduced in Section 2.2.2. It is shown in [18] that the error caused by ignoring the horizontal and vertical offset between the head and the camera is smaller than ignoring the torsional offset, when the magnitude of the offset is smaller than  $5^\circ$ . First, as mentioned previously, even with this small rotational offset, the resulting error can still be up to  $1^\circ$ . Secondly, in reality, it is difficult, even for trained operators, to adjust the wear-on goggle so that the offset is guaranteed to be small and negligible. Therefore, in our proposed calibration method, the horizontal and vertical rotation offset is taken into account, and a more accurate perspective projection model is used.

### 2.3.1 Estimating Projection Parameters from Known Eye Positions

#### Solution assuming translation parameter $\alpha$ is known

By assuming the two-radius model, from (2.8), (2.9) one gets:

$$\mathbf{P}^h = R(\theta) R_t R(\phi) R_{-t} R(\psi) \mathbf{P}^e, \quad (2.10)$$

in which,  $\mathbf{P}^h = [x^h, y^h, z^h, 1]^T$ ,  $\mathbf{P}^e = [x^e, y^e, z^e, 1]^T$ , and for pupil center,

$$\mathbf{P}_{pc}^e = [0, 0, r_p, 1]^T \quad (2.11)$$

The three rotation matrices and two translation matrices are:

$$\begin{aligned}
 R(\theta) &= \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 R(\phi) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 R(\psi) &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 & 0 \\ \sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
 \end{aligned} \tag{2.12}$$

$$R_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_{-t} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.13}$$

in which  $t$  represents the amount of translation between the horizontal and the vertical rotation center. Substituting (2.11), (2.12), and (2.13) into (2.10), one obtains the pupil center in the head

frame:

$$\mathbf{P}_{pc}^h = \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi & -t \sin \theta \cos \phi + t \sin \theta \\ 0 & \cos \phi & -\sin \phi & t \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & -t \cos \theta \cos \phi + t \cos \theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ r_p \\ 1 \end{bmatrix} \quad (2.14)$$

In (2.7), let

$$M = \mathcal{K} [\hat{R}(\theta_c) \hat{R}(\phi_c) \hat{R}(\psi_c), \hat{s}], \quad \text{and}$$

$$M = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \\ m_9 & m_{10} & m_{11} & m_{12} \end{bmatrix}. \quad (2.15)$$

By substituting (2.14) and (2.15) into (2.7), the pupil center in the image coordinate system  $\mathbf{p}_{pc}^i$

can be written as:

$$\mathbf{p}_{pc}^i \simeq M \begin{bmatrix} r_p \\ \begin{bmatrix} \sin \theta \cos \phi \\ -\sin \phi \\ \cos \theta \cos \phi \\ 0 \end{bmatrix} \end{bmatrix} + t \begin{bmatrix} \begin{bmatrix} \sin \theta(1 - \cos \phi) \\ \sin \phi \\ \cos \theta(1 - \cos \phi) \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \quad (2.16)$$

$$\text{and } \mathbf{p}_{pc}^i = [u, v, 1]^T$$

Let  $t = \alpha \cdot r_p$ , and  $[x^i, y^i, z^i]^T = \text{right hand side of (2.16)}$ . By substituting (2.15) into (2.16), one

gets:

$$\begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix} = \begin{bmatrix} r_p m_1 & r_p m_2 & r_p m_3 & m_4 \\ r_p m_5 & r_p m_6 & r_p m_7 & m_8 \\ r_p m_9 & r_p m_{10} & r_p m_{11} & m_{12} \end{bmatrix} \begin{bmatrix} \sin \theta \cos \phi + \alpha \sin \theta(1 - \cos \phi) \\ (\alpha - 1) \sin(\phi) \\ \cos \theta \cos \phi + \alpha \cos \theta(1 - \cos \phi) \\ 1 \end{bmatrix} \quad (2.17)$$

Let

$$M_r = \begin{bmatrix} r_p m_1 & r_p m_2 & r_p m_3 & m_4 \\ r_p m_5 & r_p m_6 & r_p m_7 & m_8 \\ r_p m_9 & r_p m_{10} & r_p m_{11} & m_{12} \end{bmatrix} \quad (2.18)$$

One can observe that there are 12 unknown parameters in the matrix  $M_r$  if we combine each  $r_p m_k (k = 1, \dots, 11)$  as one parameter. For each image acquired in a known eye position, i.e. , known  $\theta$  and  $\phi$ , and with detected pupil center  $(u, v)$  in this image,

$$[u, v, 1]^T \simeq [x^i, y^i, z^i]^T \quad (2.19)$$

Therefore, the following two equations hold:

$$u = \frac{x^i}{z^i}, \quad v = \frac{y^i}{z^i}.$$

One therefore obtains:

$$\begin{aligned} u \vec{f}_\alpha^t \vec{m}_{r3} - \vec{f}_\alpha^t \vec{m}_{r1} &= 0 \\ v \vec{f}_\alpha^t \vec{m}_{r3} - \vec{f}_\alpha^t \vec{m}_{r2} &= 0 \end{aligned}, \quad (2.20)$$

in which,

$$\begin{aligned} \vec{m}_{r1} &= [r_p m_1 \quad r_p m_2 \quad r_p m_3 \quad m_4]^t \\ \vec{m}_{r2} &= [r_p m_5 \quad r_p m_6 \quad r_p m_7 \quad m_8]^t \\ \vec{m}_{r3} &= [r_p m_9 \quad r_p m_{10} \quad r_p m_{11} \quad m_{12}]^t, \end{aligned}$$

and

$$\vec{f}_\alpha = \begin{bmatrix} \sin \theta \cos \phi + \alpha \sin \theta (1 - \cos \phi) \\ (\alpha - 1) \sin(\phi) \\ \cos \theta \cos \phi + \alpha \cos \theta (1 - \cos \phi) \\ 1 \end{bmatrix}$$

If the system acquires and processes multiple such images with known eye positions, there will be multiple equations similar to the ones in (2.20). By stacking them together, one obtains an array of equations in matrix form:

$$\begin{bmatrix} -\overrightarrow{f_\alpha(1)}^t & \mathbf{0} & u_1 \overrightarrow{f_\alpha(1)}^t \\ \mathbf{0} & -\overrightarrow{f_\alpha(1)}^t & v_1 \overrightarrow{f_\alpha(1)}^t \\ \vdots & \vdots & \vdots \\ -\overrightarrow{f_\alpha(n)}^t & \mathbf{0} & u_n \overrightarrow{f_\alpha(n)}^t \\ \mathbf{0} & -\overrightarrow{f_\alpha(n)}^t & v_n \overrightarrow{f_\alpha(n)}^t \end{bmatrix} \begin{bmatrix} \overrightarrow{m_{r1}} \\ \overrightarrow{m_{r2}} \\ \overrightarrow{m_{r3}} \end{bmatrix} = \mathbf{0}. \quad (2.21)$$

Let

$$Q = \begin{bmatrix} \overrightarrow{m_{r1}} \\ \overrightarrow{m_{r2}} \\ \overrightarrow{m_{r3}} \end{bmatrix}, \quad \text{and} \quad L = \begin{bmatrix} -\overrightarrow{f_\alpha(1)}^t & \mathbf{0} & u_1 \overrightarrow{f_\alpha(1)}^t \\ \mathbf{0} & -\overrightarrow{f_\alpha(1)}^t & v_1 \overrightarrow{f_\alpha(1)}^t \\ \vdots & \vdots & \vdots \\ -\overrightarrow{f_\alpha(n)}^t & \mathbf{0} & u_n \overrightarrow{f_\alpha(n)}^t \\ \mathbf{0} & -\overrightarrow{f_\alpha(n)}^t & v_n \overrightarrow{f_\alpha(n)}^t \end{bmatrix},$$

in which  $Q$  is a vector with 12 unknown parameters. Note that  $Q$ , and  $M_r$ , are defined up to a scaling factor, which means if  $Q$  ( $M_r$ ) is a solution, then  $c \cdot Q$  ( $c \cdot M_r$ ) is also a solution for any constant  $c$ .

If  $\alpha$  is known, one can solve (2.21) for  $Q$  if there are eleven equations, i.e., six images with known eye position. With more than six such images, (2.21) can be solved in the mean square sense, which means by solving for the  $Q$  with unit norm that minimizes  $\|LQ\|$ , i.e.,  $Q^T L^T L Q$ . This is a typical homogeneous linear least-squares problem, and the minimum value of  $Q^T L^T L Q$  is reached at  $Q =$  eigenvector of  $L^T L$  corresponding to its smallest eigenvalue. For the robustness

of the algorithm, nine known eye positions shown below are used in real tests:

horizontal angle (in degrees)( $\theta$ )	-30	-15	0	15	30	0	0	0	0
vertical angle (in degrees)( $\phi$ )	-0	-0	0	0	0	-20	-10	10	20

### Dealing with unknown $\alpha$

However, for a particular set of tests with one subject, we do not know the  $\alpha$  value. There does not exist a linear solution to the problem, i.e., the minimization of  $\|\mathbb{L}\mathbf{Q}\|$ , since there are *non-linear items* included in the equations if we consider  $\alpha$  an unknown parameter. However, since we have a prior knowledge that  $t$ , the translation value, is smaller than  $r_p$ , the horizontal rotation radius, it follows that the following inequality holds:

$$0 \leq \alpha \leq 1.$$

We can do a linear search between  $[0, 1]$  to find a numerical solution close to the optimal value of  $\alpha$ . Specifically, the algorithm searches for  $\alpha$  at an incremental value of 0.1 between  $[0, 1]$ , and then at an incremental value of 0.01 between  $[k - 0.1, k + 0.1]$ , in which  $k$  is the best value obtained in the previous step.

### Nonlinear optimization

The above linear solution can only find a close-to-optimal solution to minimize the algebraic distance defined by  $Q^T L^T L Q$ . In addition, the above algorithm is built upon the elliptical pupil center; actually when the eye is not in the central position, the center of the ellipse defined by the moved eye pupil boundary is not exactly the projective transformation of the circular pupil center when the eye is in the central position. Therefore, we use the shape of the elliptical pupil boundary for further optimization. It can be proven (see Section A.2 for the proof) that the shape of the elliptical

pupil boundary in different eye position is related to the shape of pupil when the eye is at the center according to (2.22):

$$C_i \simeq H_{0 \leftarrow i}^T \times C_0^* \times H_{0 \leftarrow i}, \quad (2.22)$$

in which  $C_0^*$  is a normalized symmetric  $3 \times 3$  matrix, and it is the matrix format for the ellipse equation that fits the pupil boundary in the image when the eye is in the central position.  $C_i$  is a normalized symmetric  $3 \times 3$  matrix as well, it represents the “predicted” ellipse equation for the pupil when the eye is in another position  $i$ .  $H_{0 \leftarrow i}$  is a  $3 \times 3$  matrix determined by projection parameters  $M_r$ ,  $\alpha$ , and the horizontal and vertical eye position.

Let  $C_i^*$  be the detected ellipse fitting the pupil boundary from the image in position  $i$ . Ideally, we should have  $C_i^* = C_i$ . Comparing one such pair of  $C_i^*$  and  $C_i$  gives 5 equations, and 9 images of the eye in different positions give  $5 \times 8$  equations. Minimizing the cost function defined in (2.23) using Levenberg-Marquardt algorithm results in a solution to the projection parameters  $M_r, \alpha$ , which optimally fits the shapes of pupil boundary between different views of the eye.

$$efun = \sum_{i,j,k} \left\{ C_i(j, k) - C_i^*(j, k) \right\}^2 \quad (2.23)$$

### Summary of the Algorithm

In summary, Algorithm 1 outlines the procedure to find the camera projection parameters given the pupil center and shape of nine known eye positions. The entire algorithm is implemented in Labview. It takes less than 1s to finish the calculation after collecting the nine calibration images and the corresponding pupil center and shape. Since the system only does one time calculation for each set of tests, during which there is no considerable movement between the camera and the head of the subject, this time delay is acceptable for both the subject and the operator.

**Algorithm 1** The Calibration Algorithm

Input:  $\theta_k, \phi_k$  (known eye position ),  $u_k, v_k, C_k^*$  (pupil center and shape for the corresponding image),  $k = 1, \dots, 9$

Output:  $M_r, \alpha$

---

```

( $M_r, \alpha$ ) = mainCalibrationProc( $\theta_k, \phi_k, u_k, v_k, C_k^*$ )
   $minDist = 999$ ;
   $best\alpha = 0$ ;
  for  $\alpha = 0:0.1:1$  do
    ( $tM_r, t\alpha, aDist$ ) = solveWithKnownAlpha ( $\theta_k, \phi_k, u_k, v_k, \alpha$ );
    if  $aDist < minDist$  then
      Let  $minDist = aDist$ ;
      Let  $best\alpha = t\alpha$ ;
    end if
  end for
   $minDist_1 = 999$ ;
   $best\alpha_1 = 0$ ;
   $bestM_r = []$ ;
  for  $\alpha = best\alpha - 0.1:0.01:best\alpha + 0.1$  do
    ( $tM_r, t\alpha, aDist$ ) = solveWithKnownAlpha ( $\theta_k, \phi_k, u_k, v_k, \alpha$ );
    if  $aDist < minDist_1$  then
      Let  $minDist_1 = aDist$ ;
      Let  $best\alpha_1 = t\alpha$ ;
       $bestM_r = tM_r$ ;
    end if
  end for
  ( $M_r, \alpha$ ) = nonLinearOpt ( $bestM_r, best\alpha_1, \theta_k, \phi_k, u_k, v_k, C_k^*$ );
  return;

```

---

```

( $tM_r, t\alpha, aDist$ ) = solveWithKnownAlpha ( $\theta_k, \phi_k, u_k, v_k, \alpha$ )
  Solve the linear optimization problem defined by (2.21);
  return;

```

---

```

( $M_r, \alpha$ ) = nonLinearOpt ( $bestM_r, best\alpha_1, \theta_k, \phi_k, u_k, v_k, C_k^*$ )
  Solve the non-linear optimization problem defined by (2.23) using Levenberg-Marquardt algorithm;
  return;

```

---

### 2.3.2 Calculating Eye Position During Eye Tracking

In Section 2.3.1 we described how to estimate the projection parameters from images of known eye position. After the calibration procedure, during eye tracking tests, the system collects eye images, locates the pupil center and calculates the eye position; this calculation is based on the projection parameters estimated according to the method introduced in Section 2.3.1. This section describes the details of the procedure to calculate the horizontal and vertical eye positions. Here we assume that the system has collected the images and found the pupil center. The algorithm to calculate pupil center is discussed in Chapter 4.

From (2.17), (2.18), (2.19), we have:

$$[u, v, 1]^T \simeq A \overrightarrow{f(\theta, \phi)} + \vec{b}, \quad (2.24)$$

in which,

$$A = \begin{bmatrix} r_p m_1 & r_p m_2 & r_p m_3 \\ r_p m_5 & r_p m_6 & r_p m_7 \\ r_p m_9 & r_p m_{10} & r_p m_{11} \end{bmatrix}, \quad \vec{b} = [m_4, m_8, m_{12}]^T, \quad (2.25)$$

$$\overrightarrow{f(\theta, \phi)} = \begin{bmatrix} \sin \theta \cos \phi + \alpha \sin \theta (1 - \cos \phi) \\ (\alpha - 1) \sin(\phi) \\ \cos \theta \cos \phi + \alpha \cos \theta (1 - \cos \phi) \end{bmatrix}.$$

This means we need to find a constant  $k$  so that:

$$kA^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \overrightarrow{f(\theta, \phi)} + \vec{q}, \quad (2.26)$$

where  $\vec{q} = A^{-1}\vec{b}$ .

In (2.26),  $A, \vec{b}, \vec{q}$  does not change when the projection parameters do not change. There are three unknowns  $k, \theta, \phi$ , and three equations, which means the equations can be solved. We first assume  $\alpha = 0$ , and let:

$$g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = kA^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \vec{q}, \quad (2.27)$$

we have:  $\|g\| = \|\overrightarrow{f(\theta, \phi)}\| = 1$ . Since  $\|g\| = g_1^2 + g_2^2 + g_3^2 = 1$  is a quadratic function of  $k$ , it can be solved easily. The other two parameters can be solved as below:

$$\phi = \sin^{-1}\left(\frac{g_3}{\alpha - 1}\right), \quad \theta = \tan^{-1}\left(\frac{g_2}{g_1}\right) \quad (2.28)$$

Up to now, we obtained an approximate solution of horizontal and vertical eye position  $(\theta, \phi)$  by assuming  $\alpha = 0$ . To obtain a more accurate solution, we use this approximate solution as the initial value, and again use the Levenberg-Marquardt algorithm to find a local optimum by minimizing an error function defined as:

$$err = kA^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - \left( f(\theta, \phi) + \vec{q} \right) \quad (2.29)$$

The above algorithm to calculate horizontal and vertical eye position  $(\theta, \phi)$  is implemented in Labview and the run time is around  $0.01ms$ , which does not affect the typical high speed eye tracking at  $100 - 400$  frames/second.

## 2.4 Results

The above algorithm is tested using a simulation system we developed. Based on captured eye images, the simulation system synthesizes images in known eye positions with realistic pupil/iris features, noises and interferences, including eyelid coverage, cornea reflections, and illumination changes, etc. The details of the simulation system is described in Chapter 3. To evaluate the performance of different calibration algorithms, the simulation system generates two groups of images. One group of the images are used for calibration, in which the known eye position is used to calculate the projection parameters. After recovering the projection parameters with a specific calibration method, the second group of images are used to evaluate the accuracy of 3D eye position calculation, during which the known eye position is used as the ground truth.

In the calibration step, the pupil center, shape are first calculated from the images, and then as described in Section 2.3.1, the pupil center, shape and the corresponding known 3D eye position are used to estimate projection parameters such as  $M_r, \alpha$ . We observed that during actual tests, when the subject is instructed to look at one fixed point, the pupil still fluctuates around the position. The eye movement during the fixation can be up to  $0.5^\circ$ , with the standard deviation up to  $0.2^\circ$ . Therefore, for each planed eye position, 30 images are generated; for each of them the actual eye position used to generate the image is added with a Gaussian noise with zero mean and standard deviation of  $1.7^\circ$ . The average of the estimated pupil center (shape) for these 30 images is used as the pupil center (shape) for one *known* eye position.

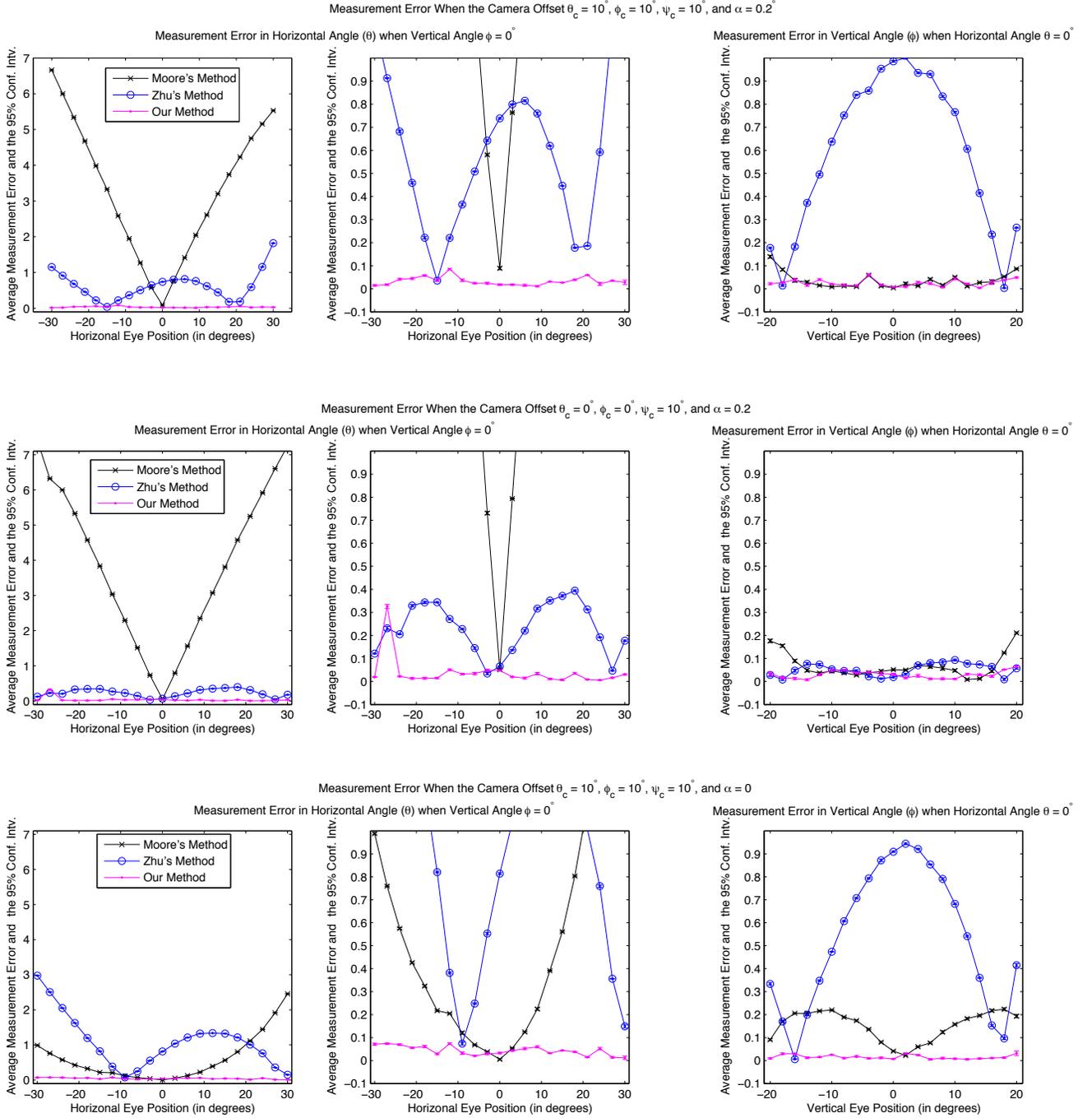
For the second group of images, the pupil center is also estimated, and with the projection parameters obtained in the calibration step, the 3D eye position is calculated according to the method in Section 2.3.2. The estimated eye position is then compared with the ground truth to find out

the error. For each image, besides interferences such as cornea reflections, occlusions, a Gaussian noise equivalent to the camera static noise is added as well.

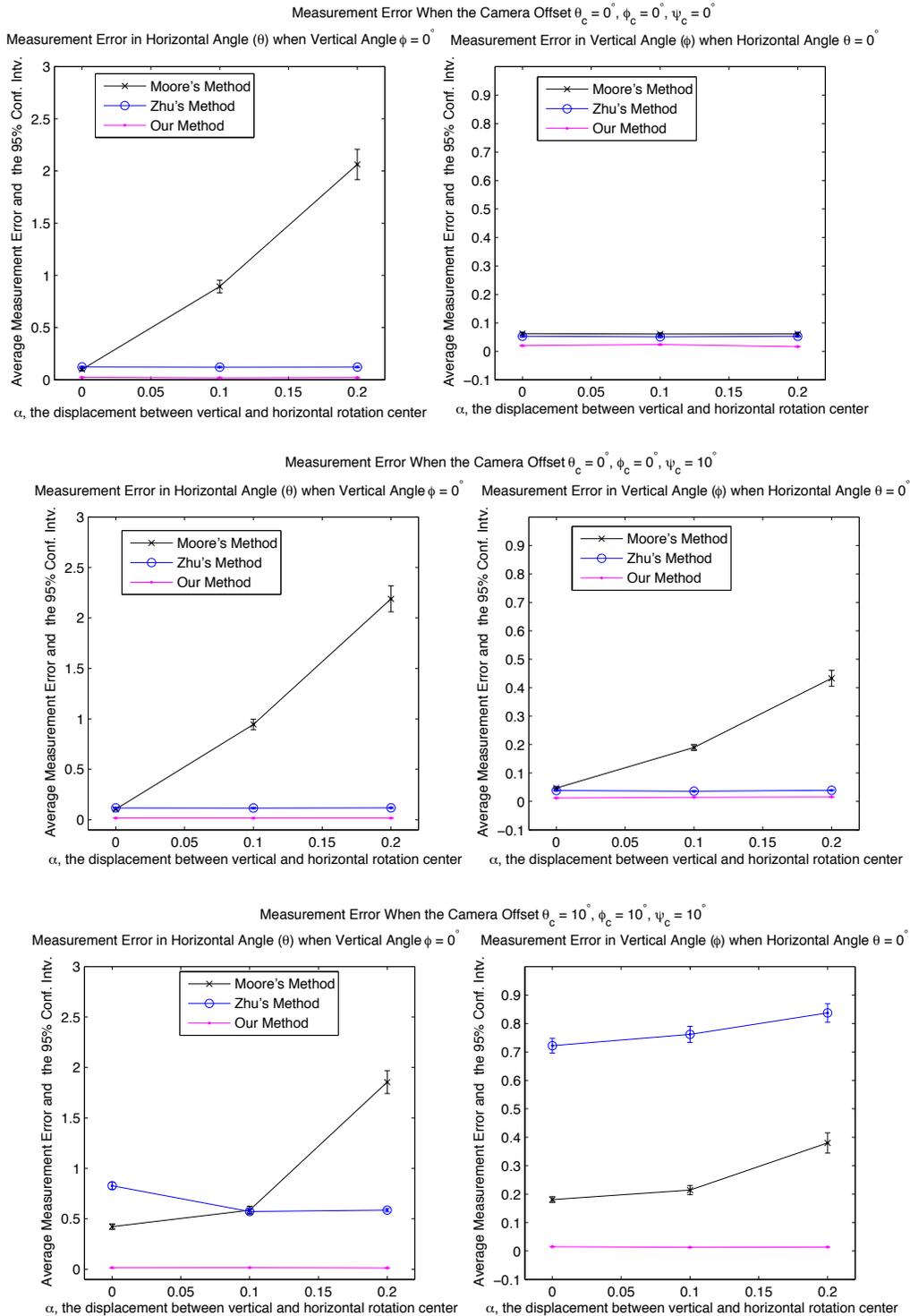
Figure 2.3 shows the measurement error based on three different calibration methods, Moore's method [18], Zhu's method [19], and our method. Three different rows show the performance with varying camera rotation offset  $\theta_c, \phi_c, \psi_c$ , and different  $\alpha$  value, which characterizes the difference in vertical and horizontal rotation center. Each point is the average of 30 images and a 95% confidence interval is plotted. From the plots, one can see that in all cases, the proposed method performs consistently better than the other two methods. In row one when  $\alpha = 0.2$ , by Moore's method, the vertical measurement is much more accurate than the horizontal measurement. This is because in Moore's method, the analytical solution to projection parameters is more dependent on the vertical pupil location; this results in large error in the horizontal estimation when the horizontal rotation center is different from the vertical center. When  $\alpha = 0$ , as shown in the third row, by Moore's method, the horizontal and vertical performance are in the same scale.

Comparing the first row and the second row, one can see that when  $[\theta_c, \phi_c, \psi_c] = [10^\circ, 10^\circ, 10^\circ]$ , the difference in the performance of Zhu's method and our method is larger than the case when  $[\theta_c, \phi_c, \psi_c] = [0^\circ, 0^\circ, 10^\circ]$ . That is due to the fact that in Zhu's method, the torsional camera rotation offset  $\psi_c$  is modeled explicitly, and the horizontal and vertical camera rotation offset  $\theta_c, \phi_c$  is not modeled explicitly.

In the third row, when  $[\theta_c, \phi_c, \psi_c] = [10^\circ, 10^\circ, 10^\circ]$ , and  $\alpha = 0$ , Moore's method performs better than Zhu's method. The reason for that is that Moore's method modeled all three camera rotation offset, but not  $\alpha$ , and in contrast, Zhu's method modeled  $\alpha$ , but not two of the camera rotation offset,  $\theta_c, \phi_c$ .



**Figure 2.3:** Measurement error with different calibration methods. The star data points show the performance of Moore’s method, the circle data points show the performance of Zhu’s method, and the dot data points show the performance of our proposed method. The left column shows the cases when the eye only moves horizontally, i.e.,  $\phi = 0^\circ$ . The middle column shows the same plots as the left column at a finer scale in the Y axis. The right column shows the cases when the eye only moves vertically, i.e.,  $\theta = 0^\circ$ . Note that the right column has the same scale in Y axis with the middle column.



**Figure 2.4:** Measurement error with different calibration methods. The star data points show the performance of Moore’s method, the circle data points show the performance of Zhu’s method, and the dot data points show the performance of our proposed method. The left column shows the cases when the eye only moves horizontally, i.e.,  $\phi = 0^\circ$ . The right column shows the cases when the eye only moves vertically, i.e.,  $\theta = 0^\circ$ . Note that the right column has different scale in  $Y$  axis with the first column.

