GAZE ESTIMATION FROM RGB-D CAMERA AND HEAD-MOUNTED EYE CAMERA

(遠隔 RGBD カメラ及びヘッドマウントアイカメラによる視線推定)

by

Jianfeng LI

Doctor of Engineer

Electrical and Electronic Engineering

Tottori University

March, 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This thesis focuses on the topic of gaze estimation from cameras. Gaze estimation is an important topic in computer vision in such areas as driver behavior analysis, security monitoring, behavior investigation, and human-computer interfaces. In particular, gaze information can offer a new means of communication with machines, such as determining a human's region of interest. In addition, there are two big categories in this area: one is gaze estimation from a remote camera, which is placed in front of an observer, and the observer would not have any direct bindings with the remote camera. The other is gaze estimation from a head-mounted eye camera, which is placed on the observer's head. The camera can have a direct view of the eye. This thesis proposes novel methods of gaze estimation based on an eye model for the remote camera and the head-mounted camera, and is composed by two parts as follows.

Part I (Gaze estimation from remote camera): The most crucial factors in the eye-model-based approach to gaze estimation are the three-dimensional (3D) positions of the eyeball and iris centers. In the proposed method, a RGB-D camera, Kinect sensor, is used to obtain the head pose as well as the eye region of the color image. Because the ray from the eyeball center to target and the ray from the eyeball center to the iris center should meet a relationship. Based on the knowledge, our method sets up a model to calibrate the eyeball center by gazing at the center of the color image camera. Then, to estimate the 3D position of the iris center, the 3D contour of the iris is projected onto the color image with the known head pose

obtained from color and depth cues of an RGB-D camera. Thus, the ellipse of the iris in the image can be described using only two parameters: the yaw and pitch angles of the eyeball in the iris coordinate system, rather than the conventional five parameters of an ellipse. The proposed method can fit an iris that is not complete due to eyelid occlusion. The average errors of vertical and horizontal angles of the gaze estimation for seven subjects are 5.9 degrees and 4.4 degrees in experiments, respectively. However, for lower resolution and poor illumination images, as tested on the public database EYEDIAP, the performance of the proposed eye-model-based method is inferior to that of the-state-of-the-art appearance-based method.

Part II (Gaze estimation from head-mounted camera): As introduced in Part I. Gaze estimation is based on the eyeball center and the iris center, so in this proposed method, we divide the continuous gaze estimation of a head-mounted eye camera into two phases. One phase, known as the calibration phase, is used to estimate the eyeball center position in relation to the coordinate system of the head-mounted eye camera. The other phase is used to fit the iris contour in 2D images employing only two parameters for gaze estimation. Based on an eye-model, iris can be extracted in a more efficient and accurate manner by projecting the 3D iris contour onto a 2D space. Given the calibrated 3D eyeball center and estimated 3D iris center, the gaze tracking can be achieved. As seen from the experimental results, the proposed method demonstrates both credible eyeball center estimation and an accurate iris contour estimation in comparison with the conventional approach using five unknown parameters. At the end, the accuracy of our gaze estimation method and other existed methods using targets on a screen was evaluated.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. Shigang Li for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank Prof. Nakanishi. Without his consistent and illuminating instruction, this thesis could not have reached its present form. And I would also like to thank Prof. Ito and Prof. Iwai, their treasure advises did a great help to this thesis.

My sincere thanks also goes to my friends, especially Dr. Hanchao Jia and Dr. Wuhe Zou, who have instructed and helped me a lot in my first year.

Last but not the least, I would like to thank my family: my parents, my sister and my wife for supporting me spiritually throughout writing this thesis and my life in general.

CHAPTER 1

Introduction

1.1 Approaches to eye movement detection

Our eyes are one of the most significant sense organs, which allow us to explore, analysis, and interact with environments by the visual information content of the physical world. The eye and its movements provide a key contribution to interpreting and understanding a person's wishes, needs, tasks, cognitive processes, affective states and interpersonal relations.

The unique geometric and photometric properties of the eyes provide important visual cues for obtaining face-related information. Thus, the research about gaze estimation becomes important.

To measure rotations of the eye, there are principally three categories (Table 1.1).

- Measurement of the movement of an object attached to the eye, such as a special contact lens with an embedded mirror or magnetic field sensor, and the movement of the attachment is measured with the assumption that it does not slip significantly as the eye rotates [38].
- Measurement of electric potentials using electrodes placed around the eyes, the electric signal that can be derived using two pairs of contact electrodes placed on the skin around one eye called Electrooculogram (EOG). If the eyes move from the center position towards the periphery, the retina approaches one electrode while the cornea approaches the opposing one. This change in the orientation of the dipole and consequently the electric potential field result in a change in the measured

EOG signal. Inversely, by analyzing these changes, eye movement can be tracked [39].

- Optical tracking without direct contact to the eye, it uses some non-contact, optical method for measuring eye motion. Light, typically infrared is reflected from the eye and sensed by a video camera or some other specially designed optical sensors.

In this thesis, we focus on the optical tracking approach using a camera.

TABLE 1.1
THREE CATEGORIES OF MEASURING ROTATIONS OF EYES

Categories	Devices
Measurement of the movement of an object attached to the eye	A special contact lens with an embedded mirror or magnetic field sensor
Measurement of electric potentials	Electrodes placed around the eyes
Optical tracking to the eye	A video camera or some other specially designed optical sensors

1.2 Eye movement detection by camera

Gaze estimation is the process of measuring either the point of gaze or the motion of an eye relative to the head. Gaze estimation is a hot topic in computer vision in such areas as driver behavior analysis, security monitoring, behavior investigation, and human-computer interfaces [1]. In particular, gaze information can offer a new means of communication with machines, such as determining a human's region of interest. The most widely used current designs are video-based gaze estimation. A camera focuses on one or both eyes and records their movement as the viewer looks at some kind of stimulus. Since gaze direction corresponds to human eyeball movement. The most crucial factors in estimating gaze direction are the eyeball and iris/pupil centers. While the iris/pupil center can be directly observed using a camera, the eyeball center must be calibrated. Most modern gaze estimation methods use the center of the pupil and infrared/near-infrared non-collimated light to create corneal reflections (glint). The vector between the pupil center and the corneal reflections can be used to compute the point of regard on surface or the gaze direction. A simple calibration procedure of the individual is usually needed before using the gaze estimation method.