Figure 2.4 shows the performance by different calibration algorithm with varying  $\alpha$ . Each point is the average of  $30 \times 42$  images in 42 different eye positions, and the 95% confidence interval is plotted as well. In the first row, when the camera rotation offset is  $[0^\circ, 0^\circ, 0^\circ]$ , Zhu's algorithm performs similar with our algorithm although slightly worse, and Moore's algorithm performs similar in vertical measurement but worse in the horizontal measurement with increasing  $\alpha$ . This is due to the fact that in Moore's method,  $\alpha$  is not modeled and the estimation of projection parameters are more based on the vertical pupil position.

In the second row when the camera rotation offset is  $[0^\circ, 0^\circ, 10^\circ]$ , Zhu's algorithm performs similar with our algorithm since  $\psi_c$  is modeled explicitly, while the performance of Moore's method is highly dependent on  $\alpha$ . In the third row, when the camera rotation offset is  $[10^\circ, 10^\circ, 10^\circ]$ , Moore's algorithm performs better than Zhu's method in the vertical measurement and also in the horizontal measurement when  $\alpha$  is small. That is again due to the fact that Zhu's method modeled  $\alpha$  but not  $\theta_c, \phi_c$ , and vice versa for Moore's method.

In all cases, the proposed method performs consistently better than the other two methods. In Table 2.3, the average and maximum measurement error are shown for a typical setting of camera rotation offset  $[10^\circ, 10^\circ, 10^\circ]$ , and  $\alpha = 0.2$ . One can see that for the proposed method, there is an order of magnitude improvement for the average measurement error compared with the other two methods.

## 2.5 Summary

We proposed a new method for calibration in horizontal and vertical eye position measurement, which can also be needed for geometric compensation in torsional measurement. Previous works

Method	Horizontal		Vertical	
	Avrg error	Max error	Avrg error	Max error
Moore's Method	1.85525	6.91488	0.37983	2.66614
Zhu's Method	0.58703	1.85614	0.83693	2.99331
Our Method	0.01255	0.04043	0.01355	0.07796

**Table 2.3:** Measurement error when the camera rotation offset is  $[10^\circ, 10^\circ, 10^\circ]$ , and  $\alpha = 0.2$

either ignore the horizontal/vertical rotation offset between the head frame and the camera frame, or approximate the imaging procedure as an orthographic projection, or assume a single center for horizontal and vertical eye rotation. These approximations produce large errors when the eye is at a large angle. With our proposed method, by explicitly modeling the imaging procedure as a perspective projection, considering all three rotation offsets between the head frame and the camera frame, and by modeling the eye rotation as a two-radius model, i.e., distinct horizontal and vertical rotation center, we have a more accurate model for the eye movement. With the help of linear and non-linear numerical methods, an algorithm is designed to solve the projection parameters during the calibration and calculate the horizontal and vertical eye rotation angles during online tracking. The calibration algorithm is implemented in Labview. It takes less than 1s to finish, which is acceptable for both the subject and the operator. The run time to recover the 3D eye position from pupil center during tracking is around 0.01ms, which does not affect the online tracking speed at 100 – 300 frames/second. Simulation results show that the proposed method brings the average error in measurements down to less than  $0.02^\circ$  for a wide range of eye position for both horizontal and vertical eye position, while the current best available method has an average error of  $0.59^\circ, 0.84^\circ$  for horizontal and vertical eye position respectively. As will be shown in the next

chapters, this improvement has serious implications and benefits.

# Chapter 3

## The Simulation System

### 3.1 Background and the Importance of Ground Truth

Video camera-based eye tracking systems are gaining more importance for their non-invasive nature and convenience of operation. In diagnosing and managing (e.g., rehabilitation) of balance disorders and other health problems, such eye measurement systems may play a key role. However, it is still difficult to evaluate and compare the performance of eye tracking systems and algorithms in terms of their accuracy, robustness, and capability to handle artifacts including occlusion and illumination changes. Although comparing the performance of video-based systems with the search-coil systems [9, 10] is one way for evaluation; however, this can be expensive and cumbersome.

The difficulty in evaluating such systems lies in the fact that no ground truth for the eye position is easily available. There is a number of video-oculography (VOG) systems available on the market and introduced in the literature [22–24]. To the best of our knowledge, however, there does not exist a software system that can produce realistic eye images in different positions, and provides the true

eye position that can be used for the evaluation of eye tracking systems. In this chapter, we present a novel eye-movement simulation system that generates realistic eye images and provides the ground truth. It can be used to evaluate the performance of different VOG systems or algorithms in tracking the horizontal, vertical and torsional positions. The simulation system uses a few samples of real eye images captured in known positions, and uses vision techniques to synthesize images in other user-specified positions. The system is also capable of simulating artifacts such as reflections, illumination changes, eyelid drops, etc. It provides a cheap, yet powerful and convenient way to evaluate the performance of eye tracking systems.

## **3.2 Proposed System: A Novel Simulation System Generating Realistic Eye Images**

### **3.2.1 Simulation of the Projective Geometry for the 3D Eye Movement**

The uniqueness of the system is that the synthesized images are based on real captured eye images. This ensures that the images generated have realistic pupil/iris features, noise level, and interferences, etc.

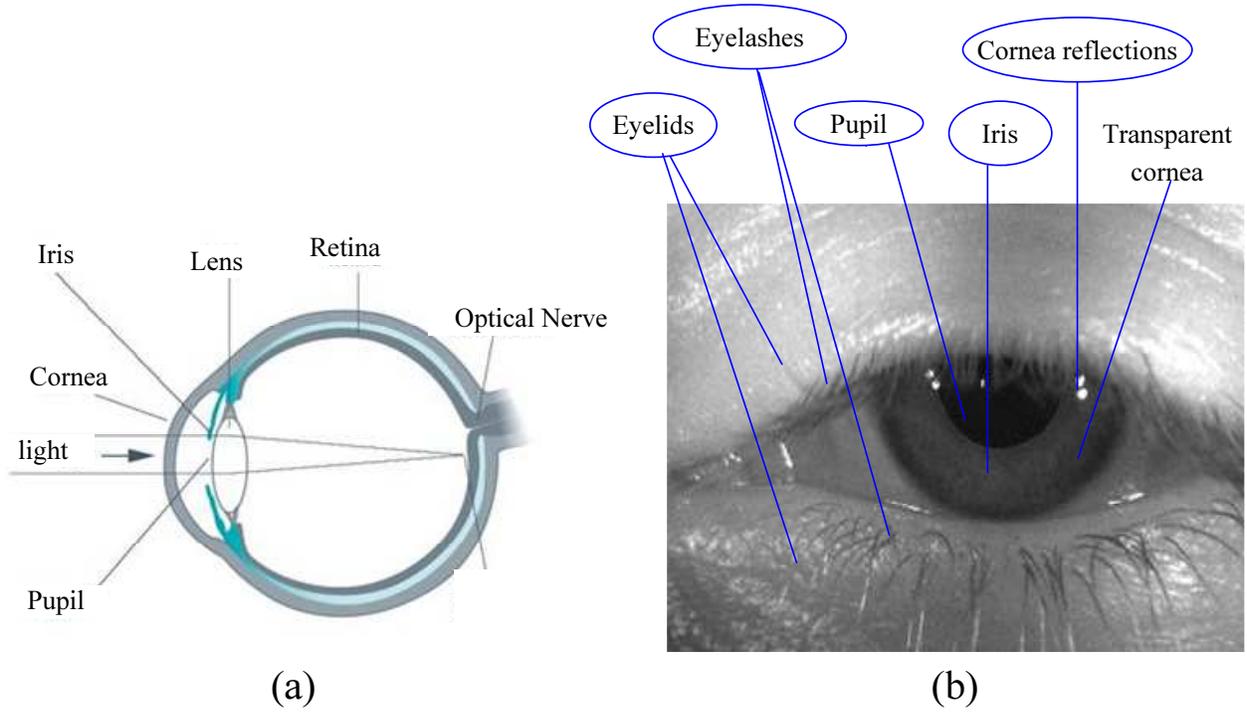
The input to the simulation system are nine template images captured in nine different known eye positions during calibration, as presented in Section 2.4. The projection parameters  $M_r, \alpha$  are calculated according to the calibration procedure as well. We developed a software to semi-automatically segment the pupil area, iris area, and the entire eye ball area between two eyelids. The eyelid area that can deform during eyelid movement are hand segmented for each image. Parameters including pupil center, pupil radius, and iris radius are calculated from each image.

Statistics including mean and variance for each area is calculated and saved. Based on the images and segmented areas, a soft 3D eye model is established. After obtaining the 3D eye model, the simulation system can then turn the eyeball and generate eye images by applying projective geometry.

Figure 3.1-(a) shows the diagram of the eye structure, and Figure 3.1-(b) shows a picture of the eye. The eyelids, eyelashes, pupil and iris area and location of cornea are shown. The geometry and photometric property of the eyeball, pupil, iris and cornea affects the resulting image; hence all these need to be simulated properly while synthesizing new images. In the eye model, the pupil/iris area is modeled as a plane in the space. The cornea and sclera area are modeled as sphere surfaces. Since only the pupil/iris area is where features are usually used for eye tracking, we ignore the detailed features in the sclera area, and only use its statistics to generate the pixels in the synthesized images. The cornea itself is transparent but reflections of lighting LED can act as interference to eye tracking. Simulation of cornea reflections is presented in the next section.

As to the the iris/pupil area, since we already know the projection parameters  $M_r, \alpha$ , with nine images and their position information available, it is possible to reconstruct the plane in the 3D space up to a scale factor. As long as the same scale factor is used to synthesize new images, the new images will be at the same scale as the real image. However, from the following analysis one can see that it is actually not necessary to reconstruct the plane explicitly to render new images.

To generate an image in a new eye position, for each pixel at  $[u, v]^T$  in the iris/pupil area, one needs to calculate its gray value. This is realized by locating its corresponding point in the 3D eye model with coordinate vector  $[x^e, y^e, z^e]^T$ , and then its corresponding position in one of the nine template images  $[u_t, v_t]^T$ . Then the pixel value of point  $[u, v]^T$  in the new image is set equal to the pixel value of  $[u_t, v_t]^T$  in the corresponding template image. Since  $u_t, v_t$  are usually floating point



**Figure 3.1:** (a) Diagram of the eye structure, (b) A picture of the eye.

variables, their pixel values are the interpolated values from the neighboring pixels.

From Section 2.3 we know that

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix} = M \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi & -t \sin \theta \cos \phi + t \sin \theta \\ 0 & \cos \phi & -\sin \phi & t \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & -t \cos \theta \cos \phi + t \cos \theta \\ 0 & 0 & 0 & 1 \end{bmatrix} R(\psi) \begin{bmatrix} x^e \\ y^e \\ r_p \\ 1 \end{bmatrix}, \quad (3.1)$$

in which  $\theta, \phi, \psi$  is the 3D position of the eye for which a new image is to be rendered. Let

$$R_{tp}(\theta, \phi) = \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}, \text{ and } F_C(t, \theta, \phi) = \begin{bmatrix} -t \sin \theta \cos \phi + t \sin \theta \\ t \sin \phi \\ -t \cos \theta \cos \phi + t \cos \theta \end{bmatrix}, \quad (3.2)$$

Equation (3.1) can be rewritten as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq M_r \begin{bmatrix} R_{tp}(\theta, \phi) & F_C(\alpha, \theta, \phi) \\ \mathbf{0} & 1 \end{bmatrix} R(\psi) \begin{bmatrix} \frac{x^e}{r_p} \\ \frac{y^e}{r_p} \\ 1 \\ 1 \end{bmatrix}. \quad (3.3)$$

If one plans to synthesize images for the same eyeball in a different environment, e.g., different offset angles between the camera and the head frame, one needs to simulate the camera geometry for different settings. In this case, the coordinate vector of pixels in the new image can be written as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq M_r^* \begin{bmatrix} R_{tp}(\theta, \phi) & F_C(\alpha^*, \theta, \phi) \\ \mathbf{0} & 1 \end{bmatrix} R(\psi) \begin{bmatrix} \frac{x^e}{r_p} \\ \frac{y^e}{r_p} \\ 1 \\ 1 \end{bmatrix}, \quad (3.4)$$

where  $M_r^*, \alpha^* = \frac{t^*}{r_p^*}$  are the new projection parameters. Note that this set of projection parameters may be different from the parameters we obtained from the calibration procedure. Since  $\theta, \phi, \psi$

are known, we can solve (3.4) for  $\frac{x^e}{r_p}$  and  $\frac{y^e}{r_p}$  as follows:

$$\begin{bmatrix} \frac{x^e}{r_p} \\ \frac{y^e}{r_p} \\ 1 \\ 1 \end{bmatrix} \simeq \begin{bmatrix} & & U & \\ 0 & & & \\ & 0 & & -1 \\ & & & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \\ 0 \end{bmatrix}, \quad (3.5)$$

in which,

$$U = M_r^* \begin{bmatrix} R_{tp}(\theta, \phi) & F_C(\alpha^*, \theta, \phi) \\ \mathbf{0} & 1 \end{bmatrix} R(\psi).$$

The relationship between the coordinate vector of a pixel in the template image and the coordinate vector of the actual point in the 3D space is determined by the imaging process while the image is captured:

$$\begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} \simeq M_r \begin{bmatrix} R_{tp}(\theta_t, \phi_t) & F_C(\alpha, \theta_t, \phi_t) \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \frac{x^e}{r_p} \\ \frac{y^e}{r_p} \\ 1 \\ 1 \end{bmatrix}. \quad (3.6)$$

From (3.5) and (3.6), one can solve for the pixel position  $[u_t, v_t]^T$  in the template image, which corresponds to the same physical point in the 3D space as the pixel in position  $[u, v]^T$  in the rendered image. In (3.6),  $\theta_t, \phi_t$  are respectively the horizontal and vertical position of the eye when the image is captured. It is assumed that the torsional angle  $\phi_t$  is zero.  $M_t, \alpha$  are the projection parameters obtained from calibration. As the first choice, the template image in the center position ( $\theta_t = 0, \phi_t = 0$ ) is always used. However, there is a possibility that part of the iris is occluded by eyelids or cornea reflections. In this case, another template image is tried at random until a template image is found in which the part of the iris is visible. As the last step of the geometric

simulation, we set the pixel gray value in the new image to be equal to its corresponding pixel gray value in the template image:

$$I_g(u, v) = I_g(u_t, v_t) \quad (3.7)$$

We will see in the next section that, because of illumination change, new components need to be added for the gray value.

### 3.2.2 Photometric Simulation

In the previous section, it is assumed that there is no illumination change when the eye rotates, i.e., the same part of the iris produces exactly the same pixel gray value at different angles. This is not exactly true in reality. The LED light sources are usually close to point sources and they are located close to the eye ball. They shed different amount of light onto the iris plane when the plane is at different angles, and therefore results in different gray value in images. The surface of cornea will produce both reflecting and diffusing light. The consequence is that the images show various illumination changes when the eye is at different angles. The cornea reflections change their shape and location during eye motion as well. An eye tracking system that does not deal with these factors is very likely not to be robust. Therefore, to evaluate eye tracking systems and algorithms, photometric simulation is important to generate realistic images for various eye positions.

The LED light sources are modeled as point sources. In general, the radiance at a point  $P$  on the object surface is:

$$L(P, \theta_o) = \sum_i \rho(P, \theta_o) \cdot B_i(P) = \sum_i \rho(P, \theta_o) \frac{\mathbf{N}(P) \cdot \mathbf{S}(i)}{r_i^2}$$

in which  $\theta_o$  is the direction of the outgoing light,  $\mathbf{N}(P)$  is the surface normal,  $\mathbf{S}(i)$  is the direction of light source  $i$ ,  $\rho(P, \theta_o)$  is determined by the surface material and is dependent on the outgoing

direction, and  $r_i$  is the distance to the light source.

The iris surface is more like a diffuse surface, so it is modeled as a Lambertian surface, i.e., a perfect diffuse surface, for which the radiance  $L_{iris}(P, \theta_o)$  can be written as:

$$L_{iris}(P, \theta_o) = \sum_i L_{iris}(i, P, \theta_o) = \sum_i \left( \rho_1(P) B_i(P) \right),$$

in which  $\rho_1(P)$  depends on the surface material but does not depend on outgoing direction  $\theta_o$ . The cornea reflection is a glossy surface, but also has some diffuse. Therefore, we model the cornea surface reflection as a combination of a Lambertian component and a specular component. We use the common *Phone's* model as shown in Figure 3.2 to approximate the specular component of the cornea surface reflection from source  $i$  as:

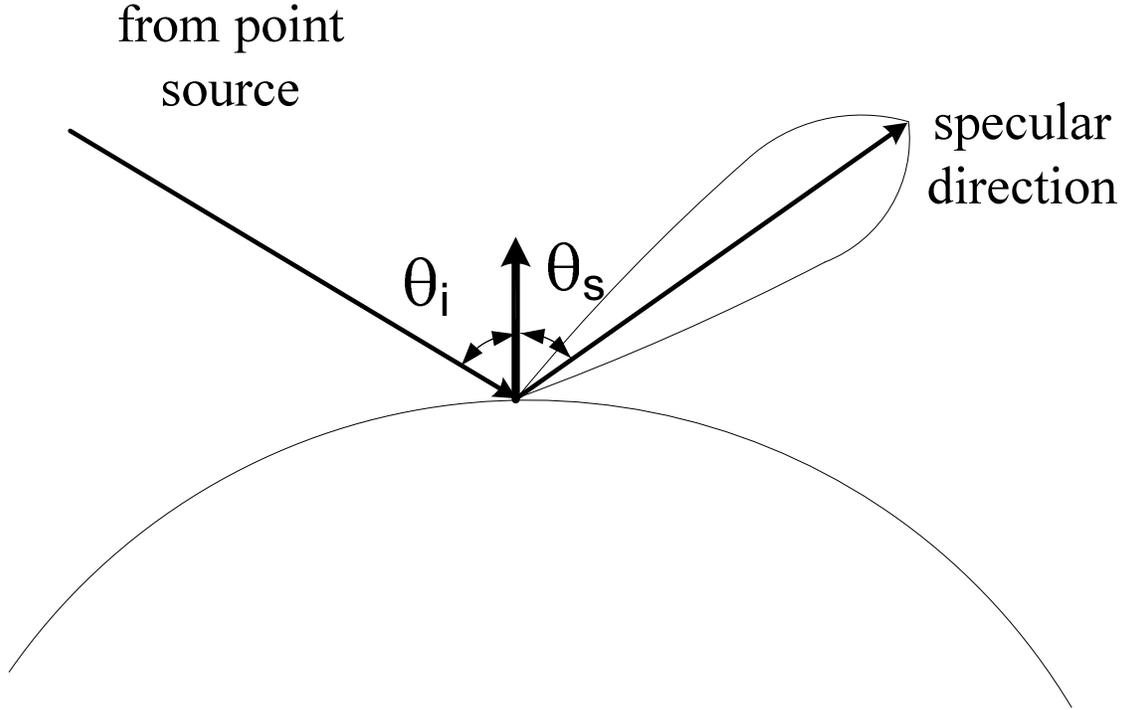
$$L_c(i, P, \theta_o) = \sum_i \left( \rho_d B_i(P) + \rho_s B_i(P) \cos^n(\theta_s(i) - \theta_o) \right),$$

in which  $\theta_s$  is the direction of specular direction, and  $\theta_o$  is the direction of the outgoing light, and  $\rho_s, \rho_d$  depend on the surface material.

Assuming that the camera response, i.e., the gray value in the image  $I(P)$ , is linear in the surface radiance, we have:

$$\begin{aligned} I(P^i) &\sim L_{iris}(P_r, \theta_o) + \sum_i L_c(i, P_c, \theta_o) \\ &\sim \sum_i \left( \rho_1(P_r) \frac{\mathbf{N}(P_r) \cdot \mathbf{S}(i)}{r_i^2} + \left( \rho_d + \rho_s \cos^n(\theta_s(i) - \theta_o) \right) \frac{\mathbf{N}(P_c) \cdot \mathbf{S}(i)}{r_i^2} \right) \end{aligned} \quad (3.8)$$

in which,  $\theta_o$  is the camera direction relative to the surface,  $P^i$  is the image point, and  $P_r, P_c$  are the corresponding point on the iris and on the cornea, respectively. In the photometric simulation, the values of  $\rho_1, \rho_d, \rho_s$  are selected so that the gray value of pixels generated are close to the template images, which are captured during the calibration process and serve as the input to the simulation system.

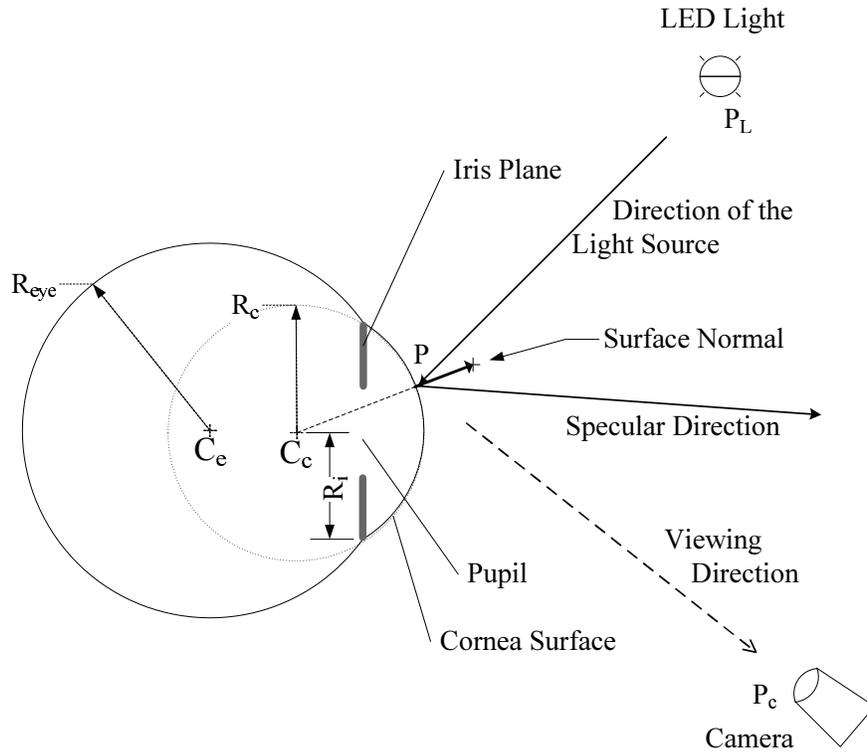


**Figure 3.2:** Phone's model used to model the specular component of the cornea surface reflection. Specular surfaces usually reflect light into a lobe of directions around the specular direction, and the reflection intensity depends on the direction.

To determine the location of cornea reflections in the image, we model the cornea as a spherical surface. Given the LED positions in the space, we determine the reflection center on the surface and then bright spots are projected onto images. This simulates the position change of the reflections in the images when the eye moves. In Figure 3.3, both the sclera surface and the cornea surface are modeled as sphere surfaces. In the eye frame, we have:

$$\begin{aligned}
 \overline{C_e C_c} &= \left[ 0, 0, \sqrt{R_{eye}^2 - R_i^2} - \sqrt{R_c^2 - R_i^2} \right]^T, \\
 \overline{C_e P} &= \overline{C_e P_L} - \overline{C_e C_c}, \\
 \overline{P P_L} &= \overline{C_e P_L} - \overline{C_e P}, \\
 \overline{P P_C} &= \overline{C_e P_C} - \overline{C_e P}.
 \end{aligned} \tag{3.9}$$

in which,  $P_L$  is the position of the LED source in the eye frame, and  $P_C$  is the position of the camera in the eye frame. Therefore, the normalized surface normal vector is  $\vec{n} = \frac{\overline{C_e P}}{\|\overline{C_e P}\|}$ , and the



**Figure 3.3:** The eye model used to calculate the position of reflections in the image.  $R_{eye}$ ,  $R_c$ ,  $R_i$  are radius of the eye ball, the cornea sphere, and the iris plane. The specular direction and the light source direction are symmetric with respect to the surface normal. For each different eye position, the light source position and the camera position coordinates need to be recalculated (transformed into the eye coordinate system).

normalized vector for the source direction is:  $\vec{l} = \frac{\overrightarrow{PP_L}}{\|\overrightarrow{PP_L}\|}$ , in which the notation  $\|\vec{v}\|$  represents

the  $L_2$  norm of a vector  $\vec{v}$ . The normalized vector in the specular reflection direction can be written

as:

$$\vec{s} = (2 \cdot \vec{n} \cdot \vec{l}) \vec{n} - \vec{l}$$

The vector  $\vec{s}$  is then compared with the camera direction, and the point with the smallest disparity is considered the center of the reflection.

### 3.2.3 Simulation of the Eyelid Movement

Occlusions from the eyelash and eyelids are common artifacts in eye images. A good tracking algorithm should be robust to these interferences. To introduce realistic interferences, we simulated the eyelid movement as well.

In the template image, the software prompts the user to hand mark the upper eyelid and lower eyelid area. Figure 3.4 shows the deformation process for the upper eyelid area. The deformation process for the lower eyelid is done in a similar way. When eyelid movement is desired, e.g., an upper eyelid drop of  $d$  pixels, the middle point  $M_o$  is moved down  $d$  pixels to  $M_n$ , and the new upper eyelid curve is fitted as a circle from the three points  $E_1, M_n, E_2$ . The gray value for the eyelid deformation area in the new image  $I_n(x, y)$  is related with the gray value for the area in the old image  $I_o(x, y')$  according to the following formula:

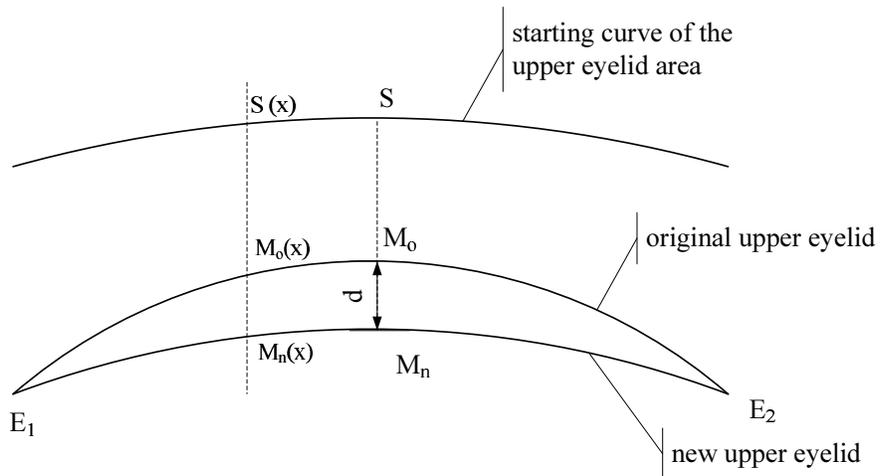
$$\begin{aligned} I_n(x, y) &= I_o(x, y'), \\ y' &= \left( y - S(x) \right) \frac{M_o(x) - S(x)}{M_n(x) - S(x)} + S(x). \end{aligned} \tag{3.10}$$

## 3.3 Results

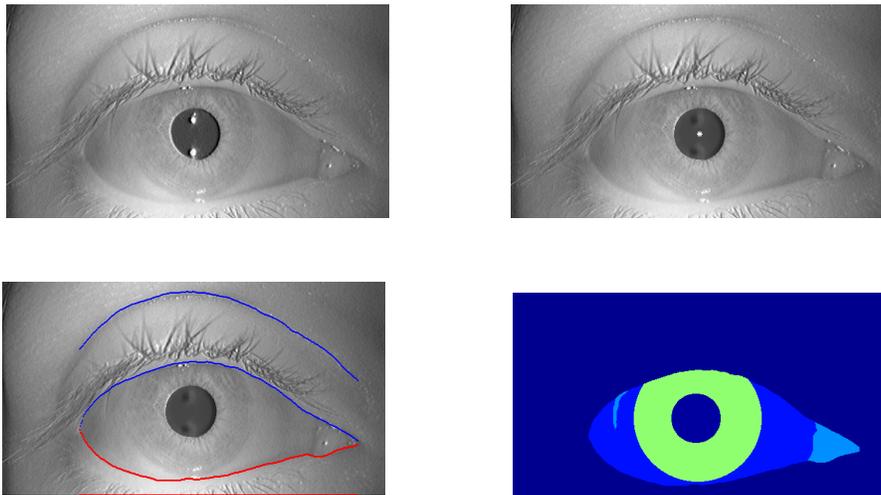
In this section, we present some examples for template images, the preprocessing procedure, and examples of some synthesized images. Figure 3.5 shows the original template image captured, the preprocessed template images with pupil center or eyelid boundary marked, and the segmented region in the image.

Figure 3.6 shows four images with different eyelid movement while the eye is in the same position.