However, the gaze estimation system based on infrared illumination have many limitations like below:

1. The infrared illumination can be affected by the sunshine in outdoor scenario.
2. The relative position between the infrared lights and the camera need to be calibrated carefully.

3. The pupil and the glint are very small, usually a high-resolution camera is needed.

So most current gaze estimation system can only work in indoor scenario. To solve this problem and apply the gaze estimation system in outdoor scenario, we focus on the research that estimates the gaze by common cameras, including the remote camera and the head-mounted camera.

For the remote camera, since an eyeball is a part of the head, gaze direction can be represented in the coordinate system of the head if the head pose is known. To estimate the gaze estimation, we must first detect the face in the image. This requires a platform that can detect the face and determine the head pose quickly and accurately when constructing a gaze estimation system. Refer to the platform, there are many mature commercial products existing until now. One of them, Kinect sensors, which can acquire both color and depth cues, have become widely used in human–machine interaction tasks. Based on the Kinect SDK (Software Development Kit), head pose and face tracking can be acquired at the rate of 30 fps. Figure 1.1 shows the platform and the head pose and a face tracking result.

For the head-mounted camera, since the camera can have a direct view of the eye, we do not need any information about the head pose or the eye position in images. To achieve such an eye image, we put a common RGB camera on a glasses frame. The platform and the image acquired by the camera are shown in Fig. 1.2. Since the eye camera is mounted on the head, the position of eyeball center at the coordinate system of head-mounted eye camera does not change unless the position of eye camera is changed. Once the position of eye center is known, what we need to do is only to determine the direction vector of gaze.

Fig. 1.1: The Kinect sensor and the head pose detection.

Fig. 1.2: Head-mounted camera and captured eye image.

1.3 Contributions

In this thesis, we focus on gaze estimation from color cues based on an eye model.

For the remote camera, we assume the head pose is known from depth cues from a RGB-D camera. In comparison to other eye-model-based methods of gaze estimation, the contributions are as follows:

- An eyeball center calibration protocol is introduced. In contrast to the conventional target-gazed calibration methods, the protocol requires the user to fixate on the camera from different directions, effectively generating samples of the eyeball oriented in different directions and reducing occlusions by a target located in front of the user.
- We present an innovative method for iris fitting, the problem of determining the eyeball orientation that aligns best with the edges found in the color image. Instead of the conventional five parameters for fitting an ellipse, our fitting process requires determining only two unknown parameters.

For the head-mounted camera, we propose a new method for the estimation of the eyeball center position and the iris contour separately for gaze tracking. By decoupling these two factors, we propose to make two contributions to the process of gaze estimation using a head-mounted eye camera as follows:

- We represent the ellipse of an iris contour projected onto the image using only two parameters. To the best of knowledge, in all estimation approaches using a head-mounted eye camera, the fitting of an ellipse to

the iris contour is performed using five unknowns during the entire process of gaze tracking.

- The other contribution is related to the calibration of the center of the eyeball. In contrast to existing methods that estimate the eyeball center position by fitting an ellipse to the iris contour using five parameters [24] and other offline calibration methods [3, 4], automatic online eyeball center calibration is realized by using our iris fitting method with two parameters iteratively and employing a multiple frame strategy.

1.4 Organization

The rest of this thesis is organized as follows: Related research is introduced in the next chapter. Chapter 3 introduces the basic and related knowledge that can help the readers to understand our method easier. In Chapter 4, the method that gaze estimation from a remote camera is introduced. And the method that gaze estimation from a head-mounted camera is introduced in Chapter 5. Finally, conclusions are given in Chapter 6.

CHAPTER 2

Fundamentals

As the subject of this thesis is gaze estimation on the remote camera and the head-mounted camera, first some fundamentals need to be introduced to help understanding this thesis. In this chapter, we start with the explanation of a physical model of human's eye. Then the camera model and projection are introduced. To understand this thesis better, the knowledge of ellipse fitting are recommended, we would introduce this part at the end of the chapter.

2.1 Eye

The eyes are wonderful sensory organs. They help people learn about the world in which people live. Eyes see all sorts of things - big or small, near or far, smooth or textured, colors and dimensions. The eyes have many parts - all of which must function in order to see properly.

2.1.1 Inside the eye

In addition to the many sections of the eyeball itself, muscles are attached to the outer walls of the eyeball. The eye muscles are attached to eyes in order to move the eyes. Figure 2.1 shows these main parts. If anything goes wrong, such as from diabetic eye disease, an individual might not be able to see as well. Visual information from the retina in the eye travels to the brain via the optic nerve. When eyes see an object, two images from them are slightly different since there is a distance between them. Therefore, the brain must mix the two images to get a complete picture.

What we think of as seeing is the result of a series of events that occur among the eye, the brain, and the outside world. Light reflected from an object passes

through the cornea of the eye, moves through the lens which focuses it, and then reaches the retina at the very back where it meets with a thin layer of color-sensitive cells called the rods and cones. Because the light crisscrosses while going through the cornea, the retina "sees" the image upside down. The brain then "reads" the image right-side up [41].

Glossary

- Aqueous Humor: a clear, watery fluid that fills the front part of the eye between the cornea, lens and iris.
- Cornea: the transparent outer portion of the eyeball that transmits light to the retina.
- Fovea: A tiny spot located in the macula that is the area of clearest vision on the retina.
- Iris: the colored, circular part of the eye in front of the lens. It controls the size of the pupil.
- Lens: the transparent disc in the middle of the eye behind the pupil that brings rays of light into focus on the retina.
- Optic Nerve: the important nerve that carries messages from the retina to the brain.
- Retina: the inner layer of the eye containing light-sensitive cells that connects with the brain through the optic nerve. It also contains retinal blood vessels which feed the retina and which can be affected by diabetes.
- Sclera: the white part of the eye that is a tough coating which, along with the cornea, forms the external protective coat of the eye.

- Vitreous Humor: a colorless mass of soft, gelatin-like material that fills the eyeball behind the lens.
- Macula: is a small area of the retina located near the optic nerve at the back of the eye. It is responsible for our central, most acute vision.
- Pupil: the circular opening at the center of the iris that controls the amount of light into the eye.

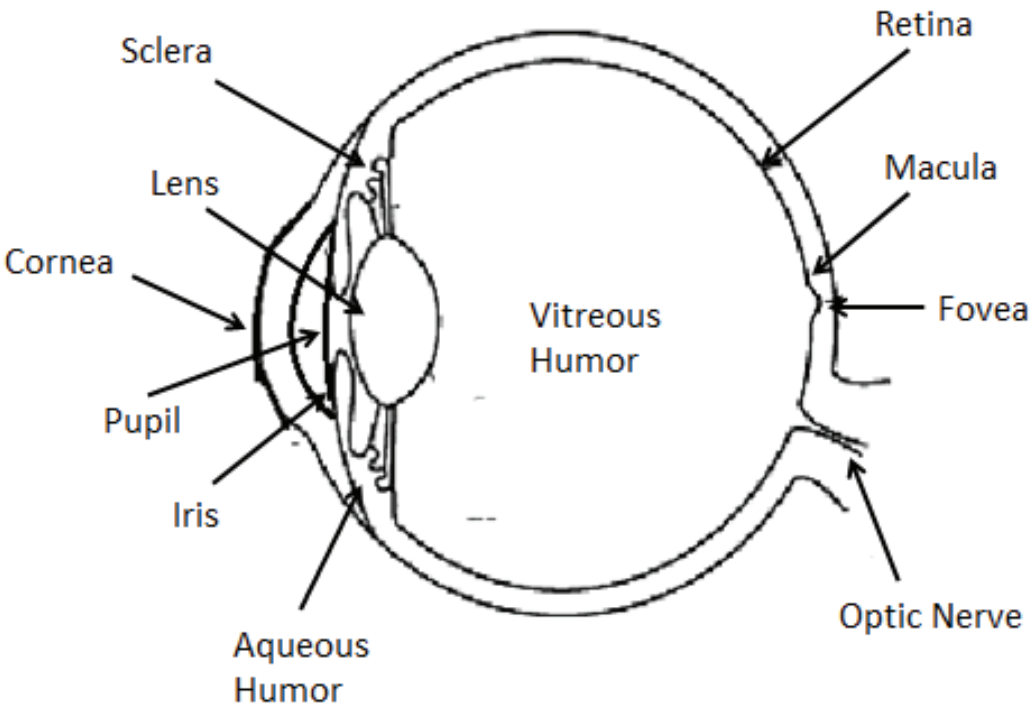


Fig. 2.1: Eye structure

2.1.2 Working principle of eye

This section refers to Ref. [40], which can help readers to understand the working principle of eye. The eye is made up of three layers: the outer layer called the fibrous tunic, which consists of the sclera and the cornea; the middle layer responsible for nourishment, called the vascular tunic, which consists of the iris, the choroid, and the ciliary body; and the inner layer of photoreceptors and neurons called the nervous tunic, which consists of the retina.

The eye also contains three fluid-filled chambers. The volume between the cornea and the iris is known as the anterior chamber, while the volume between the iris and the lens is known as the posterior chamber, both chambers contain a fluid called aqueous humor. Aqueous humor is watery fluid produced by the ciliary body. It maintains pressure and provides nutrients to the lens and cornea. Aqueous humor is continually drained from the eye through the Canal of Schlemm. The greatest volume, forming about four-fifths of the eye, is found between the retina and the lens called the vitreous chamber. The vitreous chamber is filled with a thicker gel-like substance called vitreous humor which maintains the shape of the eye.

Light enters the eye through the transparent, dome shaped cornea. The cornea consists of five distinct layers (Fig. 2.2). The outermost layer is called the epithelium which rests on Bowman's Membrane. The epithelium has the ability to quickly regenerate while Bowman's Membrane provides a tough, difficult to penetrate barrier. Together the epithelium and Bowman’s Membrane serve to protect the cornea from injury. The innermost layer of the cornea is called the endothelium which rests on Descemet's Membrane. The endothelium removes water from cornea, helping to keep

the cornea clear. The middle layer of the cornea, between the two membranes is called the stroma and makes up 90% of the thickness of the cornea.

From the cornea, light passes through the pupil. The amount of light allowed through the pupil is controlled by the iris, the colored part of the eye. The iris has two muscles: the dilator muscle and the sphincter muscle. The dilator muscle opens the pupil allowing more light into the eye and the sphincter muscle closes the pupil, restricting light into the eye. The iris has the ability to change the pupil size from 2 millimeters to 8 millimeters.

Just behind the pupil is the crystalline lens. The purpose of the lens is to focus light on the retina. The process of focusing on objects based on their distance is called accommodation. The closer an object is to the eye, the more power is required of the crystalline lens to focus the image on the retina. The lens achieves accommodation with the help of the ciliary body which surrounds the lens. The ciliary body is attached to lens via fibrous strands called zonules. When the ciliary body contracts, the zonules relax allowing the lens to thicken, adding power, allowing the eye to focus up close. When ciliary body relaxes, the zonules contract, drawing the lens outward, making the lens thinner, and allowing the eye to focus at distance.