Figure 3.7 shows four images when the eye is in different positions, and with synthesized cornea

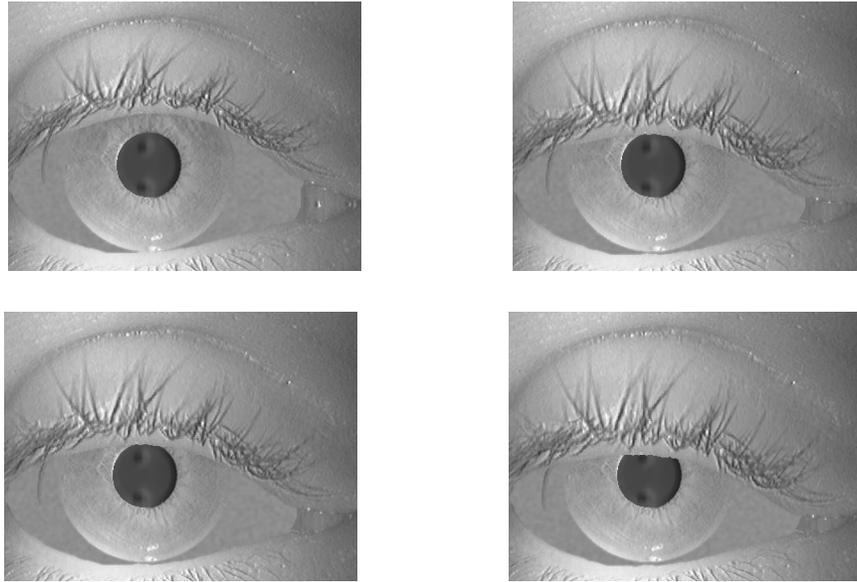


**Figure 3.4:** Deformation of the upper eyelid. The starting curve of the upper eyelid area and the original upper eyelid is hand-marked by the user.  $E_1, E_2$  are the beginning and ending columns of the eyelid deformation area.  $d$  is the amount of eyelid movement. A column line intersects the starting curve, the original upper eyelid and the new upper eyelid at  $S(x), M_o(x), M_n(x)$  lines, respectively.

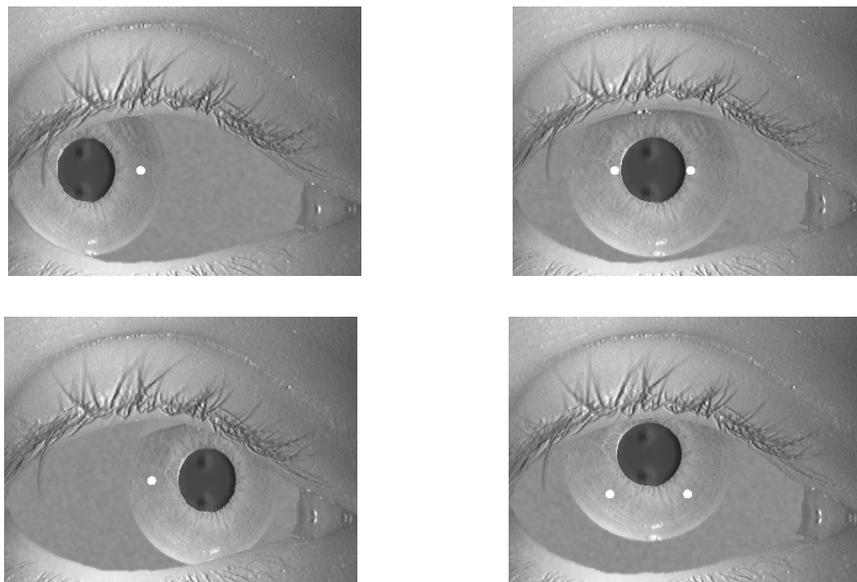


**Figure 3.5:** Preprocessing of template images. The upper left image shows the template image captured. The upper right image shows the image with the cornea reflection filled and the pupil center marked. The lower left image shows the marked eyelid boundary. The boundary of the upper eyelid area is marked in blue and the boundary of the lower eyelid area is marked in red. The lower right image shows different regions segmented.

reflections.



**Figure 3.6:** Images with different eyelid movement while the eye is in the same position



**Figure 3.7:** Images with different eye movement

### 3.4 Summary

In this chapter, we presented a simulation system that can generate video sequences in different known eye positions based on real captured eye images. Three dimensional eye movement is simulated by applying geometric transformation for the imaging process. Photometric simulation is

implemented to simulate different reflection properties of the iris and the cornea. Eyelid movement is simulated as well to add interferences similar to the real situation. These generated images can be used to evaluate the performance of calibration or eye tracking algorithms since there is ground truth information available for the eye position.

## **Part III**

# **Eye Movement Tracking**

# Chapter 4

## Horizontal and Vertical Eye Tracking

### 4.1 Existing Online Eye Tracking Systems

As mentioned before, a high-speed, accurate and online eye position measurement is desired for both laboratory and clinical usage. Saccadic eye movement can approach the speed of  $700^\circ/sec$  and can have dynamics in the time frame of tens of milliseconds. It is essential for a good eye tracking system to run at high-speed to capture this dynamics, which are important for diagnostic purposes. Digital image based online eye tracking systems available in industry typically run at  $60 Hz$  for binocular systems. To capture subtle eye movement in saccadic motion, a high time resolution of  $150 Hz$  for horizontal and vertical tracking is a must [25]. There are some existing systems that can capture eye movement at high frame rate. Clarke presents a system at a sampling rate of  $400 Hz$  by using smart sensors and a custom-designed DSP/FPGA architecture for pre-processing [26]. The SMI (SensoMotoric Instruments) system reaches a speed of  $500 Hz$  and also involves specialized hardware [22]. The customized hardware design induces considerable cost and development time.

Detecting the pupil is the most frequently used method to track the horizontal and vertical eye position [27, 28]. Unfortunately, most early pupil detection systems make the oversimplifying assumption that the pupil is circular, and that the pupil's center is at the mass center of the circular object; the algorithm is commonly referred to as the *center of mass* algorithm. In practice, even a circular pupil appears elliptical in eccentric eye position, and the pupil "object" is frequently occluded by the eyelids, eyelash and cornea reflections. Zhu et al. proposed to use the curvature characteristics of the pupil boundary and fit them to an ellipse [29]. This technique gives a more robust and accurate estimation of the pupil center. Starburst is an eye tracking system that uses a hybrid algorithm combining feature based (edge point extraction) and model-based approaches (ellipse fitting) [28]. However, the complexity of these methods adversely affects the processing rate.

In a typical pupil detection procedure based on the *center of mass* algorithm, the captured image is first transformed into a binary image by a user-set threshold. Everything under the threshold is labeled as "one" (considered object pixels) and everything else is labeled as "zero" (considered as the background). Then, one can use a blob analysis algorithm such as the Labview Virtual Instrument (VI), "IMAQ Complex Particle", to find objects (consisting of connecting object pixels) in the image. Finally, the software chooses the largest object, which is the pupil, and calculates the geometric center of the object.

For images captured with a typical head-mounted camera, a  $320 \times 240$  pixel region of interest (ROI) window is usually sufficient to capture the whole dynamics of eye movement assuming the eye is right at the center of the ROI window when the subject is looking straight ahead. However, because of variations on people's shape, sometimes it is difficult to adjust the wear-on goggle to make sure that the eye is right at the center of the captured image area. Therefore, it is necessary

to capture a larger image area and then set the ROI window for processing. In addition to the processing time, computation cost of capturing the images, user interface monitoring, and image displaying also slow down the overall frame rate.

The aforementioned factors limit the performance of the binocular system. If the full size images are captured and processed directly, the maximum system frame rate (image capturing and interface updating time included) one can currently achieve is about 80  $Hz$  on a Pentium IV computer system. We aim to find a software solution to increase the processing rate.

## **4.2 Proposed System: A High-Speed Online Eye Tracking System**

This chapter presents a high-speed head-mounted binocular eye tracking system we have recently developed. The herein reported results clearly indicate that it is possible to measure the eye movement at a speed of 150  $Hz$  by just using a general purpose computer system. The system utilizes off-axis infrared lighting and the black pupil technique so that the pupil area can be easily extracted. To detect the pupil center at high-speed, the system employs a dual-working-mode mechanism and a two-step processing algorithm. First, an approximate pupil location is searched in a low resolution image, then the image area around the approximate location is processed and an accurate pupil center is calculated. We observed that droopy eyelids' partially covering the pupil is a quite common phenomenon. Hence, a *symmetric mass center* algorithm based on geometric properties of ellipses is developed to achieve a more accurate measurement when the eyelid is partially occluded.

The approach presented in this section for finding pupil center is an improvement over the naive

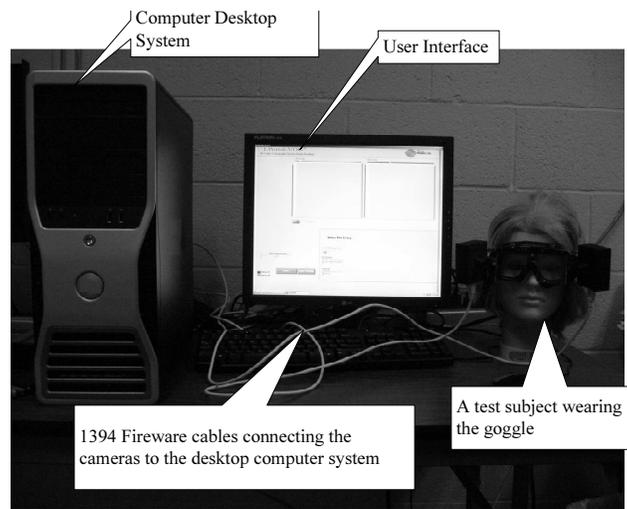
*center of mass* algorithm. The symmetric mass center algorithm we developed gives a more robust measurement for the eye position. Processing frame rate is enhanced by eliminating the bottleneck in transferring and processing of the entire image. Locating the approximate position of the pupil at a low-resolution image can significantly decrease the region that needs processing for the feature extraction step. Our algorithm improves the accuracy compared with the naive mass center algorithm while maintaining the high-speed operation.

### 4.2.1 System Overview

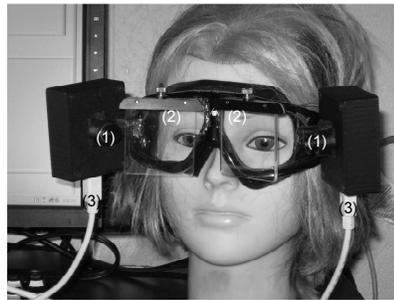
Component type	Specification	Number
computer desktop system	Pentium IV 3 <i>GHz</i> CPU 1G RAM, 800 <i>MHz</i> bus	1
software platform	Windows XP Pro, Labview 7.1	1
interface Card	IEEE 1394b Fireware, 2 Port	1
high-speed camera	Point Grey DragonFly Express	2

**Table 4.1:** System configuration

Table 4.1 shows the main components of the system. Figure 4.1 shows the goggle (the high-speed eye tracking system) with the two cameras installed. Two hot mirrors are used to reflect the eye images to the cameras and at the same time allow a broad viewing angle for the subjects. Infrared LED is used for illumination and infrared pass filters are installed before the cameras to exclude light from the environment. Two cables connect the cameras to the IEEE 1394b Fireware card installed in the computer desktop system.



(a)



(b)

**Figure 4.1:** (a) The system components.

(b) A subject wearing the goggle with two cameras installed. The numbers in the picture indicate: (1) Two cameras installed on the goggle, with infrared pass filters installed before the camera lens. (2) Hot mirrors reflecting eye images to cameras. (3) 1394 Fireware cable connecting the cameras to the desktop computer system.

The images captured by the cameras are transferred to the desktop (or laptop) computer. The image processing software we developed extracts the pupil and calculates its geometric center. From the pupil center, the horizontal ( $\theta$ ) and vertical ( $\phi$ ) rotation angle of the eye are calculated based on projection parameters according to the approach introduced in Section 2.3.2.

The software first works in the low-speed mode when full size images are captured and displayed so that the operator can select the ROI; then the software can be switched to the high-speed mode

when only the partial image inside the ROI window is captured. In the high-speed mode, a two-step processing algorithm first processes the image in the ROI window at a low pixel density resolution to find the approximate location of the pupil. After that, a tiny trace window containing the entire pupil area is defined and further processing is done in the trace window to calculate the precise location of the pupil center. The system increases the processing rate for horizontal and vertical eye position measurement up to 150  $Hz$  on a Pentium IV 3  $GHz$  computer system for both cameras, displaying measurement results and captured eye images on the interface. It achieves this high-speed by eliminating the bottleneck caused by capturing and processing the full size images without sacrificing convenience and spatial resolution.

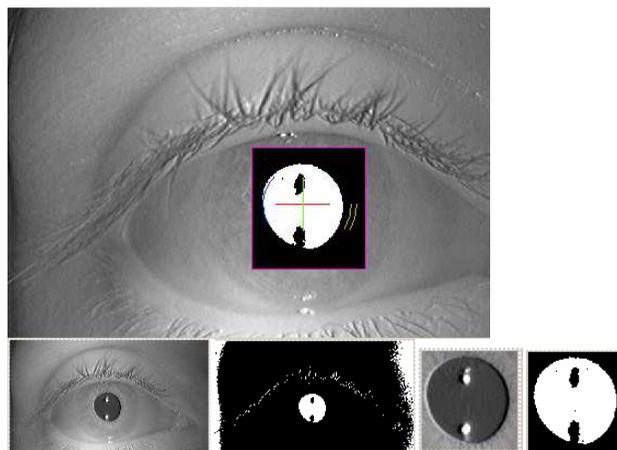
#### **4.2.2 High-Speed Pupil Detection: The Dual-Mode Capturing Software and the Two-Step Processing Algorithm**

To enhance the processing rate, we observed that the size of the image captured and processed is the key to the problem. The pupil itself is pretty small, typically less than  $120 \times 120$  pixels, but it moves around. If one can find the approximate location of the pupil first, one can then process a much smaller image to find the more accurate location of the pupil center.

We developed the new dual-mode capturing software with a *two-step* processing algorithm so that it can achieve a high frame rate without sacrificing the accuracy. We first capture and process the images at full size and full resolution ( $640 \times 480$ ) using the *center of mass algorithm* as described in Section 4.1. The operators can set the threshold and adjust the ROI window in this low-speed mode. After that, the user can switch to the high-speed mode. In the high-speed mode, the system captures the partial image in the ROI window specified above. This makes the

Fireware 1394b interface board capable of transmitting images at a higher frame rate and reduces the processing time as well.

To reach an even higher speed, we developed an algorithm to first track the approximate position of the pupil and then determine a tiny image window enclosing the pupil for further processing. The image is downsampled at  $\frac{1}{4} * \frac{1}{4}$  rate of the original resolution, and again the *center of mass algorithm* is used to locate the approximate pupil center. Then, the system specifies a tiny trace window slightly larger than the detected pupil area so that we do not lose any pixels. We process the tiny image at full pixel density resolution using the symmetric mass center algorithm described below to locate the accurate position of the pupil center.

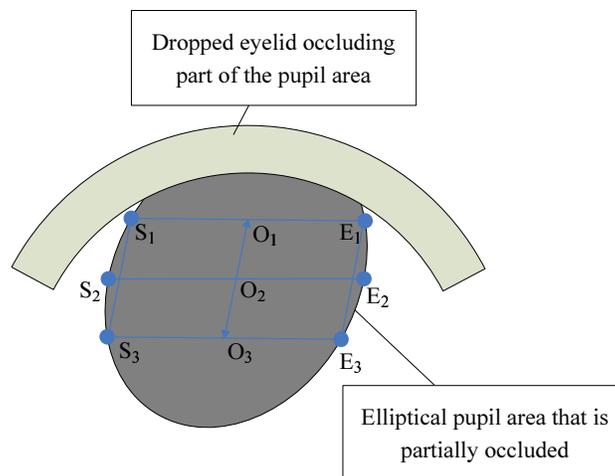


**Figure 4.2:** The upper row shows a full pixel density resolution grey-level image (the partial image defined by the ROI window), with the small threshold image inlaid on the right location. The lower row shows, from left to right, the downsampled grey-level image, the threshold image of the downsampled image, the grey-level image of full pixel density resolution in the tiny trace window, and the small threshold image in the tiny trace window.

Figure 4.2 shows the grey-level images and threshold images in different resolution. “Object” pixels close to the image borders are caused by camera distortion; they can be eliminated by the

simple criteria of being connected to the borders, and this step can be completed in the low-speed mode. By implementing the dual-mode capturing and the two-step processing algorithm described above, the current software reaches a final system speed of 150 Hz (binocular system) in the high-speed mode, including proper image and data display on the user interface.

### 4.2.3 The Symmetric Mass Center Algorithm



**Figure 4.3:** Locating the center for an elliptical pupil area that is partially occluded. The algorithm finds the maximum parallelogram  $S_1E_1E_3S_3$ , and calculates the mass center of the area enclosed by  $\widehat{S_1S_3E_3E_1}$

In the high-speed mode, after the approximate pupil location is obtained, we extract the image in the tracing rectangle at full pixel density resolution and apply a new symmetric mass center algorithm to calculate the pupil center coordinates. The center of mass algorithm assumes that the pupil area can be approximated by a circle or an ellipse. This assumption will be violated when the eyelid covers part of the pupil area and it leads to large measurement error. The symmetric mass center algorithm utilizes only the non-occluded, rotational symmetric portion of the pupil area to estimate the pupil center.

---

**Algorithm 2** Pseudo code for the symmetric mass center algorithm

Input: **ImIn** (input image)

Output: **(ox, oy)** (pupil center)

---

**(ox, oy) = mainFindPupilCenter(ImIn)**

**Imt** = DownSample (**ImIn**)

**Imb** = NMassCenter (**Imt**, **ImIn**)

**(ox, oy)** = SymmetricMassCenter (**Imb**)

**return** (**ox, oy**)

**SymmetricMassCenter (Imb)**

**for**  $i =$  each image line in **Imb** **do**

$\{S_i, E_i\} =$  ScanImgLine (**Imb**,  $i$ )

$Len_i = \|S_i E_i\|$

**end for**

**for**  $i =$  each image line in **Imb** **do**

$parag(i) =$  Find\_Parallelogram (**Imb**,  $Len_i$ ,  $i$ )

$cen(i) =$  Find\_Parallelogram\_Center ( $parag(i)$ )

**end for**

$cluster\_cen =$  Find the statistic cluster center of all  $cen(i)$

$maxParag =$  FindMaxParag ( $parag(i)$ ,  $cluster\_cen$ )

    Let  $maxParag$  be the maximum parallelogram,  $S_1 E_1 E_3 S_3$

**(ox, oy)** = PMassCenter (**Imb**,  $S_1 E_1$ ,  $S_3 E_3$ )

**return** (**ox, oy**)

---

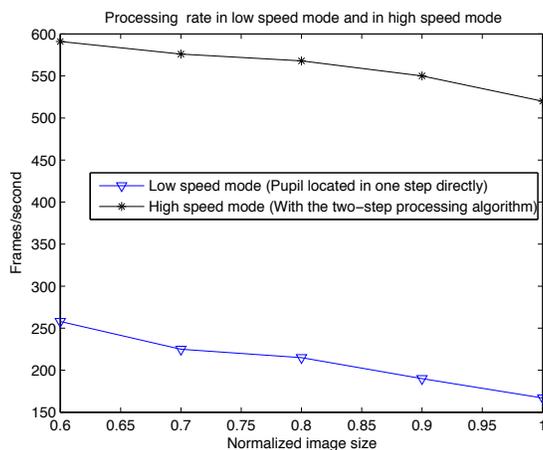
Suppose that an ellipse is partially occluded as shown in Figure 4.3, the mass center of the non-occluded area will not yield the correct ellipse center. From geometric properties of ellipses, we know that if we find the maximum parallelogram  $S_1 E_1 E_3 S_3$ , the mass center of the area enclosed by  $\widehat{S_1 S_3 E_3 E_1}$  is a good estimate of the ellipse center, given a significant area of the ellipse is not occluded. We designed our symmetric mass center algorithm according to this principle, and the pseudo code is shown in Algorithm 2. In the code, *NMassCenter* is the procedure that implements the naive mass center algorithm; it finds the rectangle area around the pupil, and outputs a small binary image of the pupil with full pixel density. *ScanImgLine* is a procedure that locates the starting and ending object pixel in the image line. *Find\_Parallelogram* is the procedure that searches for a matched image line (having the same length of object pixel) for current image line  $i$ , and outputs the parallelogram formed by the current image line and its matched line. *FindMaxParag* is

the procedure to find the maximum parallelogram, which has the maximum height and has a center close to  $cluster\_cen$ .  $P_{MassCenter}$  finds the mass center of the partial object falling between line  $S_1E_1$  and line  $S_3E_3$ ; they represent respectively the top and bottom line of the “symmetric” part of the ellipse.

## 4.3 Results

### 4.3.1 Speed of Horizontal and Vertical Position Measurement

We tested the processing rate of the software with a Pentium IV 3 GHz computer. The data is obtained by applying different algorithms on the image of same size, and without including the time for acquiring the image from the cameras and the time for user interface updating.



**Figure 4.4:** Processing rate with different ROI size. By implementing the two-step processing algorithm, the processing rate is about 2-3 times faster than locating the pupil center directly at full pixel density resolution.

Figure 4.4 shows the result. The solid line with triangles shows the processing rate (for single image) in the low-speed mode when pupil center is located in one step. The solid line with asterisks

shows the processing rate (for single image) in the high-speed mode when the two-step algorithm is applied. The  $x$  axis is the normalized image size in the ROI window defined. Table 4.2 lists the normalized image sizes and the corresponding dimensions of the ROI window. Although this is not the final system speed, it shows the effect of employing the two-step processing algorithm. Observe that the new algorithm introduced a considerable performance improvement in terms of processing rate.

ROI size	$400 \times 300$	$380 \times 250$	$340 \times 260$	$335 \times 250$	$310 \times 230$
normalized size	1	0.9	0.8	0.7	0.6

**Table 4.2:** Dimensions of the ROI window and the corresponding normalized image size

### 4.3.2 Accuracy of the Symmetric Mass Center Algorithm

In this section, we compare the accuracy of measured eye position for different algorithms by artificially varying the degree that the pupil area is occluded. The pupil center obtained when the pupil is not occluded is considered the ‘true’ center. Figure 4.5-(a) shows an eye image with the full pupil area viewable on the left and an image with pupil partially occluded on the right. The portion of pupil boundary that is not occluded is highlighted in red color.

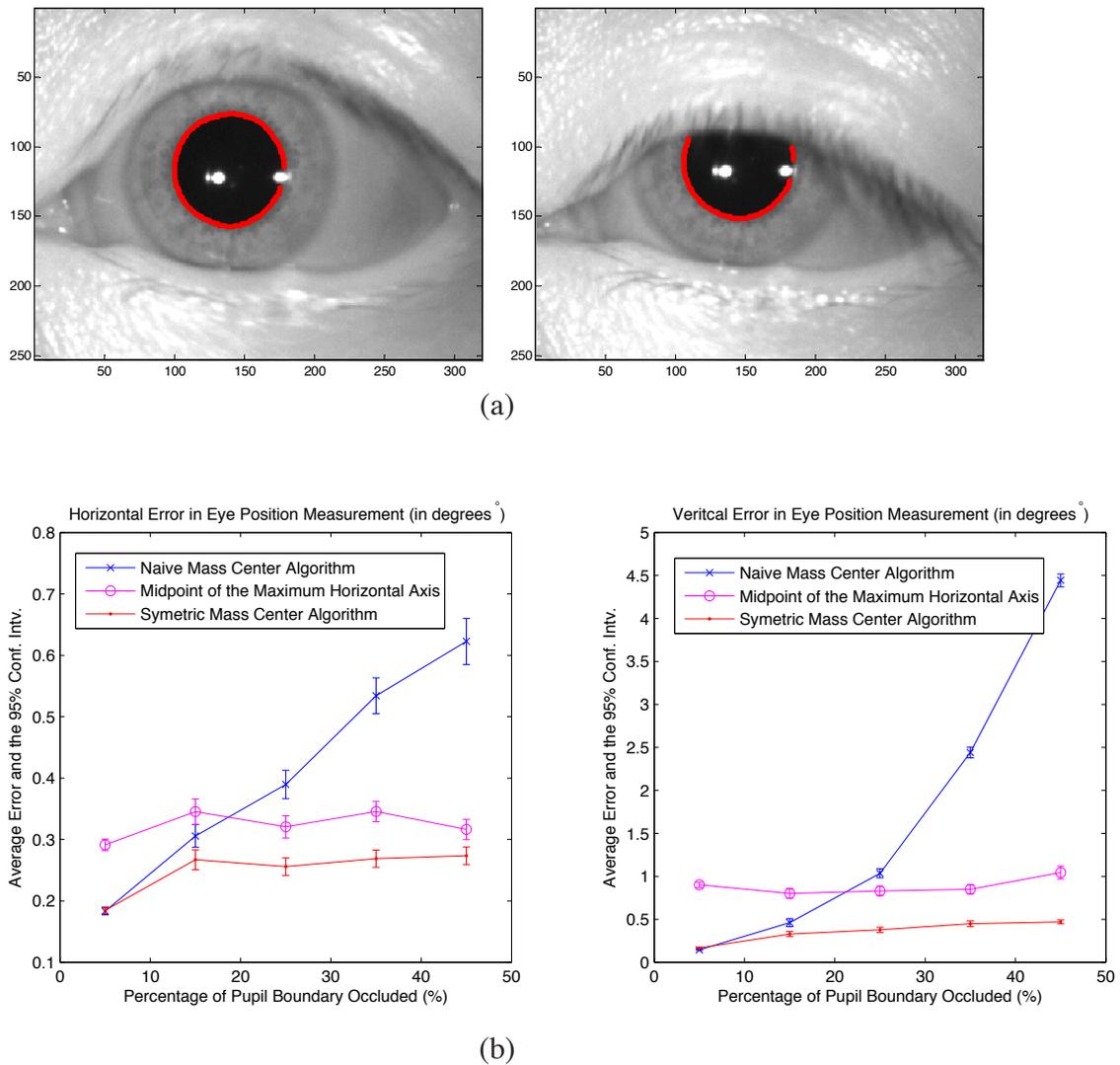
Experiments are performed to compare three different algorithms: the original naive center of mass algorithm, the algorithm that uses the midpoint of the maximum horizontal axis as the pupil center (the midpoint algorithm), and the symmetric mass center algorithm. Using the simulation system introduced in Chapter 3, 3780 images are generated in different eye positions and with different levels of occlusion. The average error and the 95% confidence interval on horizontal and vertical measurement for each occlusion level is calculated; the occlusion level is indicated by the

percentage of pupil boundary being occluded. In Figure 4.5-(b), the left figure shows the error in the horizontal measurements, and the right figure shows the error in vertical measurements. The line with crosses shows the result of the naive center of mass algorithm; the line with circles shows the result of the algorithm using the midpoint of the maximum horizontal axis, and the line with dots shows the result using the symmetric mass center algorithm. The plot shows that when the percentage of the pupil boundary being occluded increases, both the horizontal and vertical measurement error for the naive mass center algorithm increases significantly. For the midpoint algorithm, the performance is relatively stable over different occlusion levels, but it performs worse than the other two algorithms when the occlusion level is less than 20%. The reason is that the midpoint algorithm, by nature, is robust to the major source of occlusion: the upper/lower eyelid movement, but it is sensitive to the occlusion of cornea reflections and image noises. The symmetric mass center algorithm is robust to most occlusions and the performance remains stable over different degrees of occlusion. In most situations, the symmetric mass center algorithm performs better than the other two algorithms.

Tables 4.3 shows the overall average measurement error for the three algorithms. Compared with the best of the other two algorithms, the symmetric mass center algorithm provides an improvement of about 27% in horizontal measurements and 66.6% in vertical measurements.

Algorithm	Naive mass center	Midpoint	Symmetric mass center
horizontal error	0.3418°	0.3129°	0.2285°
vertical error	1.2720°	0.8946°	0.2989°

**Table 4.3:** Overall performance of different algorithms



**Figure 4.5:** Measurement results for different algorithms when the pupil area is partially occluded: (a) shows the eye images and highlights un-occluded part of the pupil boundary in red; (b) plots the measurement results for different algorithms in terms of percentage of the pupil boundary being occluded.

## 4.4 Summary

While many eye-tracking systems have been developed before, we have developed a new video-based eye tracking system with high temporal resolution using only general-purpose computing and image capturing devices. More specifically, in this PhD dissertation, we have presented a high-speed eye position measurement system that does not sacrifice either accuracy or convenience.

The high-speed 2D eye position measurement system achieves a binocular processing rate of 150 Hz for horizontal and vertical eye measurement, including necessary results and video displaying on the interface. The system employs a dual-mode capturing mechanism so that only the necessary part of the entire image is captured but at the same time maintains the convenience of operation. To calculate the pupil center, we designed a two-step processing algorithm; i.e., first processing a downsampled image to find the approximate location of the pupil and then processing the full resolution image in the small window at the approximate pupil location. By this method, we achieved the high processing rate desired, and at the same time, the new algorithm improves the accuracy in measurement by up to 66.6%.

# Chapter 5

## Torsional Eye Tracking

### 5.1 Existing Torsional Eye Tracking Methods

Most torsional eye position measurement systems use one of two general methods: the tracking of natural or attached landmark [30] on the eye, or some variant of the polar cross correlation methods [18, 31–34]. Applying artificial landmarks to the eye simplifies the computation for torsional measurement. However, the invasive nature of these tracking methods makes them unsuitable for clinical use. The polar cross-correlation methods rely on the natural radial striation pattern of the iris to measure the torsional eye position. Polar cross-correlation methods compare only hundreds of pixels along a circular sampling path (the iral signature) around the pupil center. In contrast to comparing the whole 2-D images with hundreds of thousands of pixels, these methods clearly simplify the computation. The reference signature is the signature sampled from the reference image selected by the operator through the system user interface; the test signature is the signature sampled from each subsequent image by the system. The algorithm calculates the cross-correlation function of the reference signature and the test signature. The shift in the peak of the cross-

correlation function provides a measure of relative change in torsional position.

In the iral signature method for torsional eye tracking, even a small error on the pupil center detection leads to a significant change in the iral signature. To avoid this, Groen presented an algorithm in [35] that takes a wide strip of an annulus instead of an iral signature along a thin line. The method selects and recovers a set of 36 significant patterns in the iris automatically using pattern matching. Each relocated landmark results in a single estimate of the torsion angle. A robust algorithm estimates torsion angle from this total set of individually determined torsion angles, thereby largely correcting for errors that may arise due to misjudgments of the rotation center. This method is only implemented off-line possibly because it does not satisfy the online tracking requirement.

Guillemant et al. presented the neural networks method to detect torsion in [36]. They use a combination of supervised and dynamic learning to identify the pupil, as well as iral patterns that can be used in the measurement of ocular torsion. This method has the advantage that image artifacts like reflections of the illumination lights from the cornea and occlusions of the pupil by dropping eyelids are implicitly dealt with by the algorithm. However, the current version does not take the distortion of the iral image in eccentric eye positions into consideration.

Lee et al. proposed a method for measuring ocular torsion using optical flow [37, 38]. This method measures the displacement between reference and current image using spatial and temporal gradient. Feature points are obtained from the images, and the relative optical flows of each point are calculated. The feature points are selected according to the strength of corner on the signature of iris. Iterative Lukas-Kanade algorithm is then used to calculate the torsion value. This method is fast and can measure torsion value in real time at a speed of 2 *ms* per frame, but it is not robust to pupil center misalignment and eccentric eye position.

Ott and Eckmiller proposed a laser scanning ophthalmoscope that uses prominent structures of the retina to determine ocular torsion [39]. This approach takes advantage of the fact that the structure of retina is much more prominent than that of the iris under IR light. However, it requires that the subject is looking straight ahead, and has the additional drawback that the camera obstructs the field of view. This approach is therefore not generally applicable for clinical usage.

A template-matching technique is presented in [40] to calculate torsional eye position. The algorithm samples two annular segments on the iris on two sides of the pupil center, and uses two threshold values (one high threshold and one low threshold) to remove image pixels with grey values outside of the range. It assumes that torsional rotation of the eye is limited to less than  $2^\circ$  between successive video frames. This algorithm is more robust compared to the cross correlation algorithm and the optical flow based feature tracking systems. However, the method is not robust to occlusions from other artifact, e.g., the dropped eyelid, eyelashes, and shadow. In the experiments conducted we observed that sometimes it is difficult to distinguish the real iris pixel and artifacts, by a simple pixel grey value threshold. In addition, the  $2^\circ$  limit between successive frames can cause the testing result to fall into an erroneous “local minimum”; once the program generates an error value because of eye blinking or occasionally blurred image, it is difficult to get out of that mode.

Object tracking is a classical problem in image vision. Many methods including template tracking, and feature tracking are discussed in the literature [41–45]. A common problem with these approaches is that the complexity of the model results in high computation cost, which limits their use in applications with high-speed requirement. Jurie et al. introduced a template matching algorithm in [46] to handle occlusions and illumination variations.

The robust template matching algorithm we will present to detect torsion movement in Sec-

tion 5.2 is similar to Jurie's work in the sense that we also separate the template into many small segments. The key difference is, instead of calculating the motion of each small template independently and then combining these motion parameters robustly, we eliminate the outliers in these segments in each trial motion parameter and search for the best motion parameter that produces the smallest error value.

## **5.2 Proposed System: The New Torsional Eye Tracking Algorithm**

This section presents an improved template-matching algorithm to reduce the effect of interferences to the iris pattern and to measure the torsion movement in a noisy environment, where the image may undergo occlusion and illumination change. As usual, a circular reference template around the pupil center is extracted at the reference location when the eye is looking straight ahead. Then the reference template is compared with the test template sampled from the subsequent images, when the eye might have moved away from the original reference position, namely, when the eye is in the test position.

Several mechanisms were developed to make the algorithm robust. First, we developed an iris localization algorithm to eliminate artifacts, including occlusions from eyelids, shadows and eyelashes. The iris localization is performed while capturing the reference template and during the calibration procedure, when the speed is not essential. In Zhu's algorithm [40], two partial annuli iris template on two sides of the pupil, each spanning  $90^\circ$  in the reference image, are used. Although the method is effective in excluding the occlusion in a lot of cases, it loses some useful iris information. Because most iris images have radial pattern, it is more important to include a

large angular span for the iris template than increasing the radial width of the template. In the calibration procedure, the torsion movement is also tracked based on the iris localization result. Good frames with little occlusions are collected to compute statistics, e.g., mean and variance of the cost function between the reference and the test template, which will be used later in the online tracking process.

Second, we developed a robust template matching algorithm to calculate the torsion for each frame during online tracking. One novelty of the proposed algorithm is that the reference template is partitioned into many small segments. While evaluating each hypothetical torsion value, each small segment in the reference template is compared with the corresponding area of the test template by calculating a distance function. The cost function for the hypothetical torsion movement is calculated by choosing half of the segments giving the smallest distance value and calculating the distance function between these segments and the corresponding area in the test template.

The intuition behind the algorithm is that if the segment in the reference template or the corresponding area in the test template is contaminated, the distance will be larger. When the evaluated hypothesis is the real torsion movement, if assuming at least half of these segments are not contaminated, then at least half of these segments will give a very low distance function and hence a low cost function for the hypothesis; while for other trial torsion values, taking the same number of segments will result in a larger cost function. If more than half of the segments are contaminated, e.g., during a blink, the image frame is usually ignored by detecting the pupil shape.

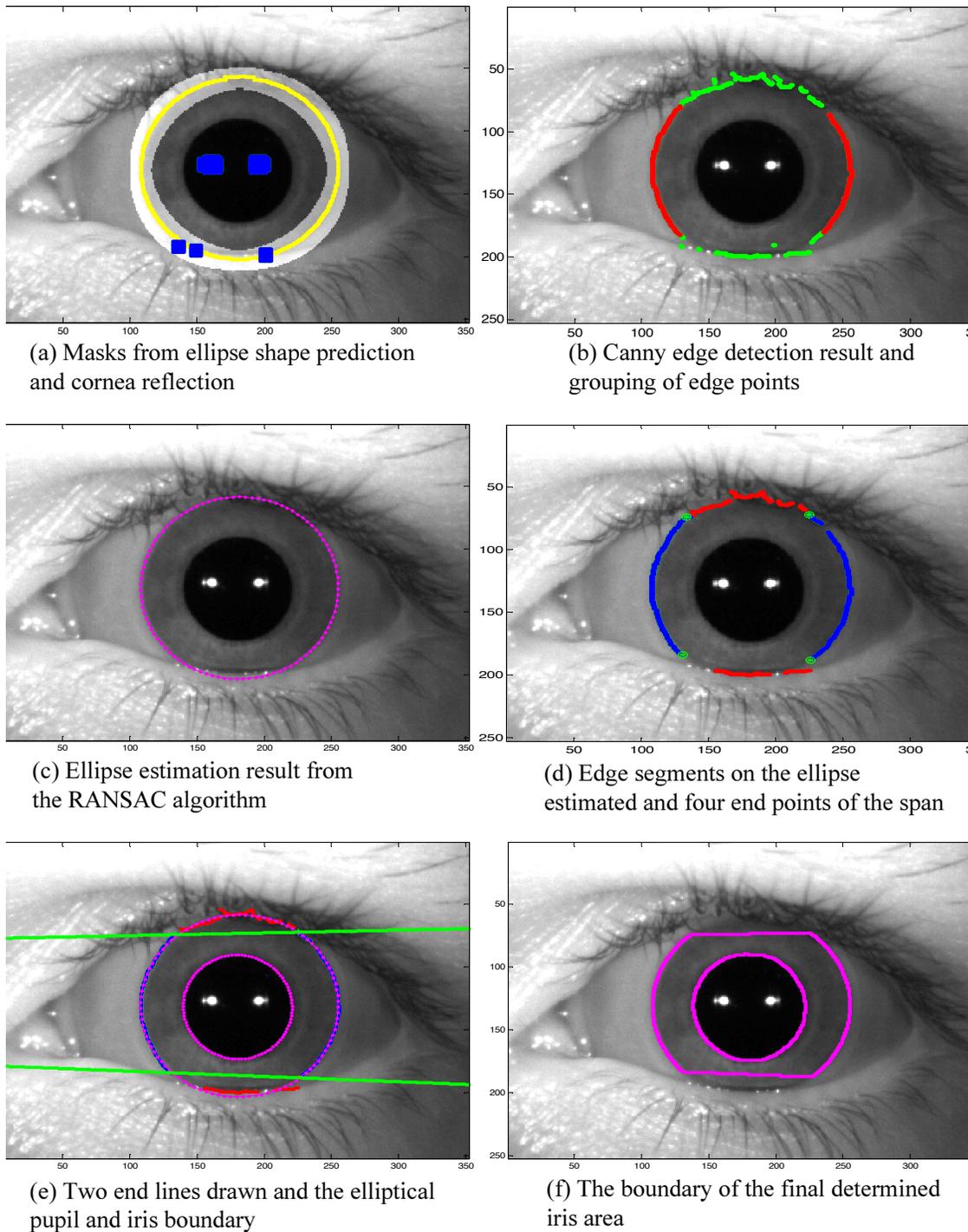
Thirdly, we employed an adaptive searching range in the tracking system. Inter-frame information is previously used in Zhu's algorithm to determine the searching range for torsion calculation [40]; an inter-frame searching range of  $[lastTor - 2^\circ, lastTor + 2^\circ]$  is used, in which  $lastTor$  represents the torsion position of the previous frame. The benefit of this technique is that it reduces

searching range and increases the frame rate, but once the tracking algorithm gives an incorrect torsion value in one image frame, it cannot jump out of it and will result in numerous of incorrect detection results afterwards, and eventually lose tracking. In our proposed system, the searching range is adaptable, i.e., the algorithm will adopt the same small searching range as Zhu's algorithm during stable online tracking while the torsion value of previous image frame is reliable, and use a larger searching range while the torsion value of previous image frame is not so reliable, and consequently a small searching range is highly likely to give an incorrect estimation.

Lastly, to compensate for the geometric distortion, a new calibration algorithm and its corresponding geometric transformation method is used to map pixels in the reference image to the test image according to the homography obtained, as described in Section A.1, by setting the torsion value to be zero.

### 5.2.1 Iris Localization Algorithm

The iris localization algorithm is based on the observation that the elliptical iris/sclera boundary shape is occluded when there are artifacts in the image. Pupil location is always located first and the horizontal and vertical eye position recovered using the method described in Chapter 2 and Chapter 4. Knowing the pupil location, the imaging parameters, and the iris boundary shape in the reference image, the iris boundary shape in the test image can be "predicted". Pixels on the predicted ellipse and its neighboring pixels are set as the mask (*shapeMask*) that will be used for the next step of edge detection. Edge points from the cornea reflection is excluded by a mask (*refltMask*) as well. In the calibration process when shape information is not available, the *shapeMask* is set to all ones. In Figure 5.1-(a), the highlighted grey area is the mask from the predicted ellipse boundary, and the blue area is the mask from cornea reflection to exclude edge



**Figure 5.1:** Intermediate results for iris localization

points. The final mask area is set as  $edgMask = shapeMask \& (!refltMask)$ .

To localize the iris area, we first find the ellipse that matches the iris/sclera boundary. A mod-

ified canny-edge detector is first used in the previous mask area (*edgMask*) to find thinned edge segments. The main modification to the original canny edge detector is that the magnitude value is calculated as the gradient in the radial direction, and only one pixel is found in each radial direction as the strong edge point. Edge points which have the  $x$  gradient magnitude larger than the  $y$  gradient magnitude are put into the group identified by  $idx1$ , which has lower possibility to be caused by eyelids or shadows. All the other points are put into another group identified by  $idx2$ . Figure 5.1-(b) shows the results of edge detection. Red pixels are points that belong to group  $idx1$ , and green pixels are points that belong to group  $idx2$ .

Based on the detected edge points, a RANSAC algorithm is performed to estimate the ellipse. The metric for choosing an estimated ellipse over another is usually the number of edge points on the ellipse. Sampson distance is calculated as the estimation for the distance from a point to the ellipse. Edge points with Sampson distance lower than a threshold (1.5) are considered “on” the ellipse. To avoid including edge points from the eyelids and shadows, instead of simply using the number of edge points “on” the ellipse, we determined a metric that favors the edge points in group  $idx1$  than  $idx2$ . The following cost function is defined and the estimation with a lower value is accepted as a better candidate in the RANSAC procedure.

$$\begin{aligned} costFun \triangleq & 10 * \left[ 2 * EdgPtnNum - 2 * Len(OnElp(idx1)) - Len(OnElp(idx2)) \right] \\ & + 2 * \sum_{idx1} elpDist(OnElp(idx1)) + \sum_{idx2} elpDist(OnElp(idx2)), \end{aligned}$$

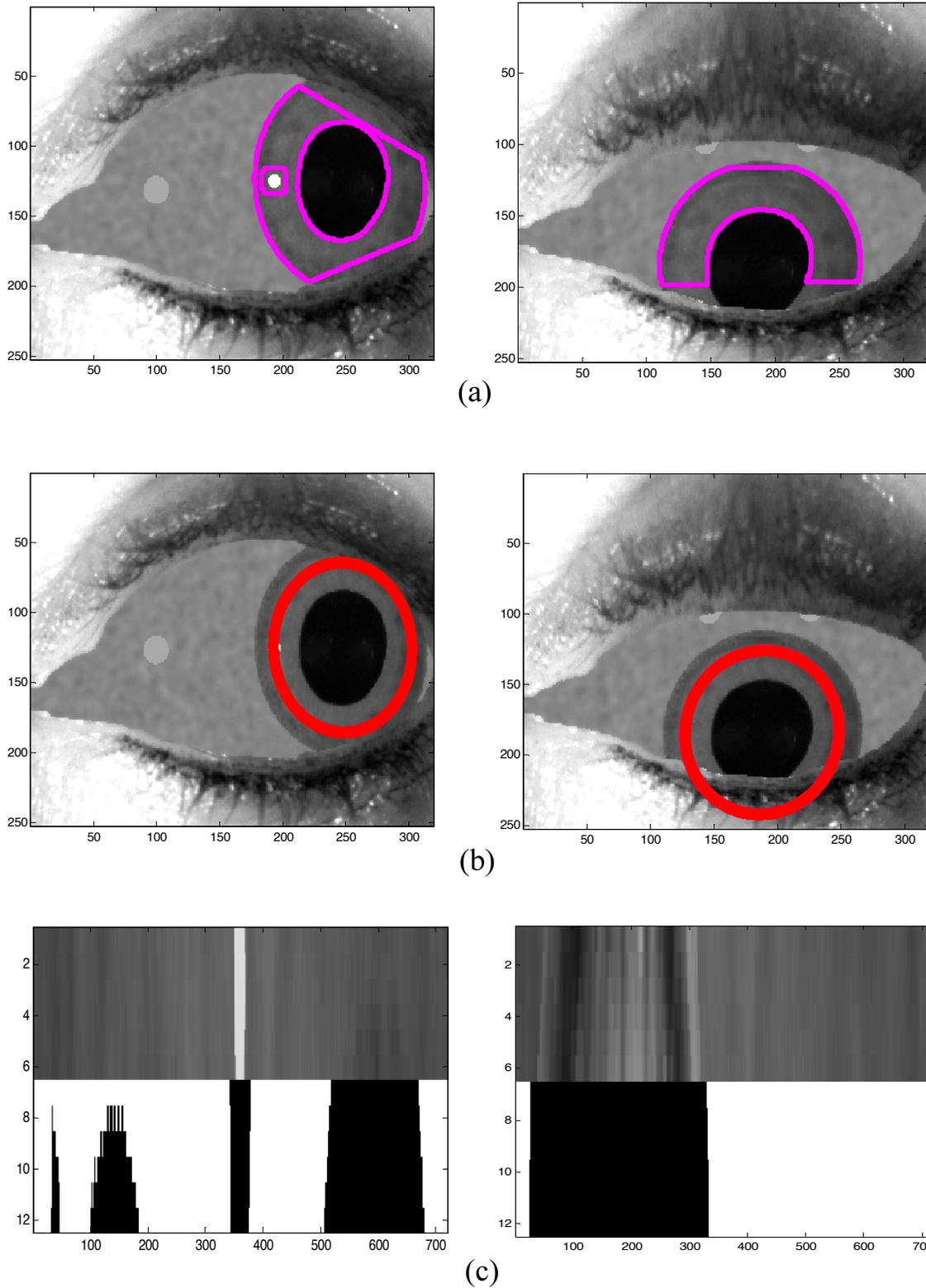
where *EdgPtnNum* is the total number of edge points, *OnElp* is the function to find the subgroup of points on the ellipse, *Len* is a function to calculate the size of the group and *elpDist* is a function to calculate the Sampson distance. Figure 5.1-(c) shows the result of ellipse detected in pink color.

After the elliptical shape of the pupil boundary is determined, the algorithm searches for the upper and lower limit of edge segments that are on the ellipse. Only two types of segments are

included in this search: 1) the segments of which the percentage of points on the ellipse is higher than a certain threshold (0.75); 2) the segments of which the percentage of edge points both on the ellipse and belonging to group *idx1* is higher than another threshold (0.5). Figure 5.1-(d) shows edge segments included in the search in blue color and the four end points in green color.

After that, an upper line is drawn passing through the two upper end edge points, and lower line is drawn passing through the two lower end edge points. The area enclosed by the iris/pupil boundary, iris/sclera boundary, the upper line and the lower line is determined as the iris area. Figure 5.1-(e) shows the two ellipses estimated and the two lines drawn, and Figure 5.1-(f) shows the boundary of the final determined iris area in pink color.

Figure 5.2 shows the iris templates captured in two different eye positions and with different eyelid occlusion levels. The two images in Figure 5.2-(a) show the iris area boundary in pink color. The two images in Figure 5.2-(b) show the multi-pixel-wide iris template in red color. The template includes the same number of pixels in each circumference. This template can be unwrapped and can be shown as a square shape. Figure 5.2-(c) shows the unwrapped iris templates in the upper part, and in the bottom part shows the corresponding masks (*irisMask*) deduced from the cornea reflection and the detected iris area in the original test image. This *irisMask* is used in the template matching algorithm we developed to calculate the torsion.



**Figure 5.2:** (a) Boundary of iris area localized in pink color;  
 (b) The iris template position in red color;  
 (c) Top image: the unwrapped iris template, bottom image: the mask identifying real iris pixels.

## 5.2.2 The Robust Template-Matching Algorithm

The robust template-matching algorithm is described in six steps below. One advantage of the algorithm is that the threshold to exclude “bad” segments is not a hard coded threshold and is adaptable to different environments.

### *The robust template-matching algorithm*

1. Sample signatures around the pupil center from the reference image. Multiple one-pixel-wide signatures on circles with different radius are sampled from the reference image and arrayed together to form a 2-D reference template. Let  $R_{i,j}$  be the normalized gray value of each pixel in the template, in which  $i$  represents the sequence number of signatures in the radial direction and  $j$  represents the sequence number of sampled pixels on a circle. The sampling rate (*samplRate*) is usually 720 pixels to ensure a  $0.5^\circ$  degree resolution. Let  $resl = samplRate/360$ .
2. Define the mask *irisMask* based on iris localization algorithm introduced in Section 5.2.1 to exclude cornea reflections and artifacts in the reference template.
3. Map each pixel on the reference template to the test image using the homography calculated according to Section A.1, and obtain the gray values:  $T_{i,j}$ . Define a mask  $M_{i,j}^t$  based on gray value to exclude cornea reflections in the test image:

$$M_{i,j}^t = imerode(Tm_{i,j}^t), \quad Tm_{i,j}^t = T_{i,j} < Th.$$

4. Divide the reference template into many small segments, each having the same number of neighboring pixels. Let pixels belong to segment  $k$  be represented by  $S_k$ .

5. Shift the reference template within the range of  $SFT \in [s1, s2]$ , search for the value  $SFT_{min}$  that minimizes the cost function  $CFUN(SFT)$ .
6. Set  $\alpha_{min} = \frac{SFT_{min}}{resl}$  as the torsional rotation between the test image and the reference image.

In step 5, the cost function  $CFUN$  is computed as:

$$CFUN(SFT) = \frac{\sum_{S_k^* \in G_k} \left( D(S_k^*, SFT) * Size(S_k^*, SFT) \right)}{\sum_{S_k^* \in G_k} Size(S_k^*, SFT)}, \quad (5.1)$$

where  $D(S_k^*, SFT)$  is the distance function of segment  $S_k^*$ ,  $Size(S_k^*, SFT)$  is the number of “valid” pixels for this segment, and  $G_k$  represents the selected group of segments whose distance function is less than the median value of all segments’ distance function for this hypothetical torsion value:

$$G_k = \left\{ S'_k \mid D(S'_k, SFT) < Median_{\{S_k\}} \left( D(S_k, SFT) \right) \right\}.$$

The distance function , and the number of “valid” pixels of each segment is defined as:

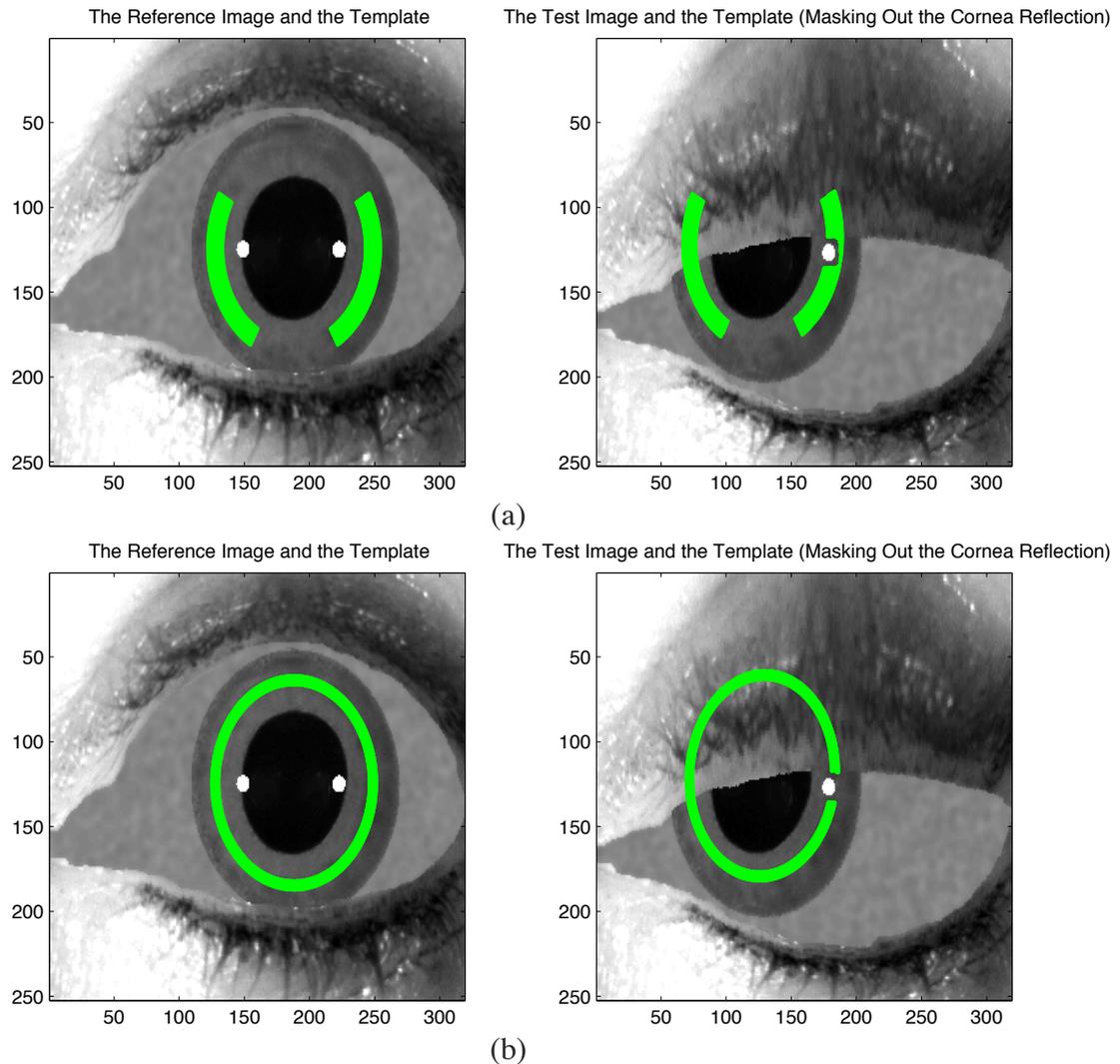
$$D(S_k, SFT) = \frac{\sum_{R_{i,j} \in S_k} \left( \left| (R_{i,j} - T_{i,j+SFT}) \right| * irisMask * M_{i,j+SFT}^t \right)}{Size(S_k, SFT)}, \quad (5.2)$$

$$Size(S_k, SFT) = \sum_{R_{i,j} \in S_k} \left( irisMask * M_{i,j+SFT}^t \right).$$

In step 5, the searching range is determined adaptively. When the tracking starts, or just after the pupil algorithm detects a blink, or when the minimum cost function of the previous frame is beyond 2 standard deviation of the mean of the cost function according to statistics collected in the calibration procedure, the searching range is set to  $[-20^\circ, 20^\circ]$ , otherwise, it is set to  $[lastTor - 2^\circ, lastTor + 2^\circ]$ , in which  $lastTor$  is the torsion position of the last frame. If the minimum cost

function of the current frame calculated utilizing the inter-frame searching range is again beyond the 2 standard deviation of the mean, then step 5 – 6 is repeated with the full searching range.

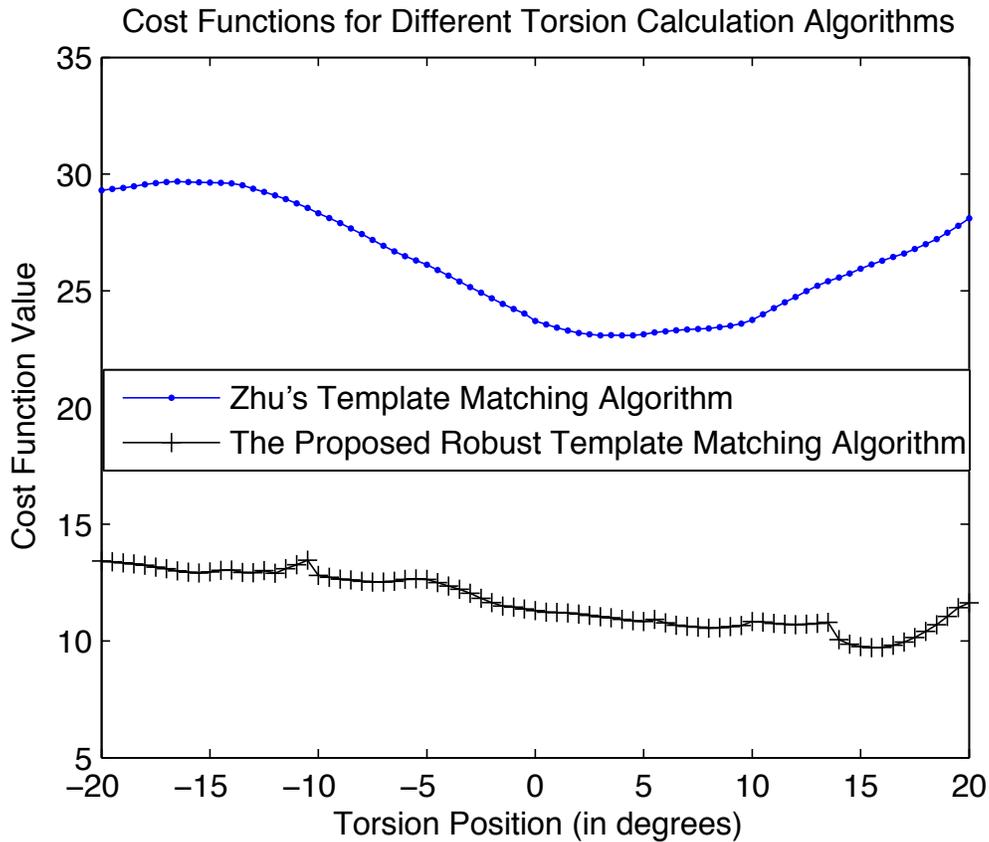
Figure 5.3 shows the reference image, test image, and the template captured under two different schemes. The width of the template in Zhu’s algorithm is two times of that of the proposed method to ensure similar computation cost. The true torsion value is  $15.5^\circ$  for this test image.