Light reaches its final destination at the retina. The retina consists of photoreceptor cells called rods and cones. Rods are highly sensitive to light and are more numerous than cones. There are approximately 120 million rods contained within the retina, mostly at the periphery. Not adept at color distinction, rods are suited to night vision and peripheral vision. Cones, on the other hand, have the primary function of detail and color detection. There are only about 6 million cones contained within the retina, largely concentrated in the center of the retina called the

fovea. There are three types of cones. Each type receives only a narrow band of light corresponding largely to a single color: red, green, or blue. The signals received by the cones are sent via the optic nerve to the brain where they are interpreted as color. People who are color blind are either missing or deficient in one of these types of cones.

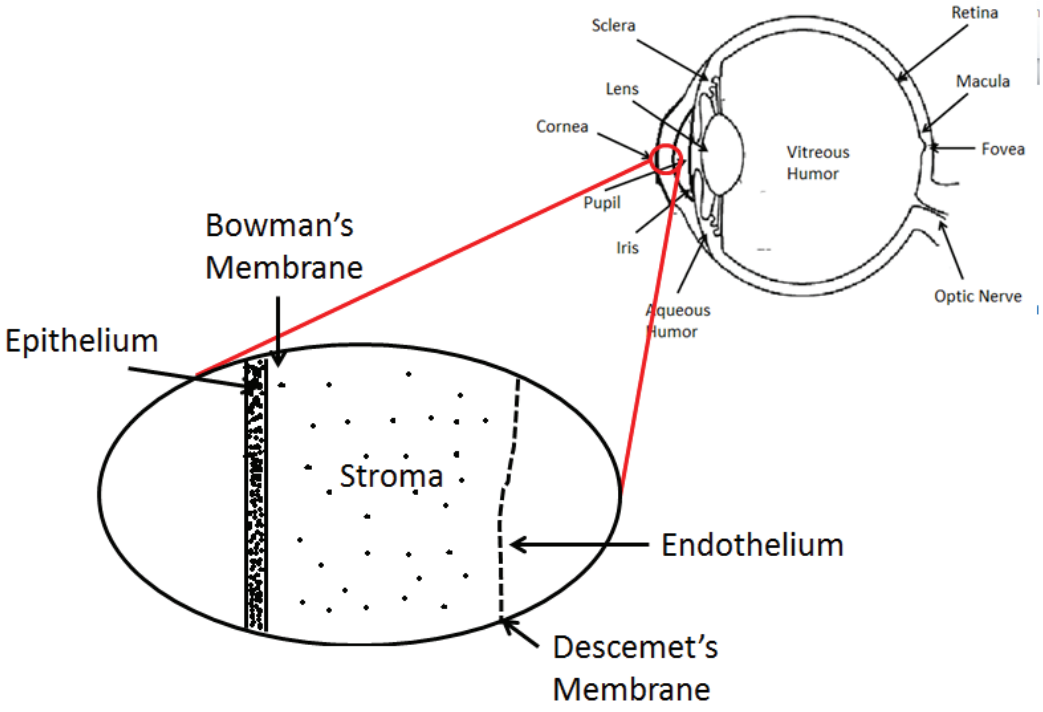


Fig. 2.2: Cornea structure

2.2 Camera model

A camera model is a function which maps our 3-dimensional world onto a 2-dimensional plane, called the image plane. Generally, this function is designed to closely model a real-world, physical camera. There are many camera models of varying complexity, and a natural dividing line which helps categorize them is whether or not they are able to capture perspective.

2.2.1 Pinhole camera model

The Pinhole camera is the simplest camera in various camera models. When using a pinhole camera, this geometric mapping from 3D to 2D is called a perspective projection. The line perpendicular to the image plane passing through the camera center as the optical axis (Fig. 2.3). Moreover, the intersection point of the image plane with the optical axis is called the principal point. We assume that the image plane is positioned parallel to the xy-plane, at position z=*f*, so the coordinate of the principal point is $(0, 0, f)^T$.

Considering a 3D scene point $(X, Y, Z)^T$ and its corresponding image point $(u, v)^T$. By looking at similar triangles, the correspondence of these two points can be written as below,

$$\begin{cases} u = \frac{X*f}{Z} \\ v = \frac{Y*f}{Z}, \end{cases} \tag{2.1}$$

Then, to avoid such a non-linear division operation, the relation below can be reformulated using the projective geometry framework, as

$$(\lambda u, \lambda v, \lambda)^T = (Xf, Yf, Z)^T, \tag{2.2}$$

This relation can be expressed in matrix notation by

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.3}$$

Where $\lambda = Z$ is the homogenous scaling factor.