**Figure 5.3:** (a) The reference and test template captured for Zhu’s template matching algorithm.

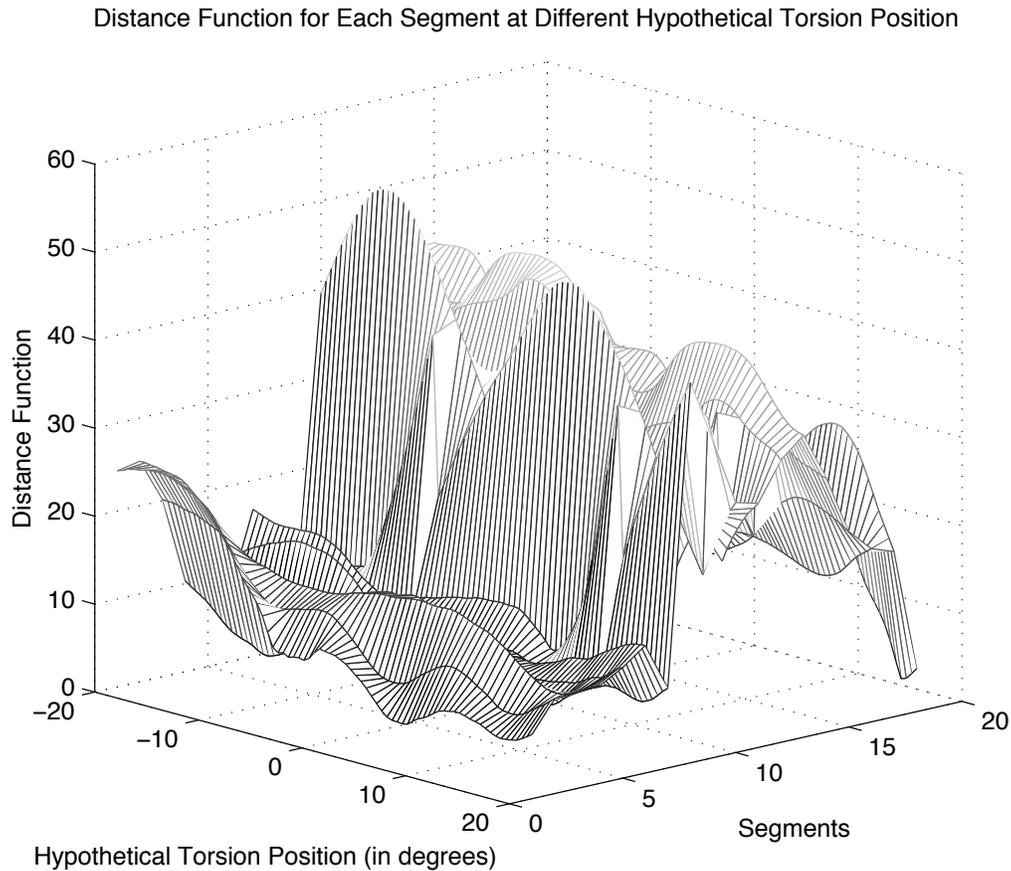
(b) The reference and test template captured for our proposed robust template matching algorithm.

Figure 5.4 shows the cost function for Zhu’s and our algorithm. One can see that for our algorithm, the minimum cost function gives the correct torsion value  $15.5^\circ$ , while the minimum cost function of Zhu’s method gives an incorrect value of  $3^\circ$ .



**Figure 5.4:** Cost function for different hypothetical torsion movement.

Figure 5.5 shows the distance function of each segment for different hypothetical torsion values. One can observe that around the correct torsion value  $15.5^\circ$ , and for segments 1 – 10, i.e., those segments not occluded, the distance function is among the smallest. The proposed algorithm correctly identifies these segments and hence the correct torsional value.



**Figure 5.5:** Distance function of each segment for different hypothetical torsion movement.

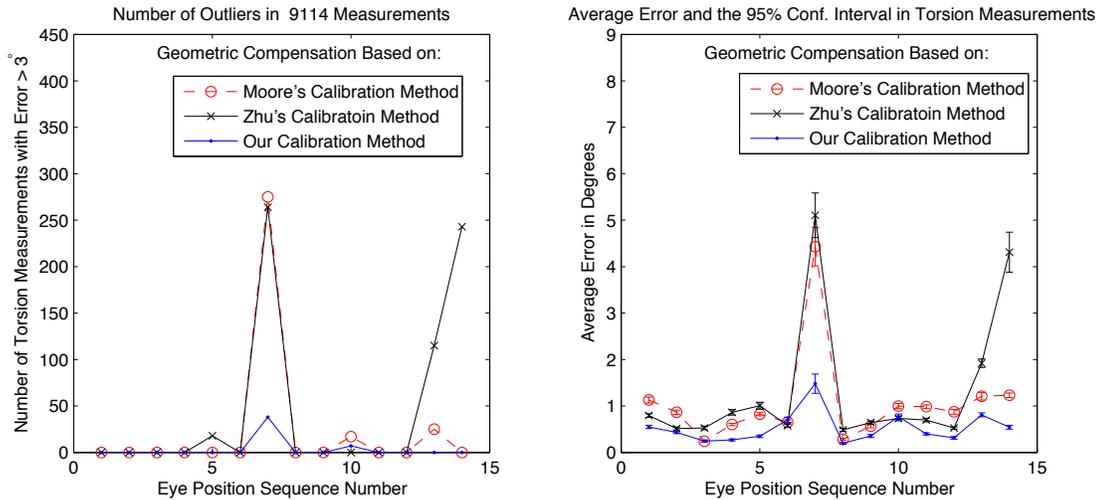
## 5.3 Results

### 5.3.1 Different Geometric Compensation Methods

In this section we discuss the effect of different geometric compensation methods on torsion estimation. Tests are done over synthetic images generated using the simulation system introduced in Chapter 3; for these images the real 3D eye position is known. The test involves 9114 images with little eyelid movement, and with different camera offset parameters: the rotation angle varies from

$[0^\circ, 0^\circ, 0^\circ]$  to  $[10^\circ, 10^\circ, 10^\circ]$  and  $\alpha$  varies from 0.0 to 0.2.

Figure 5.6 shows the torsion testing results with three different geometric compensation methods based on three different calibration algorithms respectively. We use the average discrepancy between the measurement result and the true eye position as the accuracy metric, and use the number of frames that have the discrepancy more than  $3^\circ$  (the outliers) as the robustness metric. The left plot shows the number of outliers for three methods and the right plot shows the average error and the 95% confidence interval. The eye position sequence numbers in Figure 5.6 correspond to the horizontal and vertical eye position sequence numbers in Table 5.1. The large error in position No.7, No.13, No.14 is due to the high level of eyelid occlusion when the eye is in that particular position.



**Figure 5.6:** Torsion testing results using different geometric compensation method

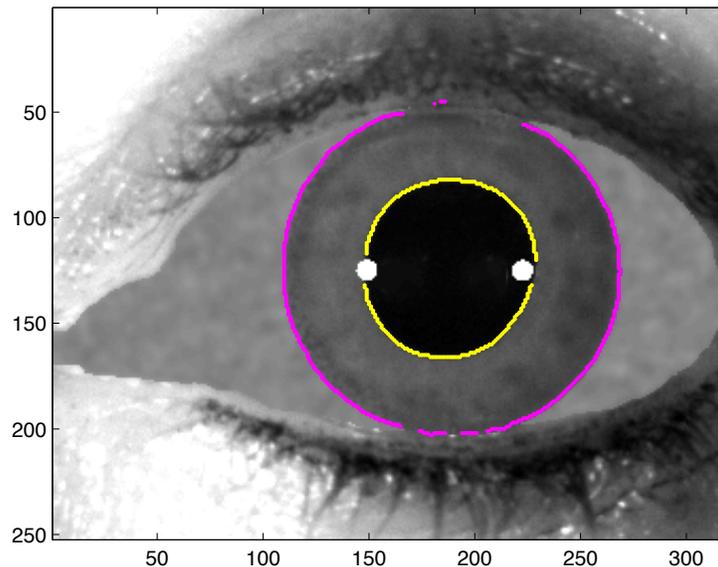
sequence number	1	2	3	4	5	6	7	8	9	10	11	12	13	14
horizontal angle	-20	-15	0	15	20	0	0	0	0	15	-15	15	-30	30
vertical angle	0	0	0	0	0	10	20	-10	-20	10	-10	-10	0	0

**Table 5.1:** Horizontal and vertical eye position for testing images.

One can see that although there are some variations, in most eye positions, the geometric compensation method based on our calibration algorithm outperforms the other two methods. Overall, the mean error in torsion estimation for these three methods are shown in Table 5.2; our method provides a 50% – 60% improvement in the accuracy of torsion measurement, and around 85% – 92% improvement in the robustness metric.

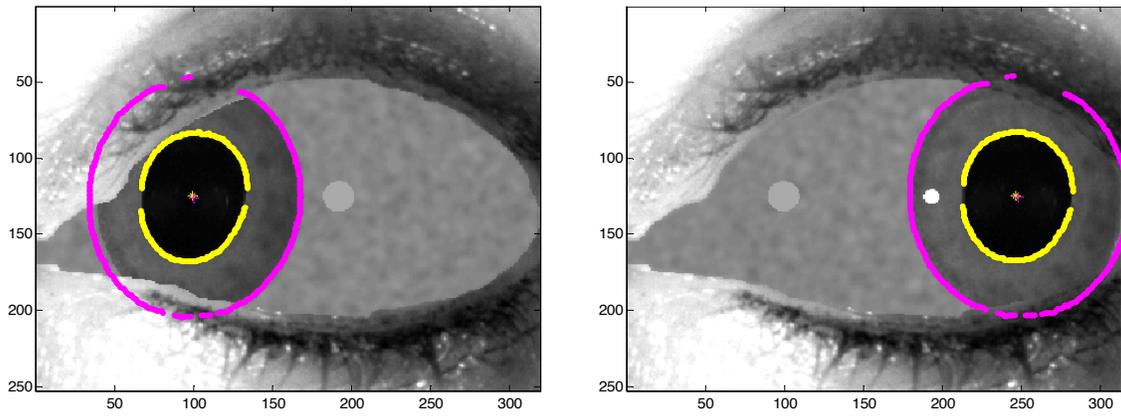
Geometric compensation method	Average error	95% Confidence interval	Number of outliers
based on Moore’s calibration algorithm	1.0634	0.0376	317
based on Zhu’s algorithm	1.3362	0.0560	640
based on our algorithm	0.5297	0.0186	45

**Table 5.2:** Average measurement error, 95% confidence interval and number of outliers for different geometric compensation methods

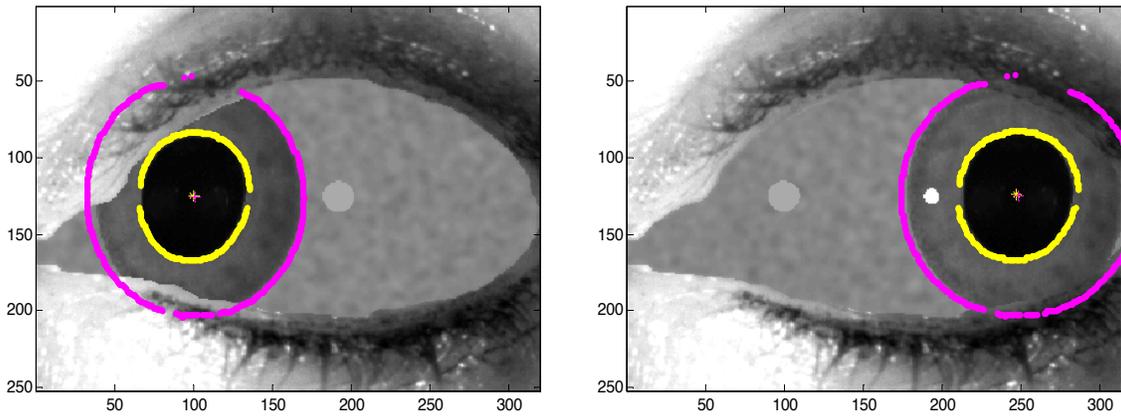


**Figure 5.7:** The reference image and the edge points detected on the pupil and iris boundary

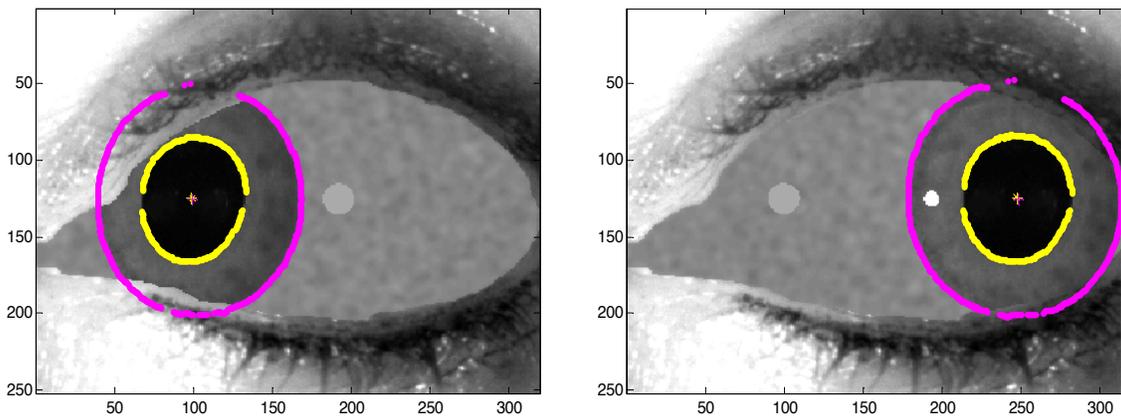
Figures 5.7 and 5.8 illustrate how well different geometric compensation methods map points



(1) Mapping of edge points by Moore's calibration method



(2) Mapping of edge points by Zhu's calibration method



(3) Mapping of edge points by our calibration method

**Figure 5.8:** Mapping of edge points on the pupil and iris boundary in the reference image to the test image by results from three different calibration algorithms.

from the reference image to the test image. Figure 5.7 shows one reference image and the edge points detected on the pupil and iris boundary. These edge points in the reference image are mapped to the test image to compensate for the geometric distortion while capturing the test template. In real test, the pixels actually mapped should be pixels in the iris area. Here the edge points are shown to demonstrate the discrepancy between the mapped pixels and the real edge points in the test image. The position of the mapped edge points are shown in pink and yellow dots in Figure 5.8 on top of the test images; two test images (the left and right columns) with three different compensation methods (in top/middle/bottom rows) are shown. One can see that for our algorithm, the mapped pixels match the edge boundary in the test image very closely, and in the two other algorithms they do not match so well. This fact shows that our new calibration algorithm and the corresponding mapping method provides a better geometric compensation and explains the difference in torsion calculation performance shown in Figure 5.6.

### 5.3.2 Robust Template Matching Algorithm

We tested the torsion algorithm using images with known eye position generated by the simulation system introduced in Chapter 3. Images were generated based on real eye images of four different persons. The horizontal and vertical eye positions are listed in Table 5.1, and the eyelid movement covers the iris template area at different degrees up to 50%. In each horizontal and vertical eye position, 31 images with torsion position in the  $[-15^\circ, 15^\circ]$  range are generated and tested in the order such that the torsion position is linearly increasing. Totally, 36456 images are tested. Figure 5.9 shows the result when the algorithms employ a full searching range in  $[-20^\circ, 20^\circ]$ . The left plot shows the average torsion measurement error, which serves as an accuracy metric for these algorithms. The right plot shows the percentage of outliers, for which the error in torsion measure-

ment is larger than  $3^\circ$ , which can be viewed as the robustness metric of the algorithms. The dotted solid lines show the results of the algorithm that localizes the iris area first, explicitly excluding the occlusions, and then performs the template matching. The solid lines with circles show the results of our proposed robust template matching algorithm. The solid lines with cross show the results of Zhu's template matching algorithm. One can see that our algorithm performs better than Zhu's algorithm both in the average error and the robustness metric; it is very close to the algorithm that localizes the iris first. The template matching algorithm with iris localization performs better than the other two algorithms when the occlusion level is high, but it is slow and designed only to run in the calibration phase to learn the statistics of the iris and the cost function; it can act as an empirical lower bound for the measurement error of different algorithms as well.

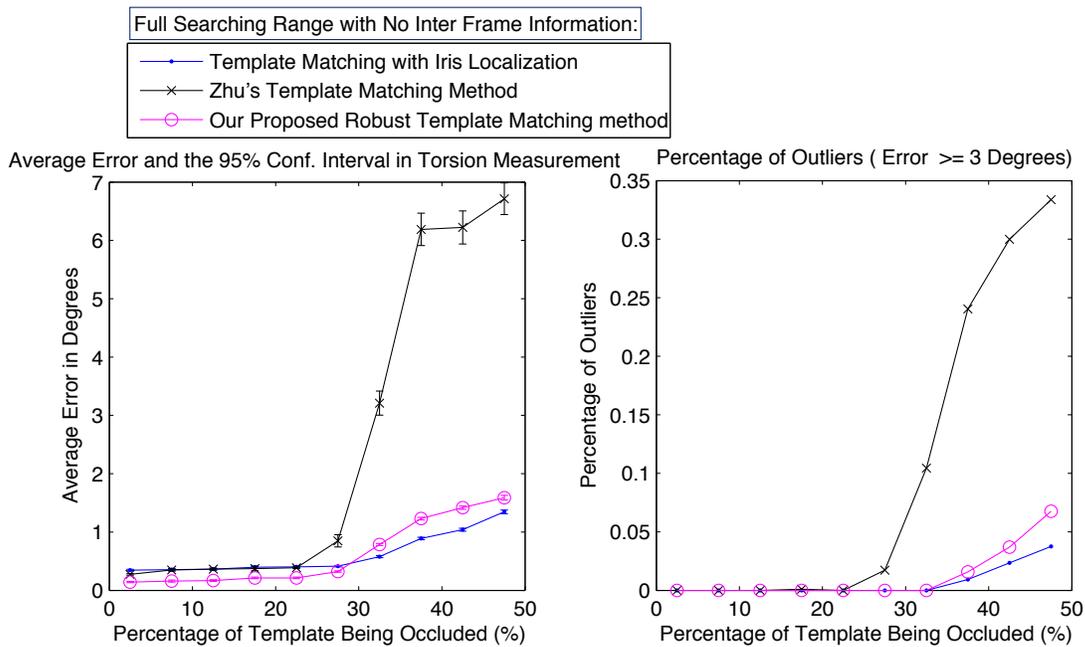
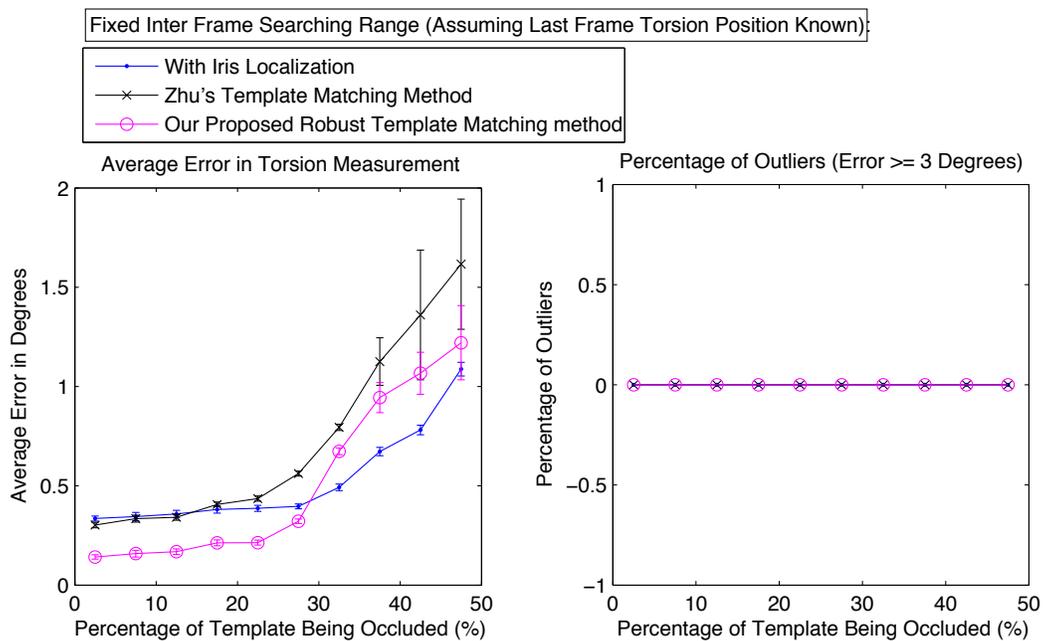


Figure 5.9: Results when the algorithms employ a full searching range in  $[-20^\circ, 20^\circ]$

Figure 5.10 shows the result when the algorithms employ a inter-frame searching range of  $[lastTor - 2^\circ, lastTor + 2^\circ]$ , and assuming the algorithm always knows the true torsion posi-

tion of the last frame. This is not a realistic scenario in practice but it is meant to show how different algorithms perform under this extreme situation. Again, the left plot shows the average torsion measurement error and the right plot shows the percentage of outliers. One can see that there is almost no outliers for all the algorithms, and the accuracy performance of three algorithms are very close to each other. When the occlusion level is low, our algorithm performs slightly better than the other two algorithms, and when the occlusion level is high, the algorithm with iris localization performs slightly better than the other two algorithms. In all cases, our proposed algorithm performs better than Zhu's algorithm.



**Figure 5.10:** Results when the algorithms employ a fixed inter-frame searching range with the last frame torsion position known

In another scenario, we assume that the algorithm knows the true torsion value for the first frame in each eye position, i.e., the first of the 31 frames in the same horizontal and vertical eye position. In Figure 5.11, the three solid lines show the results using fixed small inter-frame searching range;

the dotted, circled, crossed solid lines represent the algorithm with iris localization, the proposed robust template algorithm, and Zhu’s template matching algorithm respectively. The dashed line with triangles shows the result of our proposed robust template matching algorithm with adaptive searching range. One can see that the performance of the proposed algorithm is pretty close to the algorithm that localizes the iris first. The overhead is that for 10% of the frames, it searches the torsion value in the full range, which may slow down the frame rate a little bit, e.g., bring the originally 120 Hz frame rate system to around 100 Hz, but the improvement in accuracy and robustness is significant. Table 5.3 shows the overall average measurement error and percentage of outliers for these four algorithms. Compared with Zhu’s algorithm, the proposed algorithm with adaptive searching range improves about 55.3% in the accuracy metric and about 72.9% in the robustness metric.

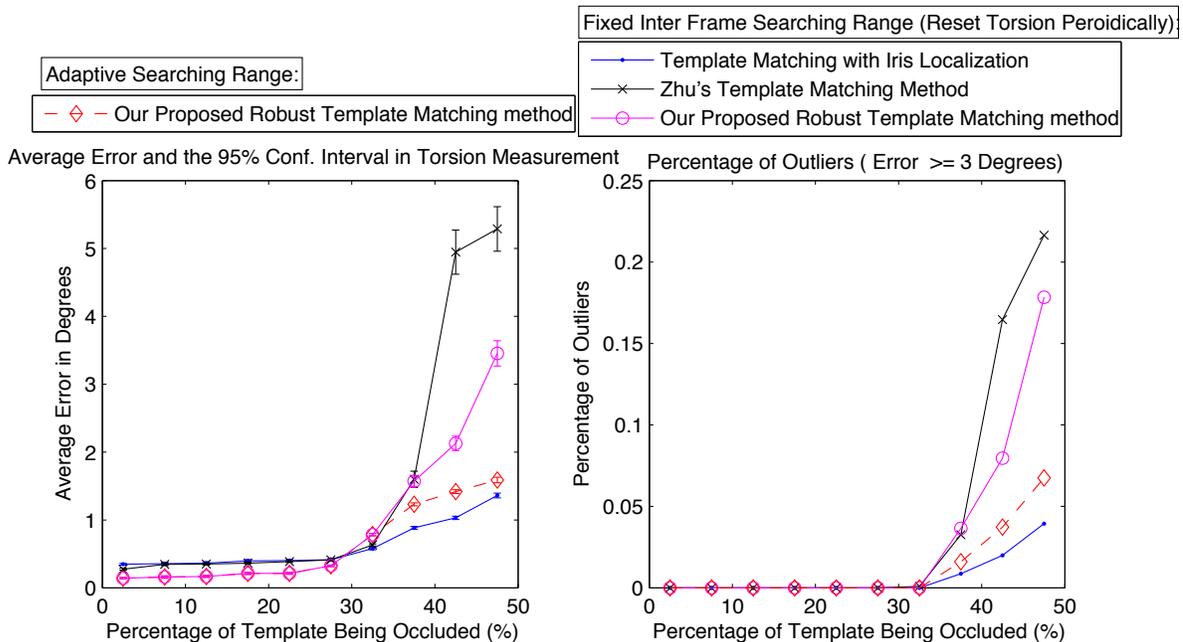


Figure 5.11: Results when the algorithms employ a fixed or adaptive inter-frame searching range

Matching method	Search range	Average error	95% Conf. interval	% of outliers
matching after iris localization	fixed	0.8821	0.0146	0.79%
Zhu’s algorithm	fixed	2.2821	0.0738	5.21%
the proposed algorithm	fixed	1.6508	0.0417	3.40%
the proposed algorithm	adaptive	1.0194	0.0192	1.41%

**Table 5.3:** Overall performance of different algorithms

## 5.4 Summary

To improve the accuracy and robustness of the torsional measurement, we developed an iris localization algorithm that runs during the calibration process, which is essentially the “learning” phase for the later tracking process. In the calibration procedure, not only the standard projection parameters are estimated, but also the reference template is captured after locating the iris, and the statistics of the cost function are collected as well.

We have also developed a robust template-matching algorithm for the tracking phase. By partitioning the template into many small segments and eliminating statistical outliers, we achieved a more accurate and robust system than existing template-matching algorithm [40], which is currently the most robust online algorithm reported. Experiments carried out with a new simulation system show that the accuracy and robustness in measurement improved by about 55.3% and 72.9% respectively compared with the template-matching algorithm, which performs better than the widely used cross-correlation algorithm.

One contribution of our work is that we have designed a simple technique to eliminate outliers

that suits the requirements of our application, which gives a respectable balance between accuracy and speed. Eliminating outliers is a well-known concept in image processing. However, in torsion eye position measurement systems, there are very few existing methods employing this concept. The template-matching algorithm proposed in [40] uses pre-determined thresholds and considers pixels outside the range as outliers. The problem with this method is that single pixel thresholds cannot distinguish outliers whose grey values are similar to the legitimate pixels in the range; but if we consider a group of neighboring pixels, we can distinguish them from the real iris area. That is the reason that our method uses a segment as a unit to identify artifacts.

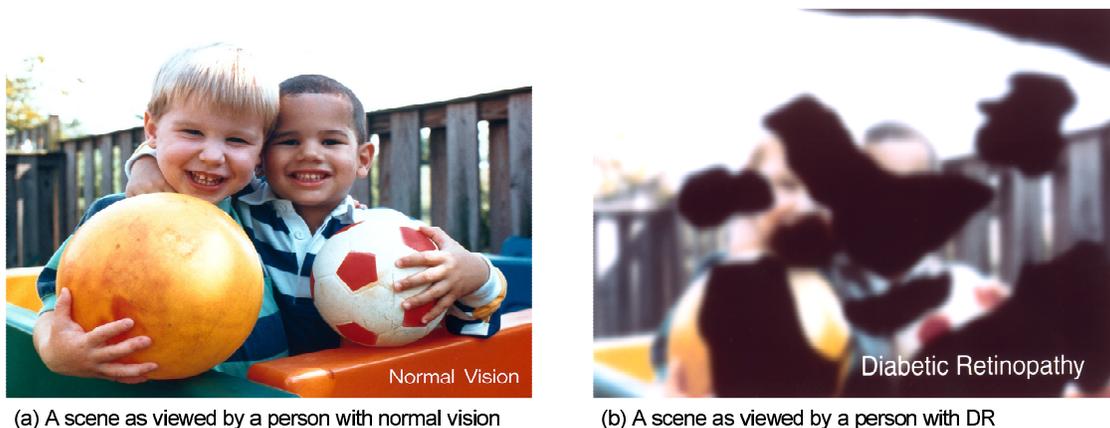
The other contribution of the work is that we developed an adaptive inter-frame searching range mechanism to avoid being trapped in the erroneous “local minimum”. The searching range is large when the cost function is unusually large, which indicates that it is highly possible that the small searching range might have given an incorrect estimation.

# Chapter 6

## Pupil Monitoring and Application in DR

### Screening

#### 6.1 Motivation: Diabetic Retinopathy Diagnosis



**Figure 6.1:** A scene as it might be viewed by a person with normal vision (a) and with DR (b). Both pictures are from <http://www.nei.nih.gov/photo/sims/>.

According to Center for Disease Control and Prevention (CDC), diabetes mellitus (DM) is one of the most prevalent, costly, and preventable chronic diseases in the United States [47]; “the direct

and indirect costs of diabetes is 174 billion a year [47].” DR is an outcome of diabetes and it refers to all the abnormalities of the tiny blood vessels of the retina caused by diabetes, such as blood vessel wall weakening, blood vessel leakage, or abnormal new blood vessels proliferating in and around the retina, etc. DR can cause blurred vision and is believed to be the leading cause of blindness among most adults [48], which poses a major threat to the quality of life. Figure 6.1 shows a scene as it might be viewed by a person with normal vision and by a person with DR.

Some key facts about diabetes are shown in the list below:

- More than 21 million people in the United States have diabetes.
- 6.2 million people with diabetes are unaware they have the disease.
- An estimated 54 million Americans aged 40 to 74 (40.1 percent of the U.S. population in this age group) have prediabetes, a condition that puts them at high risk for developing type 2 diabetes.

According to the study in [49], “Among an estimated 10.2 million US adults 40 years and older known to have DM, the estimated crude prevalence rates for retinopathy and vision-threatening retinopathy were 40.3% and 8.2%, respectively. The estimated US general population prevalence rates for retinopathy and vision-threatening retinopathy were 3.4% (4.1 million persons) and 0.75% (899000 persons)”. Further projections suggest that the number is likely to be tripled in 2050, from 5.5 million in 2005 to 16.0 million for DR, and from 1.2 million in 2005 to 3.4 million for vision-threatening DR (VTDR) [50]. “Diabetes is the leading cause of new cases of blindness among adults aged 20 – 74 years old, and it causes 12,000 to 24,000 new cases of blindness each year. [51]” “Half of the nearly one million patients who developed severe DR went blind within 5 years of diagnosis. [52]”

Early detection and timely treatment of DR can prevent vision loss [53], but about one-third of the diabetic population remains undiagnosed, as mentioned previously. Furthermore, in the early stage of DR, often there are no symptoms, nor does the person experience any pain. Even the advanced proliferative retinopathy can develop without symptoms [53]. “Up to 21% of people with type 2 diabetes have retinopathy when they are first diagnosed with diabetes. [54]” Delays in the diagnosis of diabetes allow diabetic complications to advance greatly before detection, hence increase the risks of the disease and make the treatment much more complicated.

Current methods to detect DR include: eye examination using ophthalmoscopy, nonmydriatic fundus imaging, or a fluorescein angiogram. In the ophthalmoscopy examination, the care professional checks the retina during dilated eye examinations for early signs of the disease, including leaking blood vessels, retinal swelling, damaged nerve tissue, etc [53]. The nonmydriatic fundus imaging needs trained grader to assess disease severity. For the fluorescein angiogram, a special dye called fluorescein is injected into the arm and pictures are taken as the dye circulates through the blood vessels in the retina; the test allows the eye care professional to identify any swelling, leaking or abnormal blood vessels from the photographs and recommend treatment. These methods either depend highly on the experience of the examiner or are invasive.

There is a major need for new methods to detect DR at an early stage, which can facilitate medical intervention before the disease advances to its more debilitating advanced stages.

The objective of this research is to develop an automated, noninvasive tool for widespread screening of DR, to identify those who may require medical attention and/or therapy, and to prevent vision loss. The testing can be either self-administered or performed with minimal supervision in a primary care physicians or eye care providers office.

The vascular damage to the retina caused by DR is not uniformly distributed. In the early stage,

DR only affects the peripheral part of the retina, and the central retina is usually not affected until the more advanced stage of DR. Presumably, for a patient with DR at the early stage, the pupil's response to light stimulus on the center retina would remain normal, and the pupil's response to light stimulus on the peripheral of the retina would be different from a healthy person. This difference can be exploited to design a system used for early detection of DR. The approach of using pupil's response as a bio-sensor for retinal function has been studied and validated by prior research [55–57]. However, its application as a broad screening approach for DR is quite recent [3], and requires both the optimization of the testing system and extensive statistical analysis of the patient data. Compared to existing methods, a digital image based pupillometry and analyzing system has the advantages that it provides an objective and instantaneous test result; it is non-invasive, easy to use, and does not require extensive training for operators. The output would provide a straightforward indication on whether further monitoring or a treatment is needed. The device can also be used to monitor the progressive changes of the disease as required.

## 6.2 Existing Pupil Area Monitoring Methods

Since the system is supposed to be used to perform tests among the general population when a wide range of conditions exist, the system needs to facilitate a rapid test, be robust to interference, and provide accurate information. Most existing eye tracking systems focus on eye position detection instead of the pupil size change. Existing pupil size estimation systems either assume the artifacts, including cornea reflex and eyelid occlusion, are negligible [58], or are not designed to detect the pupil change online and are not fully automatic (depending on the eyelid position) [59]. In [60], the horizontal diameter of the pupil is used to estimate the pupil size; while this seems like a fast and

reasonable approach, the horizontal diameter of the pupil can suffer from artifacts such as cornea reflections.

The pupil shape is close to a circle, and looks like an ellipse in the image when it is in an eccentric position. An ellipse estimation can recover the area of the pupil under occlusions. Ellipse estimation based on edge information is a widely used object extraction technique, but generally faces the dilemma of utilizing Hough transform in a large parameter space which consumes lots of computation power, or using local edge feature, like gradient information, to deduce the parameter space, which suffers from poor consistency and accuracy [61]. In [61], global geometric symmetry of ellipses is used to reduce the dimension of the parameter space. It still assumes that major part of ellipse edge is detected correctly; especially the vertical scan assumes the upper and lower part of the edge is not occluded. Given the fact that the upper eyelid drop is a frequent scenario, the method proposed in [61] is not suitable for our application.

Starburst is an eye tracking system using RANSAC (RANDOM SAMPLE CONSENSUS) algorithm iteratively to fit a subset of detected edge points to an ellipse [28]. Like most algorithms based on RANSAC, the estimation result may take a long time to converge or may not be accurate when a significant portion of edge points are outliers, i.e., when the pupil is partially occluded by dropping eyelids or eyelashes. In [29], a threshold on the curvature of edge points is determined to eliminate outliers before fitting edge points to an ellipse, in which the determination of curvature threshold is not trivial and the heuristics used to select edge segments can sometimes fail. Both algorithms in [28] and [29] are not implemented online.

A widely studied relevant topic is iris segmentation in the context of iris recognition. Most early iris segmentation methods assume circular shape for pupil and iris boundary [62–67]. Recently there are works targeting non-cooperative iris recognition and proposed segmentation algorithms

[68–74] that can handle off-axis eye images and robust to occlusions from reflections, eyelids and eyelashes, etc. Although impressive progress has been made in improving the accuracy and speed of segmentation, to date there are no algorithms that are both robust to image noise and are implemented online at an adequate frame rate that satisfies the requirements of our application.

The pupil tracking method presented in [75] is the most relevant work to our application. Although it is not directly designed to measure the pupil size, estimating the closest matched ellipse for the pupil boundary, plus the camera projection parameters, essentially gives us the pupil area. It is implemented online ( $25\text{ frs/sec}$ ) and addresses the occlusions using a constraint RANSAC (C-RANSAC) algorithm that runs for 260 iterations. It first detects edge points in the image. In each iteration, the algorithm randomly chooses five points to estimate an ellipse, then the ellipse is evaluated based on how many edge pixels are “on” the ellipse (*consensus*). The ellipse with the highest number of consensus is chosen as the best estimate. The evaluation step of the RANSAC algorithm in ellipse estimation is usually computationally expensive since it needs to calculate the distance of each edge point to the ellipse boundary. The algorithm in [75] enhances the speed by discarding the ellipse early when

$$0.7R < \frac{a}{b} < 1.1R,$$

where  $a$  and  $b$  are the major and minor axes of the estimated ellipse, and  $R$  is the rough radius of the reference pupil depending on the camera zooming level. One can see that the shape constraint is fixed for different eye positions although the ellipse shape is obviously related to the eye position. Being the most relevant work in the literature for our application, it still does not satisfy the speed requirement of around  $60\text{ frs/sec}$  for a binocular device, which requires the processing speed of  $120\text{ frs/sec}$ , and sometimes it still produces unsatisfactory results when the occlusion level is significant and the number of outliers are large.

In summary, existing methods either are not implemented online, or do not address the problem of eccentric pupil location and occlusions sufficiently, or do not provide a fast speed and robust detection at the same time. For the screening procedure of DR, we implemented a robust online pupil size monitoring system [76]. The system monitors the pupil size change under stimulation of circular or annular objects of different sizes and brightness produced on a LCD screen facing the subject. Pupil's response is captured by digital cameras and the eye images acquired are processed online. A novel algorithm is developed to calculate the pupil size to provide data input for the screening procedure of DR. To distinguish real pupil size changes from artifacts like blinking, eyelid drop or reflections, we use several image processing techniques to estimate the pupil size when the pupil shape is partially occluded.

To extract the pupil and estimate its size, the algorithm first locates the approximate center of the pupil by a simple threshold and blob analysis operation on a downsampled low pixel-density resolution image in the same way described in [77], then detects edge segments using the canny edge detector and separates edge segments whenever there is a high curvature in the curve, and finally iteratively fits an ellipse to the edge segments based on the shape constraints from known camera projection parameters. The edge detection procedure is applied to a small window of the full pixel-density resolution image; the window encloses the approximate area of the pupil. This ensures both a high speed operation and no loss of edge points. In the iterative ellipse fitting process, the curvature information and pupil shape constraint from known camera projection parameters are used to screen artifacts.

The novelty of the proposed algorithm is that it is the first pupil monitoring algorithm imposing an adaptive shape constraint on the ellipse, which varies with the pupil location. It makes use of the fact that the shape of the pupil boundary in the image is correlated with its position, and both

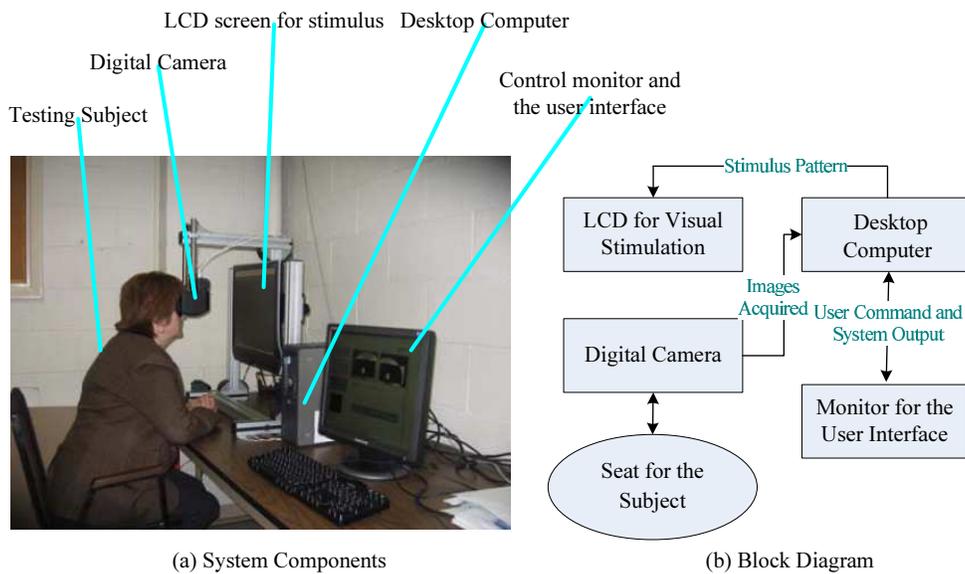
of them are dependent on the 3D eye position and camera projection parameters, which makes it easier to exclude outliers. The algorithm is fast, robust, and does not need to set tricky parameters; it satisfies the online and robustness requirements of the DR application.

## 6.3 Proposed System: Fast Online Pupil Monitoring System

### 6.3.1 System Overview

Figure 6.2-a shows the system components, and Figure 6.2-b shows the block diagram of the system. The subject is instructed to position both eyes in front of the frame, where the cameras are installed. One eye of the subject faces directly the LCD screen and is stimulated by the light from the screen, while the other eye is blocked. The desktop system controls the pattern displayed on the LCD screen to stimulate the eye, another monitor connecting to the desktop system provides the user interface. The images of both eyes are captured by the digital cameras and transferred to the desktop system. We use a Dell desktop system to run our control and analysis software.

Figure 6.3 shows the LCD screen that presents the visual stimulus and the eye image capturing device. Digital cameras are installed on a goggle frame that can exclude ambient illumination. The goggle can be moved up and down to adapt to the height of the subject; it can also be turned around so that the head position can be shifted with respect to the center of light stimulus up to  $20^\circ$  to avoid blocking of the view by the subject's nose and the goggle frame. Infrared off-axis illumination is used for image acquisition to produce a black pupil effect so that it is easy to extract the pupil. The illumination is provided by two micro LEDs operating at 950 nm for each camera, installed inside the goggle frame between the eye and the lens holder. One hot mirror is installed inside the frame for each camera, reflecting infrared light above 800 nm to the camera and allowing visible light to

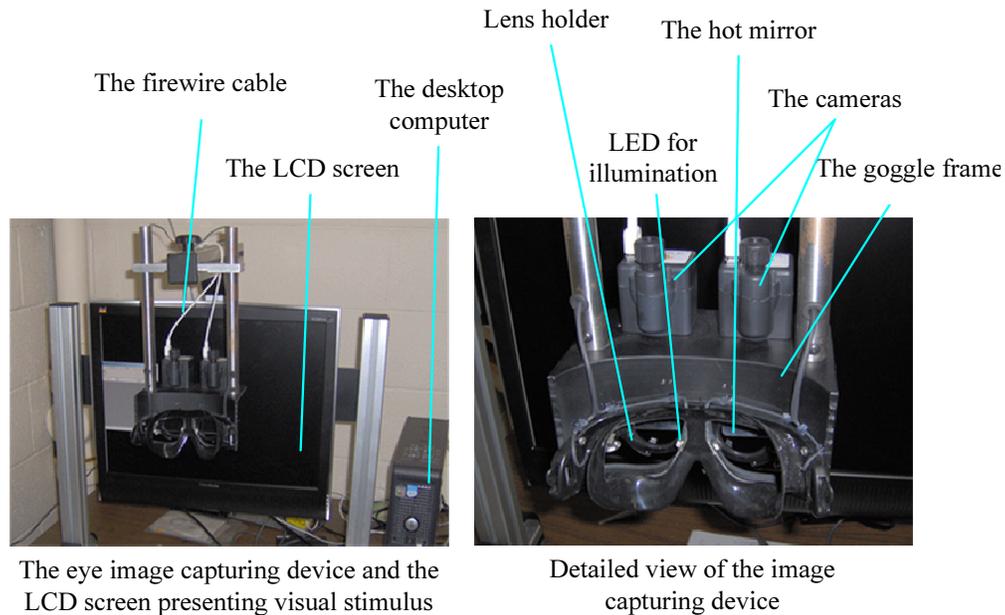


**Figure 6.2:** System components and the block diagram.

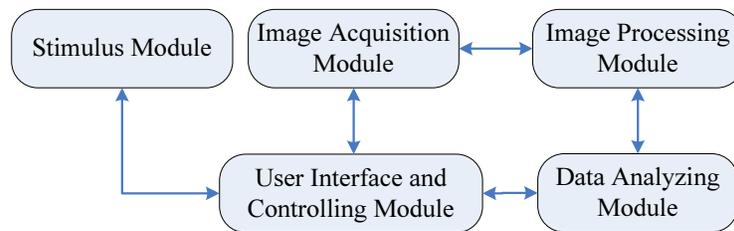
pass through. Trial lenses can be installed for astigmatism correction on the lens holder between the eye and the hot mirror.

The software block diagram is shown in Figure 6.4. The software is developed on the Labview 7.1 platform using its vision module. The stimulus module presents two types of visual stimulus, circular or annular bright objects, on the LCD screen to stimulate the central or the peripheral sections of the retina respectively. At the same time the subject is presented with the stimulus, the image acquisition and processing module captures and processes the eye images online, which contains the eye response information under the specific stimulus. The image acquisition and processing system reaches an online rate of  $80\text{ Hz}$  ( $160\text{ frs/sec}$ ) on a  $3\text{ GHz}$  Pentium IV desktop system. Software synchronization based on PC clock is used to relate the visual stimulus presented to the subject and the eye response. Based on the image processing result, the analysis module analyzes the eye response data collected and provides input for the screening procedure of DR.

For both types of stimulus, according to different test requirements, several parameters can be



**Figure 6.3:** The LCD screen that presents the visual stimulus and the eye image capturing device, including the goggle frame, the cameras connected to the desktop computer by the Firewire cables, the LEDs used for illumination, the hot mirrors reflecting the images to the cameras, and the lens holders that can hold two thin rim trial lenses.



**Figure 6.4:** System components and the software diagram

specified, including the brightness, frequency, on/off time in each period, start/stop time, and the size of the stimulus, i.e., the diameter of the center stimulus, and inner and outer diameters of the peripheral stimulus. A small fixation point is shown on the center to help the subject to focus. Example stimulus patterns for central and peripheral vision are shown in Figure 6.5. During the test, the system records the stimulus strength and the time of PC clock so that the stimulus can be related to the pupil size data acquired online. In the current prototype system, the stimulus strength is measured in relative luminance of the LCD screen in terms of grey value of the white pixels

within the stimulus area. The strength value is assigned between 0 and 100, with 100 corresponding to the maximum output of the LCD screen. In the future, it may be necessary to calibrate the brightness of different monitors.

After the online test, the data analysis module is used to extract important features from the pupil's response traces for diagnosis purpose. Pupil size is currently measured in number of pixels. For the application of DR screening, the types of feature to be extracted are determined from prior medical knowledge, including the pupil size when the eye is in dark environment (the peaks of the pupil size trace), the latency between activation of the visual stimulus and the initial constriction, the minimum pupil size under the stimulus (the valleys of the pupil size trace), amplitude of the pupil's response, i.e., the difference between the peak and the valley, the maximum constriction velocity, and the maximum pupil dilation velocity. The details of the analysis module and the proof of concept test with human subjects for using this system as a screening tool for DR is presented in Section 6.5.



**Figure 6.5:** Examples of stimulus pattern, the circular object for the central vision and the annular object for the peripheral vision.

### 6.3.2 Online Pupil Area Measuring Based on Ellipse Estimation with Shape Constraint

#### The iterative ellipse estimation algorithm

---

**Algorithm 3** The fast pupil monitoring algorithm

Input:  $Im$ : current input image

Output:  $pupil\_area$ ,  $pupil\_location$

---

( $pupil\_area$ ,  $pupil\_location$ ) = **pupil monitoring algorithm**( $Im$ )

$\{ sfim, appxPC \} = \text{threshold\_and\_blob\_analysis}(Im)$ ;

$edge\_map = \text{canny\_edge\_detector}(sfim)$ ;

$edge\_segments = \text{segmentation\_based\_on\_curvature\_threshold}(edge\_map)$ ;

$approximat\_shape\_constraint = \text{get\_constraint}(appxPC, M_r, \alpha)$ ;

$validity\_of\_edge\_points = \text{mark\_validity}(approximate\_shape\_constraint, edge\_map, appxPC)$ ;

$EdgSegQueue = \text{BuitQueue}(edge\_segments, validity\_of\_edge\_points)$ ;

$pupil\_boundary\_ellipse = \text{iterative\_ellipse\_fit}(EdgSegQueue, M_r, \alpha)$ ;

---

The main steps of the pupil monitoring algorithm is listed in Algorithm 3, in which  $M_r, \alpha$  is the camera projection parameters obtained in the calibration process as described in Section 2.3, and  $appxPC, sfim$ , are respectively the approximate pupil center and the full resolution image in the small window around the approximate pupil center obtained through the threshold and blob analysis operation.

After applying canny edge detector to the image  $sfim$ , the edge segments obtained may include outliers, such as points from the cornea reflections, eyelash, eyelid edges, or other sources of noise. We need a robust ellipse estimation that uses only the edge points on the real pupil boundary. To do that, first, the curvature information is used to segment all the edge points into a few segments, each of which has a continuous curvature; second, based on the approximate pupil location obtained in the blob analysis step, the algorithm also labels all edge points coarsely as valid or invalid points for the pupil boundary based on their gradient direction and distance to the approximate pupil center

with a loose threshold; after that, each edge segment obtained in the segmentation step is sorted in a queue by the percentage of valid edge points at a decreasing order; and finally an iterative procedure is employed to select segments and estimate the ellipse. Knowing the camera geometry, the pupil shape in the image is dependent on the pupil location. Therefore, in the iterative step, the *cost function* of the estimated ellipse is evaluated based on how inconsistent the shape and the location of the ellipse center is.

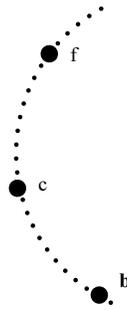
In [29], a curvature information is also used and the selection of curve segments is based on heuristics. The curvature calculated in digital images can be highly noisy, and the heuristics used in [29] sometimes can fail. The constraint on pupil shape is also used in [75], but a fixed threshold on the ratio of the two axis length of the pupil are used; instead the shape constraint in our algorithm is adaptive to the pupil location which provides a tighter constraint.

The segmentation of edge curve is based on curvature estimated from the image. By tracing the edge segment, the gradient angle is calculated and filtered using a Gaussian window of 10 edge points. At each edge point, e.g. the point  $c$  in Figure 6.6, the curvature is calculated according to the following formula:

$$curvt = \frac{\phi_b - \phi_f}{ds}, \quad (6.1)$$

in which  $\phi_f$  is the filtered gradient angle of the previous edge point  $f$  in the segment, and  $\phi_b$  is the filtered gradient angle of the next edge point  $b$ , and  $ds$  is an estimation of the curve length between point  $b$  and  $f$ .

With the curvature information, the algorithm segments connected edges when there is a sudden change in the curvature, which usually is from the eyelid occlusion, or cornea reflections on the pupil boundary. The threshold is chosen by experience and slightly biased toward breaking more segments. The reason is that the purpose in this step is to separate the edge segments from real



**Figure 6.6:** Curvature calculation in point c in an edge segment. It is based on the average gradient angle of point b and f, and curve length between them.

pupil boundary and artifacts. It is acceptable to break the boundary by mistake occasionally since there is an iterative step later on to add them into the group of segments to estimate the final pupil boundary.

The last step of the algorithm is to estimate the ellipse iteratively using the queue of edge segments. First it adds the segment with the maximum percentage of valid edge points to the *chosen group of segments*, calculates a *cost* based on the ellipse center and shape; then it takes the next segment in the queue and adds to chosen group, estimate a new cost. The new segment is accepted to the *chosen group* if the new cost is lower than the previous one; otherwise, the edge segment is discarded. In the next iteration, another segment is taken from the queue and the previous steps are repeated until the queue is empty. Finally, an ellipse is estimated from all the *chosen segments*. The cost function is described in detail in the next section. The pseudo code of the iterative procedure is shown in Algorithm 4.

### **The cost function for the ellipse estimation**

The cost function of the estimated ellipse is calculated from the projection parameters obtained in the calibration procedure. The circular pupil appears as an ellipse when projected onto an image.

---

**Algorithm 4** The iterative ellipse estimation algorithm

Input: *EdgSegQueue*: the queue of edge segments with decreasing percentage of valid edge points

Output: best\_ellipse

---

best\_ellipse = **iterative ellipse estimation**(*EdgSegQueue*,  $M_r$ ,  $\alpha$ )

```

Sc =  $\Phi$ ;           chosen_curves
best_ellipse =  $\Phi$ ;  best ellipse so far
gfun = 1000;
while isNotEmpty (EdgSegQueue) do
  Cs = head_of_queue (EdgSegQueue);
  Sc = Sc + Cs;
  new_ellipse = ellipse_estimation (Sc);  Estimate an ellipse with all edge points in Sc.
  ngfun = cost_function (new_ellipse,  $M_r$ ,  $\alpha$ );
  if ngfun  $\leq$  gfun then
    gfun = ngfun;
    best_ellipse = new_ellipse;
  else
    Sc = Sc - Cs;
  end if
end while

```

---

Intuitively, the shape of the pupil boundary depends on the eye position. We can characterize this intuition in a mathematical form. Actually, the  $3 \times 3$  symmetric matrix  $C_1$  of the ellipse equation for the pupil boundary in the image when the eye is at a particular position and the  $3 \times 3$  symmetric matrix  $C^e$  of the ellipse equation for the pupil boundary in the eye coordinate system satisfies:

$$C^e \simeq H_1^T \times C_1 \times H_1, \quad (6.2)$$

in which,  $H_1$  is a  $3 \times 3$  matrix characterizing the homography between the image and the pupil plane in the eye frame. If the camera projection parameters are known,  $H_1$  can be determined from the current eye position (which can be deduced from  $C_1$  following the method in Section 2.3.2). The details of derivation is given in Chapter A. The implication of (6.2) is that if the matrix  $C_1(i)$  for an ellipse estimation from the image is true, and we derive the  $C^e(i)$  according to (6.2),  $C^e(i)$ , it should fit the real pupil boundary in the eye coordinate system, which means it is a circle and its center is at the origin.