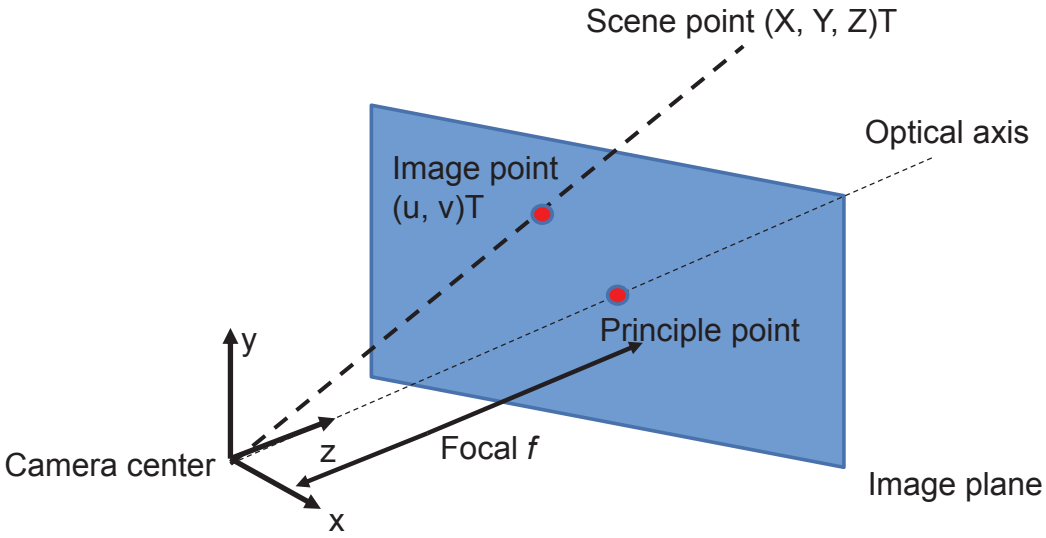


Fig. 2.3: Pinhole camera model

2.2.2 Camera calibration with OpenCV

This section refers to Ref. [42], which can help readers to understand the camera calibration with OpenCV. Cameras have been around for a long-long time. However, with the introduction of the cheap pinhole cameras in the late 20th century, they became a common occurrence in our everyday life. Unfortunately, this cheapness comes with its price: significant distortion. Luckily, these are constants and with a calibration and some remapping we can correct this. Furthermore, with calibration we may also determine the relation between the camera's natural units (pixels) and the real world units (for example millimeters).

For the distortion OpenCV takes into account the radial and tangential factors. For the radial factor one uses the following formula:

$$\begin{aligned} x_{corrected} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \tag{2.4}$$

So for an old pixel point at *(x, y)* coordinates in the input image, its position on the corrected output image will be ($x_{corrected}$, $y_{corrected}$). The presence of the radial distortion manifests in form of the "barrel" or "fish-eye" effect.

Tangential distortion occurs because the image taking lenses are not perfectly parallel to the imaging plane. It can be corrected via the formulas:

$$\begin{aligned} x_{corrected} &= x + [2p_1 xy + p_2(r^2 + 2x^2)] \\ y_{corrected} &= y + [2p_2 xy + p_1(r^2 + 2y^2)] \end{aligned} \tag{2.5}$$

So we have five distortion parameters which in OpenCV are presented as one row matrix with 5 columns:

$$Distortion_{coefficients} = (k_1\ k_2\ \ p_1\ \ p_2\ \ k_3) \tag{2.6}$$

Now for the unit conversion we use the following formula:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.7}$$

where the presence of λ is explained by the use of homography coordinate system (and $\lambda = Z$). The unknown parameters are f (camera focal lengths) and (c_x, c_y) which are the optical centers expressed in pixel's coordinate. The matrix containing these three parameters is referred to as the *camera matrix*. While the distortion coefficients are the same regardless of the camera resolutions used, these should be scaled along with the current resolution from the calibrated resolution.

The process of determining these two matrices is the calibration. Calculation of these parameters is done through basic geometrical equations. The equations used depend on the chosen calibrating objects. Currently OpenCV supports three types of objects for calibration:

- Classical black-white chessboard.
- Symmetrical circle pattern.
- Asymmetrical circle pattern.

Basically, we take snapshots of these patterns with the camera and let OpenCV find them. Each found pattern results in a new equation. To solve the equation we need at least a predetermined number of pattern snapshots to form a well-posed equation system. This number is higher for the chessboard pattern and less for the circle ones. For example, the chessboard pattern requires at least two snapshots. However, in practice we have a good amount of noise present in our input images, so for good results we will need at least 10 good snapshots of the input pattern in different positions.

2.2.3 The RGB camera

The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green and blue. An RGB camera delivers the three basic color components on three different wires. This type of camera often uses three independent CCD sensors to acquire the three color signals. RGB cameras are used for very accurate color image acquisitions.

2.2.4 The RGB-D camera

The RGB-D camera is based on the RGB camera, and the major difference is that an IR camera is used to take the depth data, that means by the RGB-D camera we can know the distance between the object and the camera. Sometimes the depth information is very useful in doing some research, RGB-D camera can supply us with more information (depth information) than the RGB camera and more convenient. Up to now, some commercial RGB-D cameras are widely used in many fields, like Microsoft Kinect, Inter RealSense, Asus Xtion Pro.

2.3 Ellipse fitting

Traditional iris fitting methods always take the iris contour as an ellipse. And based on detected iris edge points on an image, a set of the general equations of an ellipse can be formulated. Then by solving the nonlinear function set, the five parameters of the ellipse can be obtained, so the iris contour is obtained. We call these methods five parameters iris fitting methods (FIFM). The general equation of an ellipse is introduced below, which refers to Ref. [43].

The standard equation for an ellipse, $x^2 / a^2 + y^2 / b^2 = 1$, represents an ellipse centered at the origin and with axes lying along the coordinate axes. In general, an ellipse may be centered at any point, or have axes not parallel to the coordinate axes. But such an ellipse can always be obtained by starting with one in the standard position, and applying a rotation and/or a translation. For the most general formulation, we can include rotations through an angle of 0 (that is, no rotation at all) and translations by the zero vector (no translation at all). Then every ellipse can be obtained by rotating and translating an ellipse in the standard position. Accordingly, we can find the equation for any ellipse by applying rotations and translations to the standard equation of an ellipse.

It is a matter of choice whether we rotate and then translate, or the opposite. To see this, let R represent a rotation, and consider what happens to a point $x = (x, y)$ if we first translate by vector v, and then apply R. The result will be $R(x + v) = Rx + Rv$, because R is linear. But this is the same as first rotating x, and then translating by Rv. This shows that every ellipse can be obtained from one in the standard position by either a rotation followed by a translation, or a translation

followed by a rotation. In developing a general equation for ellipses, we will use rotation and then translation.

Rotation counterclockwise about the origin through an angle α carries (x, y) to $(x \cos \alpha - y \sin \alpha, y \cos \alpha + x \sin \alpha)$ (derived here). The inverse operation can be obtained by rotating through $2\pi - \alpha$, and hence carries (x, y) to $(x \cos \alpha + y \sin \alpha, y \cos \alpha - x \sin \alpha)$. Applying the methods of equation of a transformed ellipse now leads to the following equation for a standard ellipse which has been rotated through an angle α.

$$\frac{(x\cos\alpha + y\sin\alpha)^2}{a^2} + \frac{(x\sin\alpha - y\cos\alpha)^2}{b^2} = 1 \tag{2.8}$$

Expanding the binomial squares and collecting like terms gives

$$\left(\frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}\right)x^2 - 2\cos\alpha\sin\alpha\left(\frac{1}{a^2} - \frac{1}{b^2}\right)xy + \left(\frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}\right)y^2 = 1 \tag{2.9}$$

which is in the form $Ax^2 + Bxy + Cy^2 = 1$, with A and C positive. In this way we see that the equation for a rotated ellipse, centered at the origin is a quadratic with a nonzero xy term.

We have seen that a rotated ellipse, centered at the origin, is always given by an equation of the form $Ax^2 + Bxy + Cy^2 = 1$, where A and C are positive, and $B^2 - 4AC < 0$. To complete the analysis of the general equation of an ellipse, note that translating a curve by a fixed vector (h, k) simply has the effect of replacing x by $x - h$ and y by $y - k$ in the equation for that curve. Accordingly, the general equation for a rotated ellipse centered at (h, k) has the form $A(x - h)^2 + B(x - h)(y - k) + C(y - k)^2 = 1$, again where A and C are positive, and $B^2 - 4AC < 0$. Note also that

expanding the general form of the translated ellipse will introduce, for the first time, x and y terms. In fact the expanded version is

$$Ax^2 + Bxy + Cy^2 - (2Ah + kB)x - (2Ck + Bh)y + (Ah^2 + Bhk + Ck^2 - 1) = 0. \quad (2.10)$$

So the five parameters of an ellipse are axes (a, b), ellipse center (h, k), rotated angle α.

CHAPTER 3

Related Research

In this section, we first give an explanation of eye-model in brief. Then, we introduce the related works focused on gaze estimation. Last, the related works about eyeball center calibration and iris fitting related to the remote camera approach and the head-mounted approach are introduced.

3.1 Eye model

Figure 3.1 shows a simple eye-model of humans [5]. The eyeball consists of two spheres of different sizes. The anterior smaller sphere is the cornea, which contains the iris. As a result of human biological structure, the real gaze direction is not the same as the direction that passes through the eyeball center C and the iris center P. Generally, we take the real gaze direction to be the visual axis from corneal center C_0 to gaze point G. The theoretical direction is called the optical axis, and these two axes form a constant angle θ. K is the distance between the eyeball and iris centers, and K_0 is the distance between the eyeball and corneal centers. Moreover, in three-dimensional (3D) space, because the cornea resembles a ball, there should be another internal iris center P_0 beside the iris center P, as shown in Fig. 3.1. The iris plane is the cross section of the cornea perpendicular to the optical axis and through P_0. In this thesis, we call P_0 the internal iris center to distinguish it from the iris center P. L is the distance between the eyeball and internal iris centers.

Note that the optical axis is the normal vector of the iris plane. If we know the optical axis, the visual axis corresponding to gaze direction can be determined using human's biometric parameters.

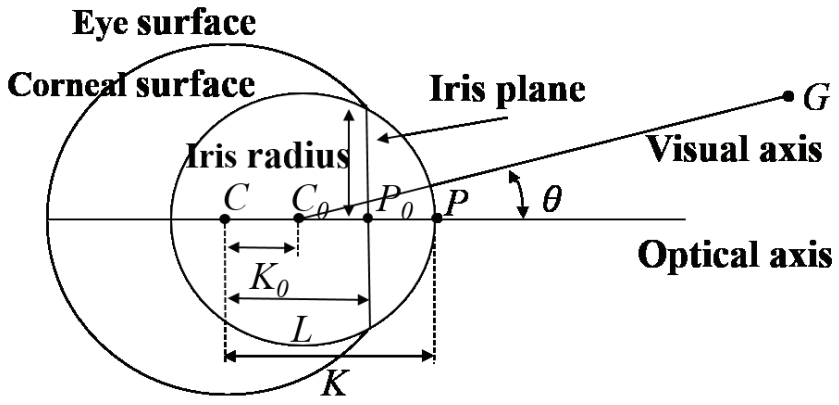


Fig. 3.1: 3D eye model. The optical and visual axes maintain a constant geometric relationship. The optical axis can be determined from eyeball center C and iris center P. The visual axis can be calculated from the known optical axis, as the eye parameters are known.

3.2 Gaze estimation

Gaze estimation methods can be divided into feature-based and appearance-based ones. Appearance-based methods do not extract features explicitly, but learn a direct mapping from high-dimensional eye images to the low-dimensional space of gaze coordinates [6, 7, 8]. *Lu et al.* [8] introduced an accurate adaptive linear regression method for mapping from sparsely collected training samples, but the method requires a fixed head pose. To solve the appearance-based gaze estimation problem under free head motion, *Lu et al.* [7] decompose the problem into sub-problems, including initial estimation under fixed head pose and subsequent compensations for estimation biases. Each sub-problem is solved using either a learning-based method or geometric calculation. However, appearance-based methods require large sets of training data to deal with eye image variations when learning a general mapping, owing to differences in appearance, illumination, head pose, scale, and eyelid movement. *Noris et al.* [9] used specialized head-mounted hardware to track the gaze in unconstrained environments. *Zhu et al.* [10] constructed a highly nonlinear generalized gaze mapping function that accounts for head movement by using support vector regression. These methods require an additional device.

Unlike appearance-based methods, feature-based methods extract features such as corneal infrared reflections, pupil center, and iris contour [11, 12, 13]. These features are used to set up a 3D eye model and then estimate the gaze direction. Feature-based methods can be highly accurate under free head movement. However, this kind of method has the disadvantage that special cameras or lights are required to extract eye features [14, 15]. One or more infrared lights are used to illuminate the

eye region and to build the corneal refection on the corneal surface, while one or more cameras are used to capture the image of the eye [5]. Instead of using infrared lights and their corneal reflections, *Ishikawa et al.* [16] used a global head model to track the entire head and eye contour, but they used template matching to find the iris and refined the iris location by using an ellipse fitting algorithm. *Chen et al.* [4] proposed a 3D eye gaze estimation and tracking algorithm based on facial feature tracking using a single camera, but they labeled the iris center manually.