From  $C^e(i)$ , one can deduce the ellipse parameters:

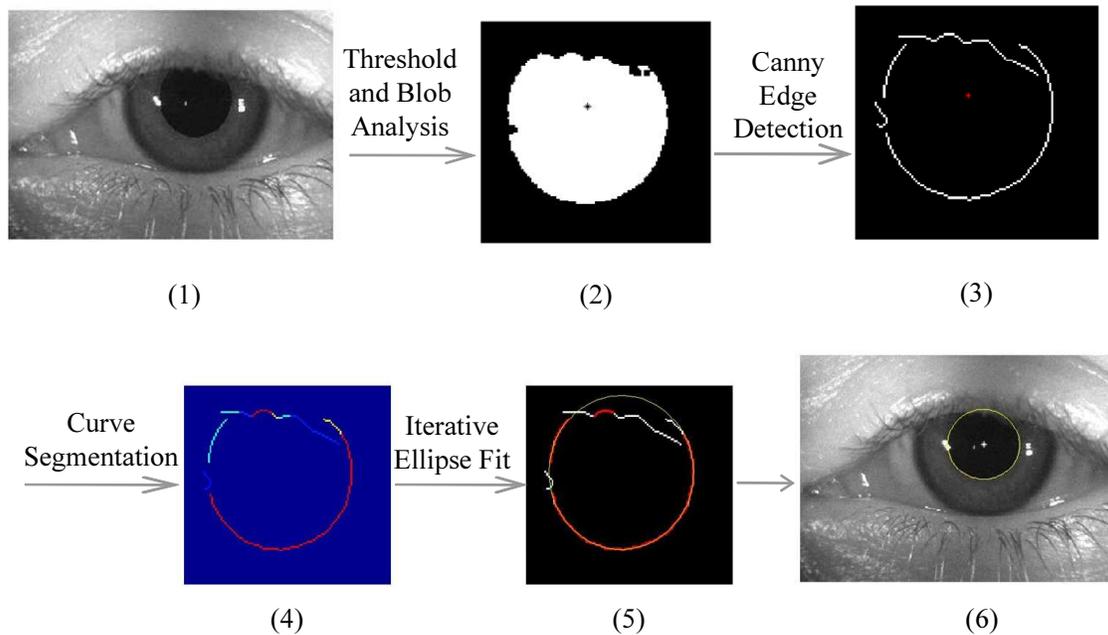
Ellipse center	Axis length	Rotation angle
$x_c, y_c$	$a, b$	$\xi$

**Table 6.1:** Ellipse parameters

It follows naturally to define the cost function as:

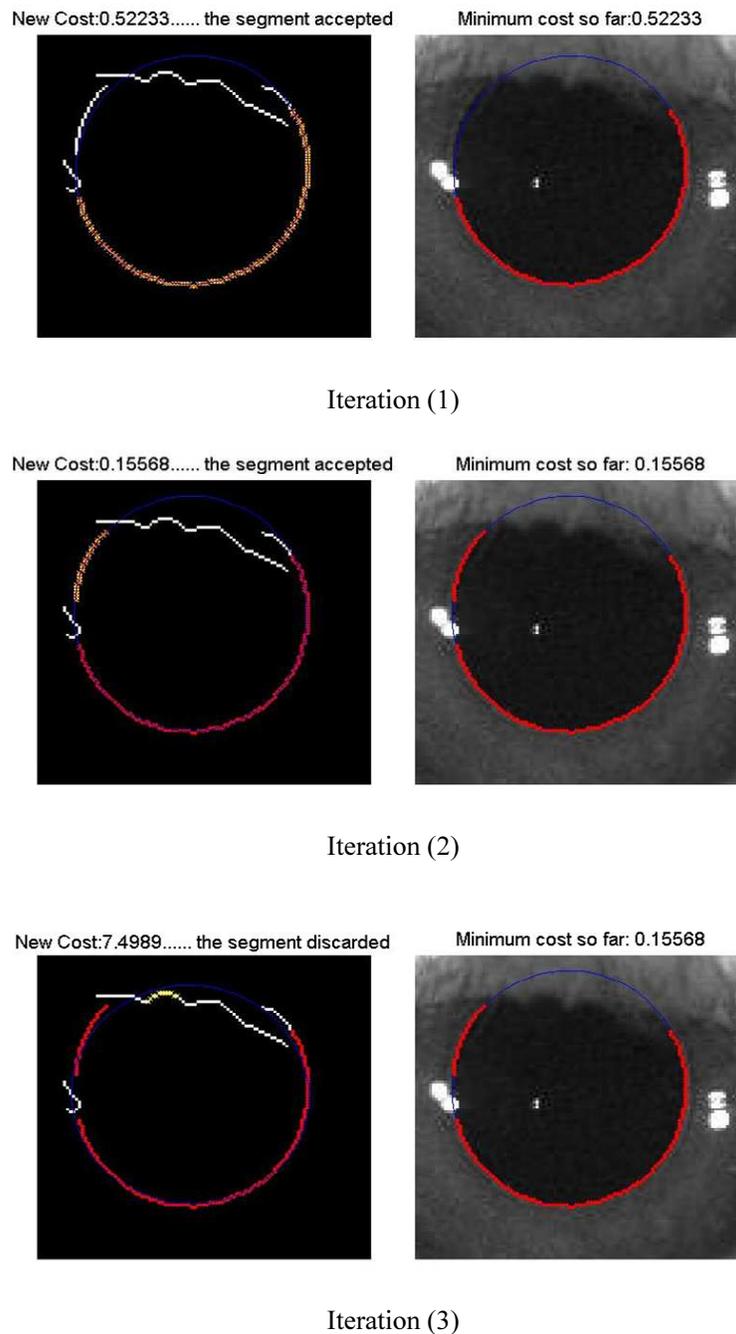
$$f_g = x_c^2 + y_c^2 + (a/b - 1)^2; \quad (6.3)$$

## 6.4 Results



**Figure 6.7:** Flow of the algorithm and intermediate results:

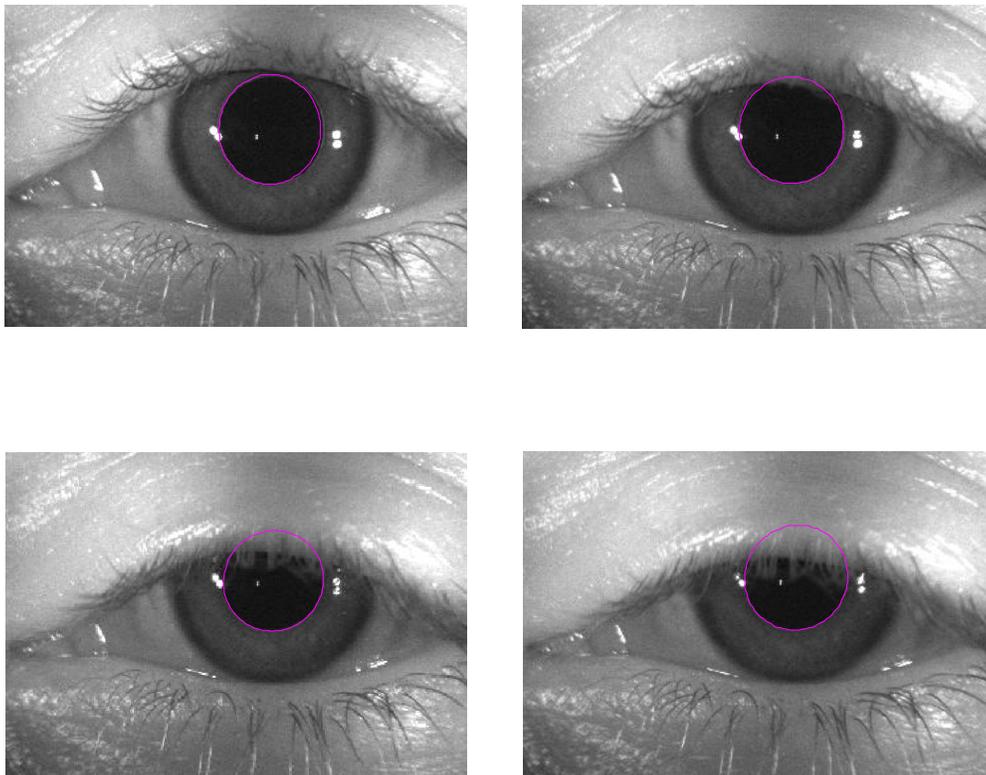
- (a) The input grey-level image.
- (b) The result of threshold and blob analysis.
- (c) Output of the Canny edge detector.
- (d) The result of curve segmentation based on discontinuity of curvature.
- (e) The result of the iterative ellipse fitting process.
- (f) The final ellipse estimated shown in the input image.



**Figure 6.8:** Three iterations in the iterative ellipse fitting procedure. Red segments are those already chosen from previous iterations; the yellow segment is the one currently being processed. In the title of the left column, the new cost when the current segment is added is shown, and the decision whether to accept the segment or to discard is displayed as well. In the right column, the blue curve shows the ellipse estimated from currently chosen segments.

Figure 6.7 shows the flow of the algorithm and intermediate results. In image (2), the binary image of full resolution in a small window enclosing the pupil and the approximate pupil center is

shown. In image (4), different edge segments are shown in different colors. In image (5), points in red color represent the final edge points used to estimate the ellipse, and the yellow curve is the best ellipse estimation obtained. Figure 6.8 shows three iterations in the ellipse fitting procedure. In the first row, the segment is accepted since it is the first segment processed; in the second row, the new segment is chosen as well, since after adding the new segment to the chosen group, the new cost is lower than the previous cost; in the third row, the new segment is discarded since the new cost is larger than the previous cost. In Figure 6.9, four different input images with different levels of occlusion are shown; for each image, the final estimated ellipse for the pupil boundary is plotted in pink.



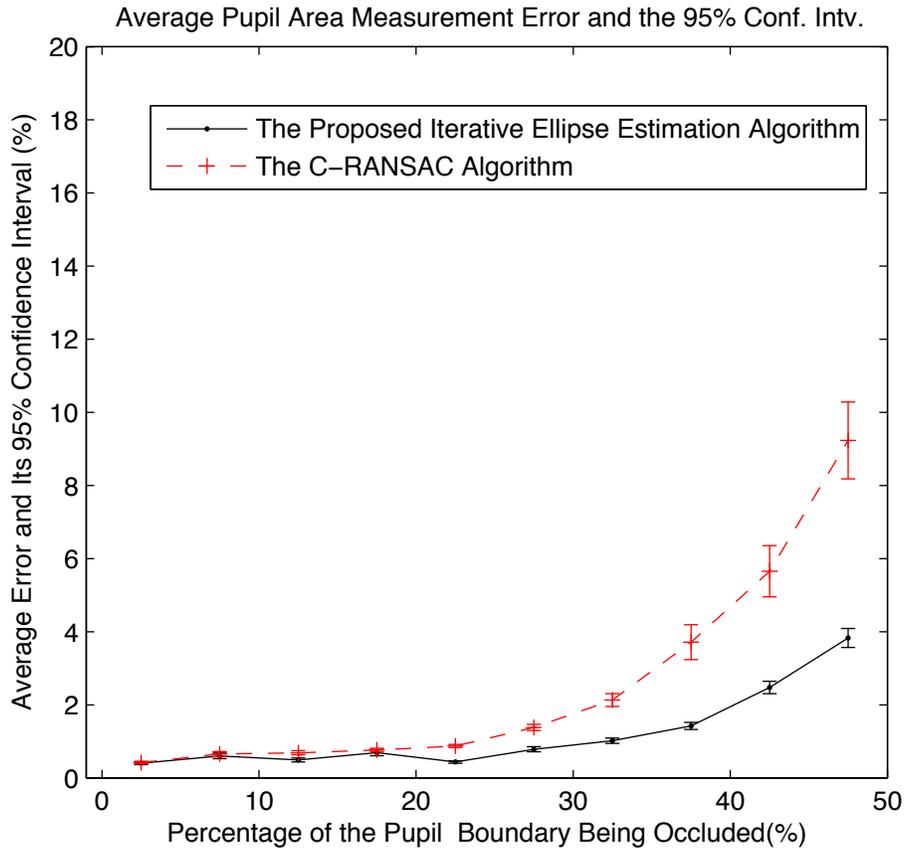
**Figure 6.9:** Eye images and the estimated elliptical pupil boundary.

We also tested the algorithm using the simulation system we developed as described in Chapter 3. The simulation system generates images at 15 different eye positions as listed in Table 6.2. In

each eye position, the eyelids/eyelashes are moved up and down to synthesize images of different levels of occlusion. We measure the occlusion by the percentage of pupil boundary being occluded. For each eye position and each occlusion level, 10 images are generated by adding Gaussian noise with zero mean and standard deviation of 1.7 to the image. Overall, 5600 images are synthesized and tested. Figure 6.10 shows the results we get. Each data point in the plot shows the average measurement error for images, whose occlusion level falls in the bin between  $(x - 5\%, x + 5\%]$ , in which  $x$  is the horizontal coordinate (occlusion level) where the data point is plotted. The occlusion level is defined as the percentage of pupil boundary being occluded by eyelids, eyelashes or reflections. The 95% confidence interval for each bin of images are plotted as well. The red dashed line shows the results with the C-RANSAC algorithm proposed in [75] and the black solid line shows the results using our proposed method. One can see that at lower occlusion levels ( $< 20\%$ ), the performance of both algorithm is pretty close although our algorithm is slightly better. At higher levels of occlusion, our algorithm outperforms the C-RANSAC algorithm.

$\theta$	-20	-15	0	15	20	0	0	0	0	-15	15	-15	15	-30	30
$\phi$	0	0	0	0	0	10	20	-10	-20	10	10	-10	-10	0	0
$\psi$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 6.2:** The 3D eye positions for which images are synthesized using the simulation system.



**Figure 6.10:** The average measurement error and the 95% confidence interval for each bin of images with the occlusion level in  $(x - 5\%, x + 5\%]$ .

## 6.5 Application in DR Screening and Proof of Concept Test with Human Subjects

It is well known that the pupil size and response can be affected by many factors including age, iris color, media opacities, and medication or even within the same person at different anxiety levels. It is difficult to collect enough statistics to address all of these aspects. To control these variations, we use the central response of the pupil within the same eye as an internal reference and compare

the peripheral response with it. The overall light sensitivity of the central part of retina is usually stable until later stages of DR. The idea of using this intra ocular comparison method for DR screening was first proposed by Gorin in [3] and the objective is to use the ratio of stimulus levels when central and peripheral responses reach the same amplitude as the major criterion. We used a slightly different ratio metric in the proof of concept test here, so that we do not need to perform so many tests to find the specific stimulus levels when central and peripheral responses reach the same amplitude. Basically, for each feature, we calculate the ratio of:

$$\frac{P_p - P_c}{P_c}, \quad (6.4)$$

where  $P_p$  is the parameter for the peripheral response and  $P_c$  is the parameter for the central response at the same stimulus level; the parameter may be the amplitude, velocity, or response time determined from the pupil trace. The parameter for the central response is considered the reference value since by our hypothesis, at the early stage of DR, the central vision is to be preserved and peripheral retinal sensitivity would reflect the changes from DR.

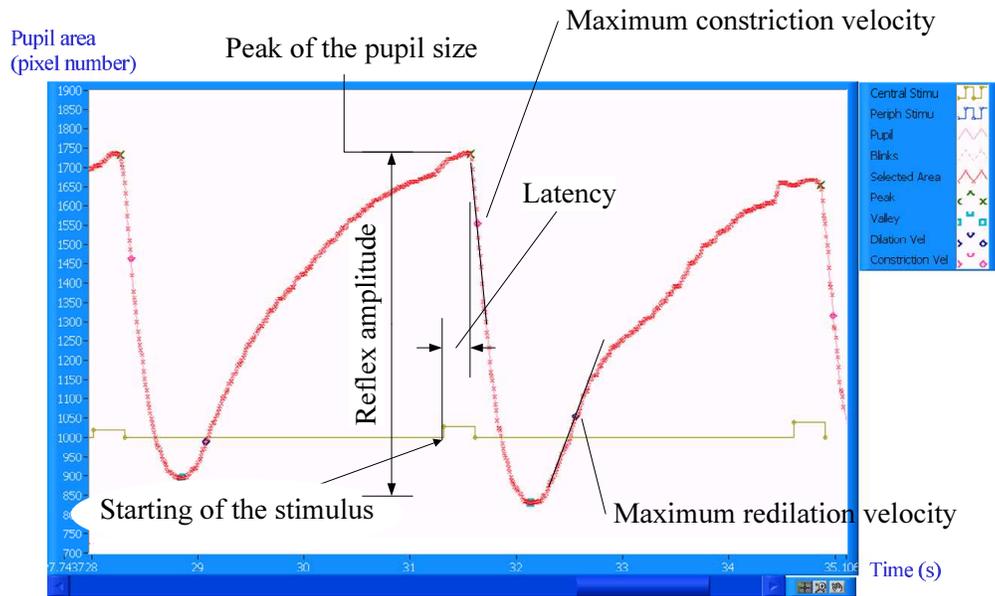
### **6.5.1 Analysis of the Pupil Response**

The output of the online pupil monitoring process is sequences of pupil size and the corresponding stimulus type and strength, along with the PC clock timing. The analysis module processes the traces obtained and extract important features. The first step of the analysis is denoising. Although the image processing module can tolerate a certain degree of interferences, there are still artifacts remaining in the traces; one important artifact is introduced by blinking of the subject. We implemented a blink removal procedure based on the amplitude, the velocity (calculated from the first

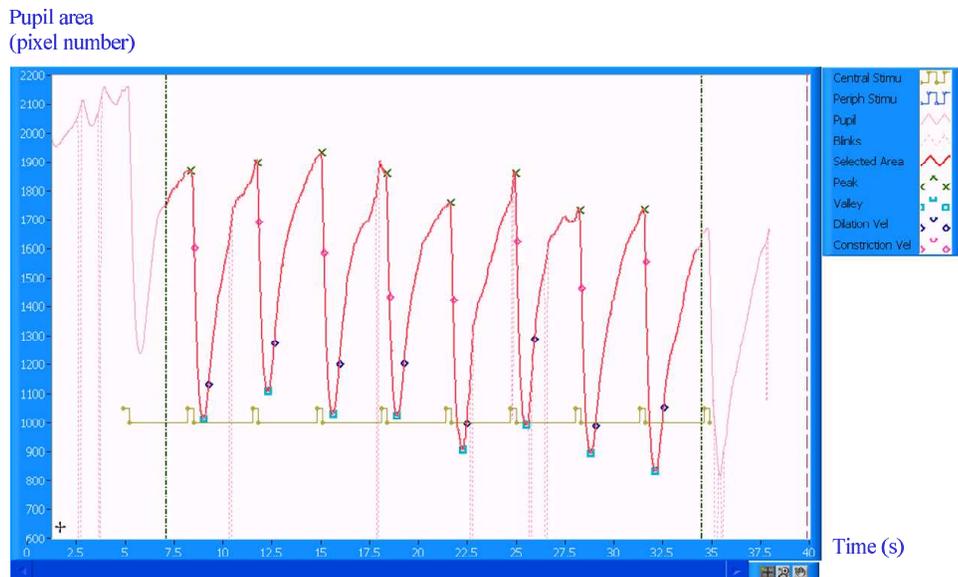
derivative of the trace), and the width of the negative pikes in the trace. Figure 6.11 shows the interface and examples of traces with blinks to be removed marked in red dots.



**Figure 6.11:** Tools to remove blinks. The black plots are the eye response traces, and the red dots represent the blinks detected.



(a) Parameters to be measured from the pupil size trace.



(b) A typical pupil size trace.

**Figure 6.12:** In the pupil size trace plot, the solid red plot is the chosen part of the eye response trace for analysis, the dashed red part is the blink removed previously, and the yellow plot represents the on/off time of the central stimulus.

After removal of the blinks, features including the peaks, valleys, latency, reflex amplitude, maximum constriction velocity, and maximum dilation velocity are extracted from the traces. Figure 6.12-a illustrates the parameters in the trace, and Figure 6.12-b shows a typical pupil's response

to a light stimulus. In each cycle of the stimulus, the point when the pupil starts constriction is detected between the instance that the stimulus is on and the point where the maximum constriction velocity is achieved, at the point when the first derivative of the pupil size trace reaches the maximum negative value. The peak of the pupil size is measured at the point when the pupil starts constriction, and the latency measures the time between the stimulus is on and the point when the pupil starts constriction. The valley is found when the pupil size reaches the minimum value in the cycle. The maximum constriction velocity and the maximum dilation velocity are calculated respectively from the maximum negative first derivative and the maximum positive first derivative of the trace.

### **6.5.2 Proof of Concept Test with Human Subjects**

A proof of concept test is performed with the cooperation of two entities:

1. UCLA, Jules Stein Eye, Department of ophthalmology by Dr. Michael Gorin,
2. Neuro Kinetics, Inc., Pittsburgh, PA by Dr. Alexander Kiderman.

Two identical systems were built and installed at these two locations. Subjects who were supposed to suffer from retinopathy were identified from the clinical practices of the retina faculty at the Jules Stein Eye Institute by Dr. M. Gorin. Normal subjects in the control group were selected and tested by Dr. Alex Kiderman at Neuro-Kinetics, in Pittsburgh, under the direction of Dr. M. Gorin [78]. The nature and the purpose of the study were explained to the subjects, and the informed consent of the patients was obtained from them as well. The main objective of the study is to verify that the system provides an immediate indication of midperipheral ischemia, which is the earliest sign of DR progression.

In the preliminary phase of the test, it is found that three parameters, the constriction velocity, the latency, and the amplitude have the highest repeatability for the same subject in different tests. Hence, these three features were chosen as the primary outcome measures in the study. A total of 32 subjects were tested; for 2 of them, there is not sufficient clinical information about the severity of the DR available. Therefore, 30 subjects were included in the final data analysis. These participants include normal subjects and subjects with full range of severity of DR. Table 6.3 lists the distribution of the subjects [78]. Certain criteria is set up to eliminate other causes of retina change, for example, subjects were chosen so that they have no history of prior ocular trauma or surgery in the eyes; normal subjects must have a history of normal eye examination in the year prior to participation in the study; patients with history of ocular disease, diabetes, and especially glaucoma and high myopia are excluded; ages in the normal group are chosen to closely match the diabetic patients, etc. [78].

To eliminate accommodation of the eye, proper refractive correction lenses are inserted in the lens holder to account for the distance to the LCD screen. Before the test, the subject is dark-adapted for at least 5 minutes until the dilated pupil size is stable. Then tests are performed by exposing the eye to center and peripheral stimulus with varying on/off duration, and with increasing intensity until the brightness level is one step above the one that elicits the maximum pupil's response. Each test lasts no more than 2 minutes and subsequent tests are separated by a 1 – 2 minutes break. Each session is no more than 45 minutes, and each subject is supposed to take 3 sessions on different days.

Figure 6.13<sup>1</sup> shows the average response for one subject. Each square dot represents the average of the parameter value of the 20 repeated tests for one brightness level. The solid line plots are

---

<sup>1</sup>Plots are extracted from the report for the NIH Grant in [78].

Medical info.	Normal	With Diabetics, no DR	Mild DR	Moderate DR	Severe DR
number of subjects	14	7	4	3	2

**Table 6.3:** Distribution of the subjects participated in the test and included in the final data analysis

the second order polynomial fit of the average parameter values obtained at different brightness level for each type of test. The formulas from the polynomial curve fit is used to calculate the normalized parameter value at any specific brightness level according to (6.4).

Color of the bar	Prior knowledge of the subject with regard to clinical retinopathy
blue	normal
green	diabetics with no or minimal clinical retinopathy
yellow	mild DR
purple and red	moderate and severe retinopathy

**Table 6.4:** Color of the data bar and its represented prior knowledge of the subject with regard to clinical retinopathy.

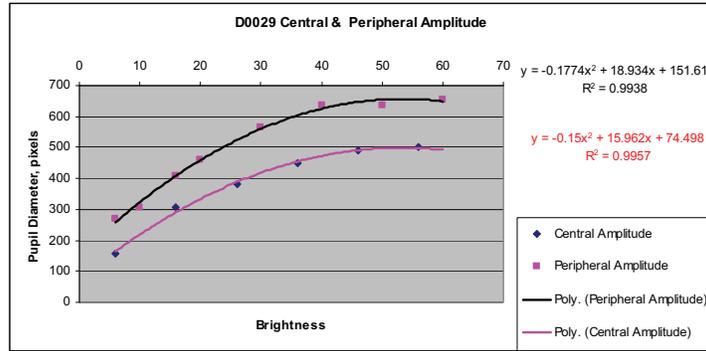
In Figure 6.14<sup>2</sup>-(a,b), each bar represents the average normalized parameter value between the stimulus level of 10 and 60 for each individual. Figure 6.14-(c) shows the average latency for both central and peripheral tests. Table 6.4 shows the color of the bar and its represented prior knowledge of the subject with regard to clinical retinopathy. In each group, the subjects are listed in the order of increasing age. One can see that with the increasing severity of DR, the normalized constriction velocity and amplitude both show the trend of decreasing value, except for D0019, which seems to be an abnormal sample. Later the subject D0019 was found to have evidence of non-diabetic related retinal disorder not known before. In the group with minimal or no DR, the subjects show little traditional signs of DR. However, by our test, a few samples in this group show

<sup>2</sup>Plots are extracted from the report for the NIH Grant in [78].

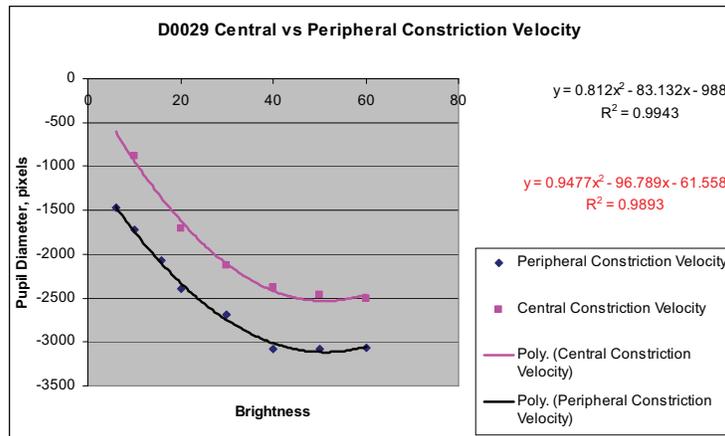
values clearly distinct from the normal group, and several of them were found to have significant peripheral ischemia by Optos fluorescein angiography with wide angle studies covering up to 200 degrees of the retina [78]. This result shows the potential of using the device to screen DR at the very early stage. A much large number of samples are needed to analyze the system quantitatively, and to help building the statistical model for the actual screening process. The latency plot in Figure 6.14 shows less change between groups and further research is needed to determine whether this feature is useful in the screening process.

## 6.6 Summary

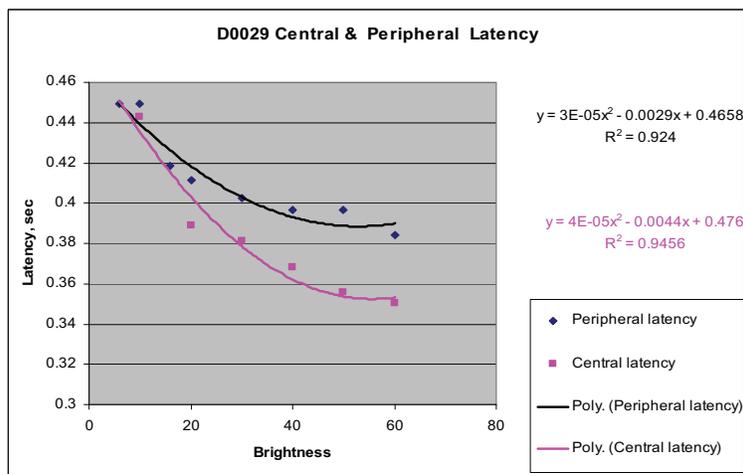
This section presented an accurate and robust image based pupillometry designed for the screening of DR based on different pupil's response to peripheral and central stimulus. Test results on pupil area measurement show that the designed system provides a more accurate measurement of the pupil size under interferences compared with the best known online pupil area calculation algorithm. It runs at  $80Hz$  for a binocular system and satisfies the frame rate requirement of the application. Using the online pupil monitoring system we have developed, a proof of concept test is completed, and the result shows that there is a clear correlation between the normalized value of constriction velocity, amplitude and the severity of DR, which was previously a hypothesis. Further research is needed to analyze the data on pupil's response to set up the norm for the purpose of screening DR or determining the degree of pathology.



(a) Subject D0029, Central and Peripheral Amplitude vs. Brightness

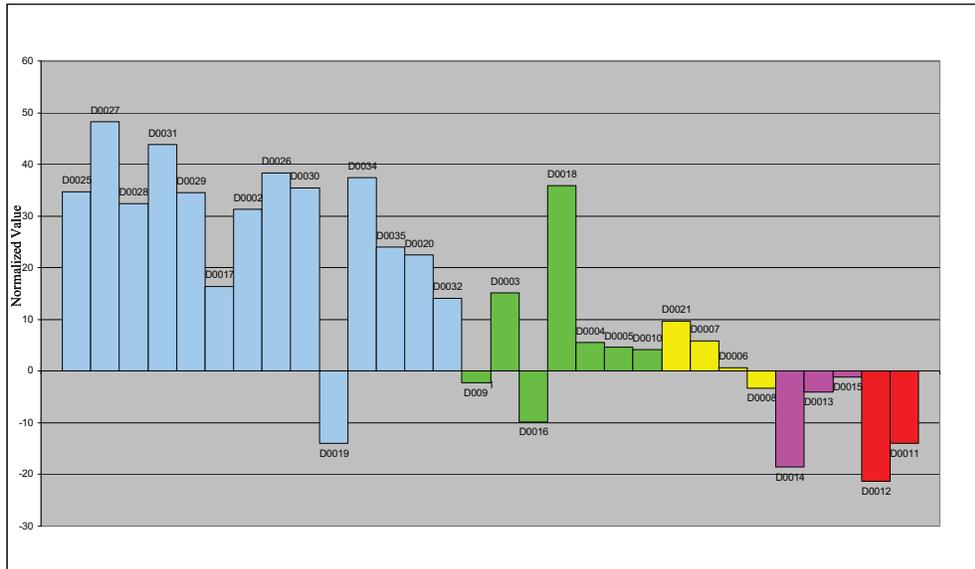


(b) Subject D0029, Central and Peripheral Maximum Constriction Velocity vs. Brightness

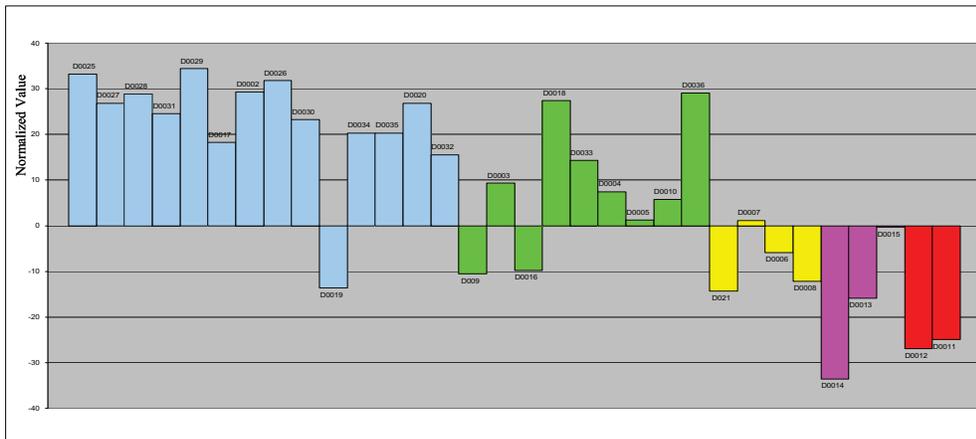


(c) Subject D0029, Central and Peripheral Latency vs. Brightness

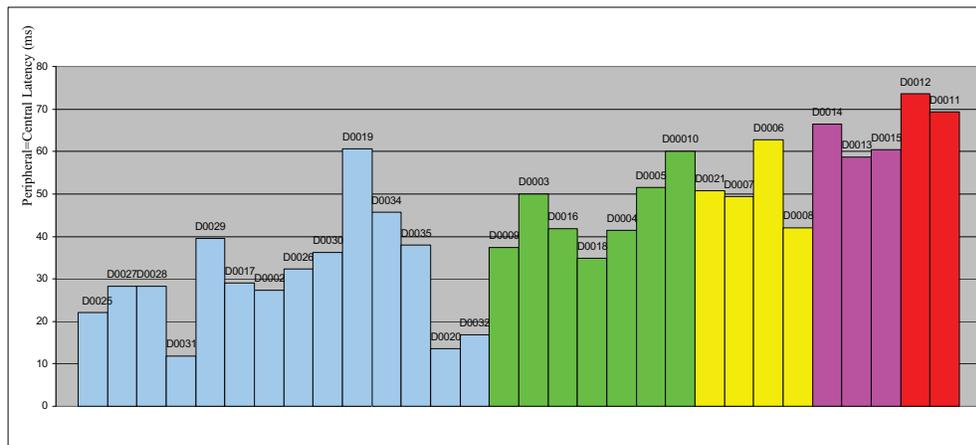
**Figure 6.13:** The average response for subject D0029, and the fitted second-order polynomial plots and equations.



(a) Normalized Peripheral vs Central Constriction Velocity



(b) Normalized Peripheral vs. Central Amplitude



(c) Peripheral - Central Latency (ms)

**Figure 6.14:** Feature parameters for each individual subject who participated in the test. Light blue: normal subjects; Green: diabetic patients with no DR; Yellow: diabetic patients with mild DR; Purple: diabetic patients with moderate DR; Red: diabetic patients with severe DR.

## **Part IV**

# **Conclusions and Future Work**

# Chapter 7

## Summary of Contributions and Future Work

The main contribution of the thesis is the design and implementation of an accurate, robust and fast eye tracking system. We proposed a new calibration algorithm to estimate imaging parameters, developed a simulation system to establish the ground truth, and implemented a high-speed horizontal and vertical tracking system, an online pupil monitoring system and a robust torsion tracking system.

We proposed a new calibration method to estimate the imaging parameters, which are needed to estimate the  $3D$  eye position. Previous calibration methods either ignore the horizontal/vertical rotation offset between the head frame and the camera frame, or approximate the imaging procedure as an orthographic projection, or assume a single center for horizontal and vertical eye rotation. These approximations produce measurements with large error when the eye is at a large angle. We use a more accurate model for the eye movement and employ numerical methods to estimate the imaging parameters. Simulation results show that the proposed method reduces the average error in measurement to less than  $0.02^\circ$  for a wide range of eye positions for both horizontal and

vertical eye position, while the current best available method has an average error of  $0.59^\circ$ ,  $0.84^\circ$  for horizontal and vertical eye position, respectively.

We developed a simulation system that can synthesize eye images using captured real eye images. The main purpose of our simulation effort is to provide close-to-real eye images in terms of pupil shape, iris pattern, and artifacts, and at the same time, a ground truth of the 3D eye position, which is very important information for evaluating the performance of different algorithms.

We developed a robust and online eye tracking system for measuring horizontal, vertical and torsion eye position and for monitoring the change in pupil size. For the horizontal and vertical eye position measurement, a high frame rate of  $150Hz$  for a binocular system is achieved by employing the multi-resolution processing technique. The new symmetric mass center algorithm also improves the measurement accuracy by up to  $66.6\%$  compared with the best available algorithm with similar frame rate. For the torsional eye tracking, the contribution lies in three aspects, firstly we collected more information in the calibration phase than other existing algorithms, which helps for the tracking later on and the cost in time is tolerable; secondly, a mechanism to eliminate outliers is developed to achieve a robust and accurate estimation; last but not least, an adaptive inter-frame searching range is used to ensure high frame rate and at the same time to avoid falling into the “erroneous” local minima or losing the tracking. A simulation test shows that compared to the best algorithm available, the robustness metric improved by about  $94\%$  and the accuracy metric improved by about  $81\%$  under similar computation cost.

Another contribution of the work is that we developed an online pupil size monitoring system and analyzing method for the application of diabetic retinopathy screening. Test results on pupil area measurement show that the designed system provides a more accurate measurement of the pupil size under interferences compared with the best known online pupil area calculation algorithm. It

runs at  $80Hz$  for a binocular system and satisfies the frame rate requirement of the application. Using the system we developed, a proof of concept test on human subjects is done to verify the DR screening method proposed by Dr. Gorin, which is based on different pupil response to peripheral and central stimulus; the test shows that there is a clear correlation between the normalized value of constriction velocity, amplitude and the severity of diabetic retinopathy, which was the presumed hypothesis before these measurements were made.

One innovation of this PhD dissertation is that we investigated the relationship of the shapes of pupil and iris boundary in the eye images captured at different  $3D$  positions. This relationship is used in the calibration process to estimate the projecting parameters, e.g., the camera offset; it is used in the pupil size monitoring algorithm to “predict” the pupil shape when the pupil center is known; it is used in the iris localization algorithm as well. In the calibration case, for instance, it gives more constraints than the pupil center, and in the other two cases, it results in a faster and more accurate estimation for the pupil and iris boundary.

One very interesting direction for future work is the analysis of eye tracking traces and norm establishment for the diagnosis or screening of certain diseases. For example, in the diabetic retinopathy screening case, more tests on human subjects and further research are needed to analyze the data on pupil response and to set up the norm for the purpose of screening DR or for determining the degree of pathology. Self calibration without laser guide is another interesting research topic. For eye tracking systems used in clinical environment where there is no sophisticated device for laser guided calibration, or for portable systems that the subjects bring back home to record eye movements, a self calibration system may be needed. Self calibration is possible, in theory, if sufficient number of views of the eye at different positions are captured and the homography is calculated between them. The current feature extraction tool for the DR Screening software is

still semi-automatic and requires human intervention sometimes to eliminate artifacts; a software system which is fully automatic is desired and will be developed in the future. Testing with real captured eye images can be performed to further validate the accuracy and effectiveness of our algorithm and system. It is desirable to further improve the frame rate for certain applications, and this can be achieved by utilizing custom designed hardware, making use of the computing power of GPU (Graphical Computing Unit), or/and performing adaptable image acquisition, which only acquires part of the image necessary according to the current eye position.

# **Part V**

## **Appendix**

## **Appendix A**

# **Homography Between Different Views of the Eye and Relation Between Shapes of Pupil (Iris) Boundaries**

In this chapter, we derive the homography between different views of the eye and the mathematical relationship between the shapes of the pupil(iris) boundary in two images. The relation between the equation of the ellipse fitting the pupil(iris) boundary in one image in a known eye position, and the equation of the ellipse fitting the pupil(iris) boundary in the eye coordinate system is derived as well. We first present the method to calculate the homography between them, then we describe the mathematical relationship between the ellipses.

## A.1 Derivation of Homographies

Let  $H_{0\leftarrow 1}$  be the  $3 \times 3$  homography matrix between the image for the current eye position  $[\theta_1, \phi_1, \psi_1]^T$ , and the image for the central eye position  $[0, 0, 0]^T$ , one has:

$$\begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} \simeq H_{0\leftarrow 1} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}, \quad (\text{A.1})$$

in which, the homogeneous coordinate vector  $[u_1, v_1, 1]^T$  is the projection of an object point in the image when the eye is in current position, and  $[u_0, v_0, 1]^T$  is the projection of the same object point in the image when the eye is in the central position.

From Section 3.2.1, we have:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} M_r \begin{bmatrix} R_{tp}(\theta, \phi) & F_C(\alpha, \theta, \phi) \\ \mathbf{0} & 1 \end{bmatrix} R(\psi) \end{bmatrix} \begin{bmatrix} \frac{x^e}{r_p} \\ \frac{y^e}{r_p} \\ 1 \\ 1 \end{bmatrix}, \quad (\text{A.2})$$

in which  $R_{tp}(\theta, \phi)$ , and  $F_C(\alpha, \theta, \phi)$  are defined as in (3.2).

Let

$$G \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} M_r \begin{bmatrix} R_{tp}(\theta, \phi) & F_C(\alpha, \theta, \phi) \\ \mathbf{0} & 1 \end{bmatrix} R(\psi) \end{bmatrix}. \quad (\text{A.3})$$

Let  $H_0$  represents the homography between the image in the central eye position and the pupil plane in the eye frame, and  $H_1$  represents the homography between the image in current eye position and the pupil plane in the eye frame, namely:

$$\begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} \simeq H_0 \begin{bmatrix} x_e/r_p \\ y_e/r_p \\ 1 \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \simeq H_1 \begin{bmatrix} x_e/r_p \\ y_e/r_p \\ 1 \end{bmatrix}. \quad (\text{A.4})$$

One has:

$$H_0 \simeq \mathfrak{E}_{3 \times 3} \left( \begin{bmatrix} 0 \\ G \\ 0 \\ 0 \end{bmatrix} \right) \quad \text{and} \quad H_1 \simeq \mathfrak{E}_{3 \times 3} \left( \begin{bmatrix} \theta_1 \\ G \\ \phi_1 \\ \psi_1 \end{bmatrix} \right), \quad (\text{A.5})$$

in which,  $\mathfrak{E}_{3 \times 3}$  represents a matrix operator defined as below:

$$\mathfrak{E}_{3 \times 3} \left( \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \right) = \begin{bmatrix} v_1 \\ v_2 \\ v_3 + v_4 \end{bmatrix}.$$

Furthermore, from (A.4) one has:

$$\begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} \simeq H_0 \times H_1^{-1} \times \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}. \quad (\text{A.6})$$

From (A.1), (A.6), it follows that the homography  $H_{0 \leftarrow 1}$  between the two images is:

$$H_{0 \leftarrow 1} \simeq H_0 \times (H_1)^{-1} \quad (\text{A.7})$$

## A.2 Mathematical Relation between the Shapes of Pupil Boundary

The pupil boundary in the eye frame is assumed to be a circle, and it appears as an ellipse when projected onto an image. An ellipse can be represented by a  $3 \times 3$  symmetric matrix  $C$  so that:

$$p^t C p = 0,$$

*APPENDIX A. HOMOGRAPHY BETWEEN DIFFERENT VIEWS OF THE EYE AND  
RELATION BETWEEN SHAPES OF PUPIL (IRIS) BOUNDARIES*

---

in which  $p = [x, y, 1]^T$  is the homogeneous coordinate of a point on the ellipse.

Let the pupil boundary in the template image in the central eye position  $([0, 0, 0]^T)$  be an ellipse characterized by a  $3 \times 3$  symmetric matrix  $C_0$ :

$$[u_0 \ v_0 \ 1] \times C_0 \times \begin{bmatrix} u_0 \\ v_0 \\ 1 \end{bmatrix} = \mathbf{0}, \quad (\text{A.8})$$

and let the the pupil boundary in current position  $[\theta_1, \phi_1, 0]^T$  be an ellipse defined by a  $3 \times 3$  symmetric matrix  $C_1$ :

$$[u_1 \ v_1 \ 1] \times C_1 \times \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \mathbf{0}. \quad (\text{A.9})$$

From (A.1), (A.8), and (A.9), one obtains:

$$C_1 \simeq H_{0 \leftarrow 1}^T \times C_0 \times H_{0 \leftarrow 1}. \quad (\text{A.10})$$

In the eye coordinate system, the pupil boundary can be approximated by a circle around the pupil center, which means points on the pupil boundary satisfies:

$$\begin{bmatrix} \frac{x^e}{r_p} & \frac{y^e}{r_p} & 1 \end{bmatrix} C^e \begin{bmatrix} \frac{x^e}{r_p} \\ \frac{y^e}{r_p} \\ 1 \end{bmatrix} = 0, \quad \text{and} \quad C^e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -r^2 \end{bmatrix}, \quad (\text{A.11})$$

in which,  $r$  is the pupil radius. When the pupil size changes, assuming symmetric change in the pupil boundary, the radius  $r$  in (A.11) changes. From (A.4), (A.8), (A.9), and (A.11), one obtains:

$$C^e \simeq H_1^T \times C_1 \times H_1 \quad (\text{A.12})$$

$$C^e \simeq H_0^T \times C_0 \times H_0$$

*APPENDIX A. HOMOGRAPHY BETWEEN DIFFERENT VIEWS OF THE EYE AND  
RELATION BETWEEN SHAPES OF PUPIL (IRIS) BOUNDARIES*

---

If the camera projection parameters and the eye position are known, from (A.12) one can derive the ellipse equation for the pupil boundary in the image from the ellipse equation for the pupil boundary in the eye coordinate system, or vice versa. Furthermore, since the pupil boundary itself is modeled as a circle in the eye coordinate system, the effect of torsional motion to the shape of pupil boundary in the image can be ignored. By assuming zero torsional movement, one can derive the homography from (A.5) and (A.7) between the two images once the horizontal and vertical eye position is known, hence derive the pupil shape in that position.

# Bibliography

- [1] Kuroiwa Y. and Toda H., “Clinical aspects of abnormal eye movements,” *Rinsho Shinkeigaku = Clinical Neurology*, vol. 43, no. 11, pp. 765–768, Nov. 2003.
- [2] Shimizu N., “Diagnosis and clinical assessment of abnormal eye movements,” *Rinsho Shinkeigaku = Clinical Neurology*, vol. 35, no. 12, pp. 1351–1355, Dec. 1995.
- [3] Michael B. Gorin, “Method and apparatus for screening for retinopathy,” *Patent filed internationally*, 2006, information available at: <http://www.wipo.int/pctdb/en/wo.jsp?wo=2006041625>.
- [4] Susan J. Herdman, *Vestibular Rehabilitation*, F. A. Davis Company, 2007.
- [5] Fetter M. and Sievering F., “Three-dimensional eye movement analysis in benign positioning vertigo and nystagmus,” *Acta Otolaryngology*, vol. 115, no. 3, pp. 353–357, May 1995.
- [6] R.S. Allison, M. Eizenman, and B.S.K. Cheung, “Combined head and eye tracking system for dynamic testing of the vestibular system,” *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 11, pp. 1073–1082, Nov. 1996.
- [7] Ozgirgin N., “Vestibular screening,” *Pro Otology*, vol. 3, no. 1, pp. 30–35, 2003.
- [8] “Discussion (section 2),” in *Proceedings 3D VOG Workshop*, Tübingen, Germany, online resource, available at: [http://web.unispital.ch/neurologie/vest/VOG/VOG\\_workshop.pdf](http://web.unispital.ch/neurologie/vest/VOG/VOG_workshop.pdf).
- [9] D. A. Robinson, “A method of measuring eye movements using a scleral search coil in a magnetic field,” *IEEE Transactions on Biomedical Engineering*, vol. 10, pp. 137–45, Oct. 1963.
- [10] J. M. Furman, H. Collewijn, T. C. Jansen, and A. V. Van Den Berg, “Human gaze stability in the horizontal, vertical and torsional direction during voluntary head movements, evaluated with a three-dimensional scleral induction coil technique,” *Vision Research*, vol. 27, no. 5, pp. 811–828, 1987.
- [11] A. Duchowski, “Eye-based interaction in graphical systems: Theory and practice,” *online resource*, *Clemson University*, available at: <http://vret.ces.clemson.edu/sigcourse/notes/c05-TOC.pdf>.
- [12] D. V. Finocchio, K. L. Preston, and A. F. Fuchs, “Obtaining a quantitative measure of eye movements in human infants: A method of calibrating the electrooculogram,” *Vision Research*, vol. 30, no. 8, pp. 1119–1128, 1990.
- [13] S. Shih, Y. Wu, and J. Liu, “A calibration-free gaze tracking technique,” in *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, 2000, vol. 4, pp. 4201–4205.

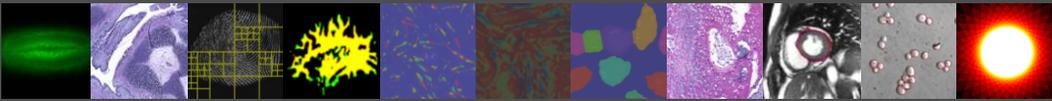
- 
- [14] K. Irie, B. A. Wilson, R. D. Jones, P. J. Bones, and T. J. Anderson, "laser-based eye-tracking system," *Behavior Research Methods, Instruments, & Computers*, vol. 34, no. 4, pp. 561–572, 2002.
- [15] B. A. Wilson, R. D. Jones, P. J. Bones, and T. J. Anderson, "A flying-spot laser scanner for tracking eye movements," *Proceedings of the 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Bridging Disciplines for Biomedicine*, vol. 1, pp. 146–147, 1996.
- [16] David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2002.
- [17] E. Wolff, *The anatomy of the eye and orbit, 2nd Edition ed. 1940*, London:H.K. Lewis & Co, 2002.
- [18] S. T. Moore, T. Haslwanter, I. S. Curthoys, and S. T. Smith, "A geometric basis for measurement of three-dimensional eye position using image processing," *Vision Research*, vol. 36, pp. 445–459, 1996.
- [19] Danjie Zhu, Mikhail Kunin, and Theodore Raphan, "Robust high-speed binocular 3d eye movement tracking system using a two-radii eye model," in *Proc. of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2006)*, New York City, USA, Aug. 2006.
- [20] Kai Schreiber and Thomas Haslwanter, "Improving calibration of 3-d video oculography systems," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 4, pp. 676–679, 2004.
- [21] G. A. Fry and W. W. Hill, "The center of rotation of the eye," *American Journal of Optometry*, vol. 39, no. 11, pp. 581–595, 1962.
- [22] Sensor Motoric Instruments, "Fastest video-based eye tracking," *online resource*, available at: <http://www.smi.de/>.
- [23] Applied Science Laboratories, "Innovation in eye tracking," *online resource*, available at: <http://www.a-s-l.com/>.
- [24] SR Research, "Complete eye tracking solutions," *online resource*, available at: <http://www.sr-research.com/>.
- [25] R. John Leigh and David S. Zee, *The neurology of eye movements, 4th edition*, Oxford University Press, 2006.
- [26] A. H. Clarke, C. Steineke, and H. Emanuel, "High image rate eye movement measurement : A new generation of tracker using cmos image sensors and dedicated fpga processing," in *Proceedings 3d VOG Workshop*, Tubingen, Nov. 1999.
- [27] Z. Zhu, K. Fujimura, and Q. Ji, "Real-time eye detection and tracking under various light conditions," in *Proceedings of the 2002 symposium on Eye tracking research and applications*, ACM Press, 2002.
- [28] L. Dongheng, D. Winfield, and D. J. Parkhurst, "Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005)*, San Diego, US, June 2005, vol. 3.

- [29] D. Zhu, S. T. Moore, and T. Raphan, "Robust pupil center detection using a curvature algorithm," *Computer Methods and Programs in Biomedicine*, vol. 59, no. 3, pp. 145–57, 1999.
- [30] E. Schneider, A. Lange, et al., "A video-based algorithm for measuring three-dimensional eye movements with scleral markers," in *Proceedings 3d VOG Workshop*, Tübingen, Nov. 1999.
- [31] J. E. Bos and B. de Graaf, "torsion quantification with video images," *IEEE Transactions on Biomedical Engineering*, vol. 41, no. 4, pp. 351–357, Apr. 1994.
- [32] M. Hatamian and D. J. Anderson, "Design considerations for a real-time ocular counter roll instrument," *IEEE Transactions on Biomedical Engineering*, vol. 30, pp. 278–288, 1983.
- [33] T. Vieville and D. Masse, "Ocular counter-rolling during active head tilting in humans," *Acta Otolaryngol.*, vol. 103, pp. 280–290, 1987.
- [34] S. G. McCoy S. T. Moore, I. S. Curthoys, "VTM – an image processing system for measuring ocular torsion," *Computer Methods and Programs in Biomedicine*, vol. 35, pp. 219–230, 1991.
- [35] E. Groen, J. E. Bos, P. F. M. Nacken, and de Graaf B., "Determination of ocular torsion by means of automatic pattern recognition," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 5, pp. 471–79, May 1996.
- [36] P. Guillemant, E. Ulmer, and G. Freyss, "3d eye movement measurements on four comex's divers using video ccd cameras, during high pressure diving," *Acta Otolaryngol. (Stockh)*, pp. 288–292, 1995, suppl 520.
- [37] Lee I.B., Choi B.H., Hwang J.M., Kim S.S., and Park K.S., "Measurement of ocular torsion using iterative optical flow," in *Proc. of the 26th Annual International Conference of the Engineering in Medicine and Biology Society, EMBC 2004*, 2004, vol. 2.
- [38] InBum Lee, ByungHun Choi Kwang, and Kwang Suk Park, "Robust measurement of ocular torsion using iterative lucas-kanade," *Computer Methods and Programs in Biomedicine*, vol. 3, 2007.
- [39] D. Ott and R. Eckmiller, "Ocular torsion measured by tv- and scanning laser ophthalmoscopy during horizontal pursuit in humans and monkeys," *Invest Ophthalmol Vis Sci*, vol. 30, pp. 2512–2520, 1989.
- [40] D. Zhu, ST. Moore, and T. Raphan, "Robust and real-time torsional eye position calculation using a template matching technique," *Computer Methods and Programs in Biomedicine*, vol. 74, pp. 201–209, 2004.
- [41] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, Mar. 2004.
- [42] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, June 2004.
- [43] P. N. Belhumeur G. D. Hager, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, Oct. 1998.

- [44] G.D. Rasmussen, C.; Hager, “Probabilistic data association methods for tracking complex visual objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, 2001.
- [45] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 5, 2003.
- [46] F. Jurie and M. Dhome, “Real time robust template matching,” in *British Machine Vision Conference*, Sept. 2002.
- [47] “Chronic disease overview,” *Internet Resource*, information available at: <http://www.cdc.gov/NCCdphp/overview.htm>.
- [48] The American Optometric Association, “Diabetes is the leading cause of blindness among most adults,” *Internet Resource*, information available at: <http://www.aoa.org/x6814.xml>.
- [49] ohn H. Kempen, Benita J. O’Colmain, and etc. M. Cristina Leske, “The prevalence of diabetic retinopathy among adults in the united states,” *Archives of Ophthalmology*, vol. 122, no. 4, pp. 552–563, 2004.
- [50] Jinan B. Saaddine, Amanda A. Honeycutt, and etc. K. M. Venkat Narayan, “Projection of diabetic retinopathy and other major eye diseases among people with diabetes mellitus united states, 2005-2050,” *Archives of Ophthalmology*, vol. 126, no. 12, pp. 1740–1747, 2005.
- [51] “National diabetes fact sheet, 2007,” *Internet Resource*, information available at: [http://www.cdc.gov/diabetes/pubs/pdf/ndfs\\_2007.pdf](http://www.cdc.gov/diabetes/pubs/pdf/ndfs_2007.pdf).
- [52] “Fact sheet diabetic retinopathy,” *Internet Resource*, information available at: <http://www.nih.gov/about/researchresultsforthepublic/DiabeticRetinopathy.pdf>.
- [53] “Facts about diabetic retinopathy,” *Internet Resource*, information available at: <http://www.nei.nih.gov/health/diabetic/retinopathy.asp>.
- [54] “Diabetes and retinopathy (eye complications),” *Internet Resource*, information available at: <http://www.diabetes.org/diabetes-statistics/eye-complications.jsp>.
- [55] F. Fotiou, K. N. Fountoulakis, A. Goulas, L. Alexopoulos, and A. Palikaras, “Automated standardized pupillometry with optical method for purposes of clinical practice and research,” *Clinical Physiology*, vol. 5, no. 20, pp. 336–347, 2000.
- [56] K. Karavanaki, G. Davies A, L. P. Hunt, P. H. Morgan, and J. D. Baum, “Pupil size in diabetes,” *Archive in Diseases in Childhood*, vol. 71, pp. 511–515, 1994.
- [57] O. Bergamin and R. H. Kardon, “Latency of the pupil light reflex: Sample rate, stimulus intensity, and variation in normal subjects,” *Investigative Ophthalmology and Visual Science*, vol. 44, no. 4, pp. 1546–1554, 2003.
- [58] Jieun Kim, Kyungmo Park, and Gon Khang, “A method for size estimation of amorphous pupil in 3-dimensional geometry,” in *Proceedings of the 26th Annual International Conference of the IEEE EMBS, EMBS 2004*, Sept. 2004, vol. 2.

- [59] D. R. Skander, M. J. Collins, S. Mioschek, and M. Trunk, "Automatic pupillometry from digital image," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 9, pp. 1619–1627, Sept. 2004.
- [60] Y. Chen, H. J. Wyatt, and W. H. Swanson, "Pupillary evaluation of retinal asymmetry: Development and initial testing of a technique," *Vision Research*, vol. 45, pp. 2549–2563, Sept. 2005.
- [61] A. A. Sewisy and F. Leberl, "Detection ellipses by finding lines of symmetry in the images via an hough transform applied to straight lines," *Image and Vision Computing*, vol. 19, pp. 857–866, 2001.
- [62] J.G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.
- [63] R.P. Wildes, "Iris recognition: an emerging biometric technology," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1348–1363, 1997.
- [64] T. A. Camus and R. Wildes, "Reliable and fast eye finding in closeup images," in *Proceedings. 16th International Conference on Pattern Recognition*, 2002, vol. 1, pp. 389–394.
- [65] C. Sanchez-Avila, R. Sanchez-Reillo, and D. de Martin-Roche, "Iris-based biometric recognition using dyadic wavelet transform," *IEEE Aerospace and Electronic Systems Magazine*, vol. 17, no. 10, pp. 3–6, 2002.
- [66] X. Liu, K. W. Bowyer, and P. J. Flynn, "Experiments with an improved iris segmentation algorithm," in *Proc. Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, 2005, pp. 118–123.
- [67] Xinhua Feng, Chi Fang, Xiaoqing Ding, and Youshou Wu, "Iris localization with dual coarse-to-fine strategy," in *Proc. 8th International Conference on Pattern Recognition (ICPR'06)*, 2006, pp. 553–556.
- [68] H. Proenca and L.A. Alexandre, "Iris segmentation methodology for non-cooperative recognition," in *IEE Proceedings Vision, Image and Signal Processing*, 2006, 2, pp. 199–205.
- [69] H. Ranjzad, H. Ebrahimnezhad, and A. Ebrahimi, "A novel compound approach for iris segmentation," in *Third International Conference on Convergence and Hybrid Information Technology, ICCIT 08.*, 2008, pp. 657–661.
- [70] Aditya Abhyankar and Stephanie Schuckers, "Active shape models for effective iris segmentation," in *Proc. SPIE: Biometric Technology for Human Identification*, 2006.
- [71] E.M. Arvacheh and H.R. Tizhoosh, "Iris segmentation: Detecting pupil, limbus and eyelids," in *Proc. IEEE International Conference on Image Processing*, 2006, pp. 2453–2456.
- [72] Zhaofeng He, Tieniu Tan, Zhenan Sun, and Xianchao Qiu, "Towards accurate and fast iris segmentation for iris biometrics," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, 9, pp. 1670–1684.

- [73] Jinyu Zuo, N.D. Kalka, and N.A. Schmid, "A robust iris segmentation procedure for unconstrained subject presentation," in *2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference*, Baltimore, MD, USA, 2006, pp. 1–6.
- [74] W. J. Ryan, D. L. Woodard, A. T. Duchowski, and S. T. Birchfield, "Adapting starburst for elliptical iris segmentation," in *BTAS 2008, 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*, Arlington, VA, US, 2008.
- [75] Carlo Colombo, Dario Comanducci, and Alberto Del Bimbo, "Robust tracking and remapping of eye appearance with passive computer vision," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 3, no. 4, 2007.
- [76] X. Long, O. K. Tonguz, and A. Kiderman, "Real time pupil size monitoring as a screening method for diabetic retinopathy," in *IEEE Biomedical Circuits and System Conference Nov. 27-30 2007 (BIOCAS 2007)*, Montreal, Canada, Nov. 2007.
- [77] X. Long, O. K. Tonguz, and A. Kiderman, "A high speed eye tracking system with robust pupil center estimation algorithm," in *Proc. of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, Lyon, France, Aug. 2007.
- [78] Alexander Kiderman, "Developing a noninvasive method and device for assessing the degree of mid-peripheral retinal ischemia in diabetic retinopathy," 2009, Phase 1 Closeout Report for NIH Grant 1R43EY018025-01.



## Abstract

This dissertation presents an image based eye tracking system for human disease diagnostic/screening systems. Current eye tracking systems either do not satisfy the speed or cost requirement, or are not robust enough for clinical use. This dissertation focuses on developing an accurate, robust, and fast online eye tracking system.

Discovering the 3D eye position from 2D images requires the recovering of image projection parameters through calibration. In contrast to existing work, we use the more accurate two-radius rotation model for eye movement and the perspective projection model for the imaging process. The parameters of the elliptical pupil boundary from multiple views of the eye are used as the constraints to estimate the projection parameters. Simulation results show remarkable improvements in measurement accuracy.

The dissertation presents an accurate, robust, and high-speed horizontal and vertical eye tracking system on a general computer system. A high frame rate of 150Hz for binocular systems is achieved by applying multi-resolution image processing techniques. A novel symmetric mass center algorithm is developed to handle occlusions. A torsional eye tracking system robust to interference is presented as well. Simulation results show significant improvement in accuracy and robustness.

This dissertation presents an accurate and online pupil size monitoring system for the screening of Diabetic Retinopathy based on pupil response. Test results on pupil area measurement show that the designed system provides a more accurate measurement of the pupil size under interference. The dissertation also presents a proof of concept test on human subjects to verify the screening method using the pupil monitoring system developed.