With respect to RGB-D camera-based gaze estimation, Refs. [17, 18] combine a Kinect camera with another high-resolution camera, and Refs. [19, 20] use a single Kinect camera. Ref. [19] uses an appearance-based method to learn a direct mapping from the eye image to gaze parameters, with the error of gaze direction reported to be more than 20° for a moving target under extreme head movements. Ref. [20] is regarded as a hybrid approach in the sense that an eye model is incorporated into the learning process, with the average gaze direction error reported to be 3.4° in an experiment with people looking at screen targets.

Since gaze estimation aims to measure the 3D direction of the visual axis of the eyeball, which is determined from eyeball and iris centers, as shown in Fig. 3.1, we believe that the eye-model-based method is a good choice, if we can calibrate the related eye model parameters accurately. With the appearance-based method, the mapping between the eye region image and the gaze direction is learned by training from samples. Gaze extrapolation must be conducted, and over-fitting can occur during the training phase.

3.3 Remote camera approach

When a remote camera is used, users' head moves freely at the camera coordinate system. To estimate the optical axis of eyeball, we need to estimate the center position of eyeball continuously. Conventional methods calibrate the eyeball center in head coordinate system, and then estimate the head pose continuously. Iris fitting is also challenging because of the image's low quality.

Xiong et al. [3] use a simple onetime calibration procedure to obtain the eyeball center. Nine points are predefined on the monitor screen to be looked at, the calibration is achieved by minimizing the sum of angles between predicted gaze direction and the ground truth. Although nine points pattern method is a very common way for eyeball calibration, there are various principles in different researches. *Chen et al.* [4] compute the 3D position of the eyeball center based on the middle point of two eye corners. While *Li et al.* [31] use the principle that the optical axis and visual axis have a fixed angle. Besides nine points pattern method, there is a more convenient method that does not need any offline step. *Wang et al.* [24] make use of the observation that the iris contour while being a circle in 3D is an ellipse in the image, the eyeball center can be estimated from the ellipse/circle correspondence, but this method requires an accurate iris contour detection. About iris fitting in less unconstrained environments, there are various methods are proposed. *Mahadeo et al.* [21] propose a region based segmentation method for accurate eyelid detection in images with variable illumination and significant blur. Eyelashes are divided into two categories and eliminated. *Du et al.* [22] employ a coarse-to-fine segmentation scheme to improve the overall efficiency, uses a direct least squares fitting of ellipse

method to model the deformed pupil and limbic boundaries, and develops a window gradient-based method to remove noise in the iris region.

3.4 Head-mounted camera approach

Different from a remote camera, a head-mounted eye camera has a direct view of eye and no head pose is needed. Given a set of iris contour candidate feature points, we need to find the best fitting ellipse from the eye image.

To date, extensive research has been performed on this topic. The least squares fitting of an ellipse is a common choice [23], but gross errors made in the feature detection stage can strongly influence the accuracy of the results. Therefore, a more efficient fitting algorithm is proposed. *Li et al.* [2] fit an ellipse to a subset of the detected edge points using the random sample consensus (RANSAC) paradigm [25]. The best fitting parameters are then used to initialize a local model-based search for the ellipse parameters that maximize the fit to the image data. So far, the ellipse fitting method [2] is considered to be one of the most efficient methods and is applied in many studies. *Zhang et al.* [26] explore a new networking mechanism using smart glasses, through which users can express their interest and connect to a target simply by a gaze. Moreover, *Takemura et al.* [27] present more appropriate information about a persons' gaze when moving over a wide area and include visualization of the scan paths when the user with a head-mounted device makes natural head movements.

CHAPTER 4

Gaze estimation from remote camera

In this chapter, we describe the proposed method of gaze estimation from a remote camera, using a RGB-D camera, Kinect.

Our method's basic principle is similar to that of [31] and [28]. These studies track the iris center and calibrate the personalized eyeball center. Our method has these characteristics:

- The method is simple, based only on the eye model and known head pose.
- We introduce a method of eyeball center calibration by gazing at the camera center from different directions (Fig. 4.1b). Traditionally, a common point is used as a target for data training or parameter calibration (Fig. 4.1a). With our approach, the problem of the face being

(a) **(b)**

Fig. 4.1: a) Gazing at a target for data training or parameter calibration. b) Gazing at the center of the RGB camera for data training or parameter calibration.

occluded by the target during eyeball center calibration can be eliminated.

- We estimate the iris center by fitting an ellipse on the RGB image using the eye model and coordinate transformation.

With Kinect, we can build a head coordinate system (Fig. 4.2a), using an iris-center tracking algorithm [29] to obtain the 2D position of the iris center (Fig. 4.2b) and fit the iris edge on images to obtain a more accurate iris center (Fig. 4.2c). A calibration method is used to determine the eyeball center position (Fig. 4.2d). As the iris and eyeball centers are both known, the gaze direction can be estimated (Fig. 4.2e). The following sections describe the details of each step.

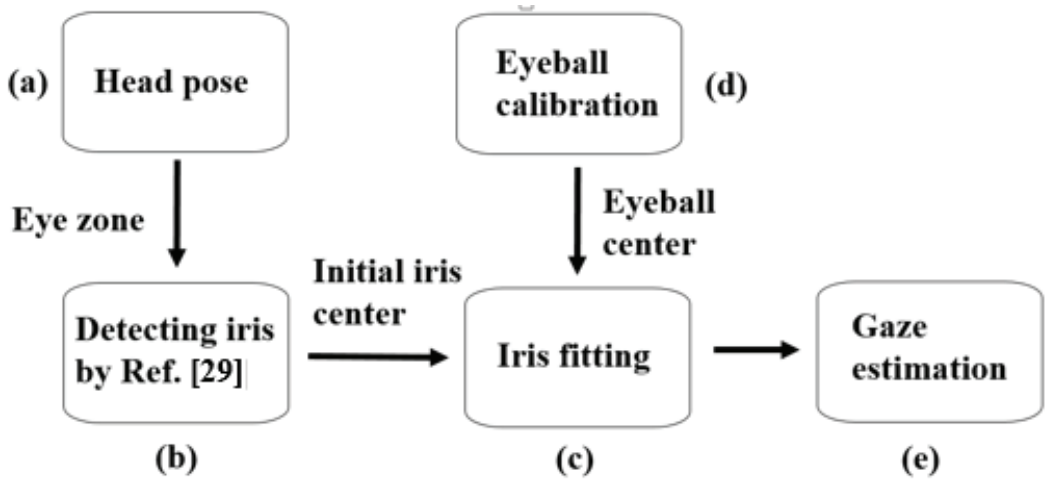


Fig. 4.2: Main steps of method. a) Achieving head pose and facial feature points by Kinect, the eye zone can also be located. b) The initial position of iris center using the method in [29], and the 2D iris center would be transformed to 3D space by a model. c) From the eye-zone image, edge points of iris would be detected, and iris center position can be obtained by the proposed fitting algorithm. d) Calibrating the eyeball center in head coordinate system based on 3D eye model, we take the RGB camera of Kinect as a target for gazing at from different directions, as the fixed angle relationship, eyeball center can be calibrated. e) Gaze estimation can be calculated in real-time.

4.1 The RGB-D camera--Kinect

Kinect is an input device for motion sensing, which is produced by Microsoft for the Xbox 360 video game console and Windows PCs (Fig. 4.3). Based around a webcam-style add-on peripheral for the Xbox 360 console, it enables users to control and interact with the Xbox 360 without the need to touch a game controller, through a natural user interface using gestures and spoken commands.

Microsoft released Kinect software development kit (SDK) for Windows. This SDK will allow developers to write Kinect enabled apps in C++/CLI, C#, or Visual Basic .NET.

Fig. 4.3: Kinect sensor.

4.1.1 The sensor

The Kinect sensor is connected to a small base with a motorized pivot and is designed to be positioned lengthwise above or below the video display. The device has two versions i.e. Kinect for Xbox 360 and Kinect for Windows (for commercial purpose).

The device features (Fig. 4.4)

- RGB camera.
- Depth sensor (IR).
- Multi-array microphone.
- Motor to adjust camera angle.

In addition to the above features, Kinect for Windows offer few extra features i.e.

- Facial recognition

  enables to track multiple points in your face like Skeleton Tracking.
- Near Mode

  enables the camera to see objects as close as 40 centimeters in front of the device without losing accuracy or precision, with graceful degradation out to 3 meters.
- Seated or 10 Joints Mode

  skeletal tracking which provides the capability to track the head, neck and arms of either a seated or standing user.

4.1.2 RGB camera

The default RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter, but the hardware is capable of resolutions up to 1280×960 (at a lower frame rate) and other formats such as UYVY.

4.1.3 Depth sensor (IR)

The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions. The sensing range of the depth sensor is adjustable, and the Kinect software is capable of automatically calibrating the sensor based on gameplay and the player's physical environment, accommodating for the presence of furniture or other obstacles.

The monochrome depth sensing video stream is in VGA resolution (640 × 480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity. The Kinect sensor has a practical ranging limit of 3.9 – 11 ft. distance when used with the Xbox software.

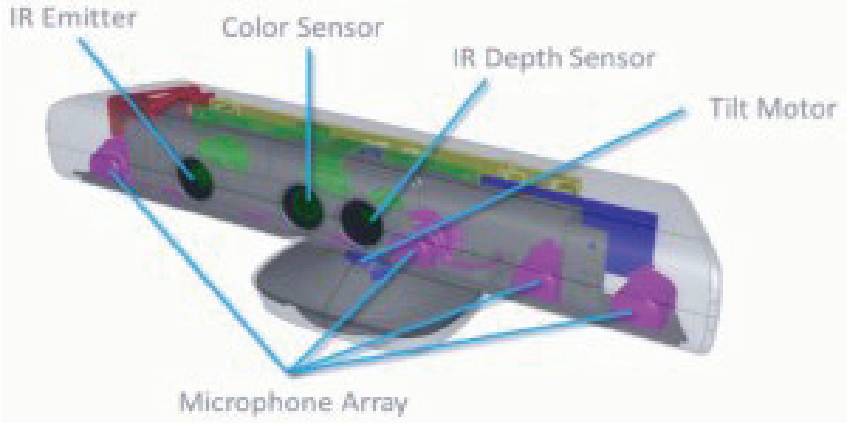


Fig. 4.4: Kinect structure.

4.1.4 Field view

The area required to play Kinect is roughly 6 m^2, although the sensor can maintain tracking through an extended range of approximately 2.3 – 20 ft.

The horizontal field of the Kinect sensor at the minimum viewing distance of ~0.8 m (2.6 ft.) is therefore ~87 cm (34 in), and the vertical field is ~63 cm (25 in), resulting in a resolution of just over 1.3 mm (0.051 in) per pixel.

4.1.5 Microphone array

The microphone array features four microphone capsules and operates with each channel processing 16-bit audio at a sampling rate of 16 kHz.

4.1.6 Face tracking and 3D head pose

The X,Y, and Z position of the user's head are reported based on a right-handed coordinate system (with the origin at the sensor, Z pointed towards the user and Y pointed up – this is the same as the Kinect's skeleton coordinate frame). Translations are in meters. The user's head pose is captured by three angles: pitch, roll, and yaw (Fig. 4.5).

The Face Tracking SDK tracks the eighty-seven 2D points indicated in the following image (Fig. 4.6)

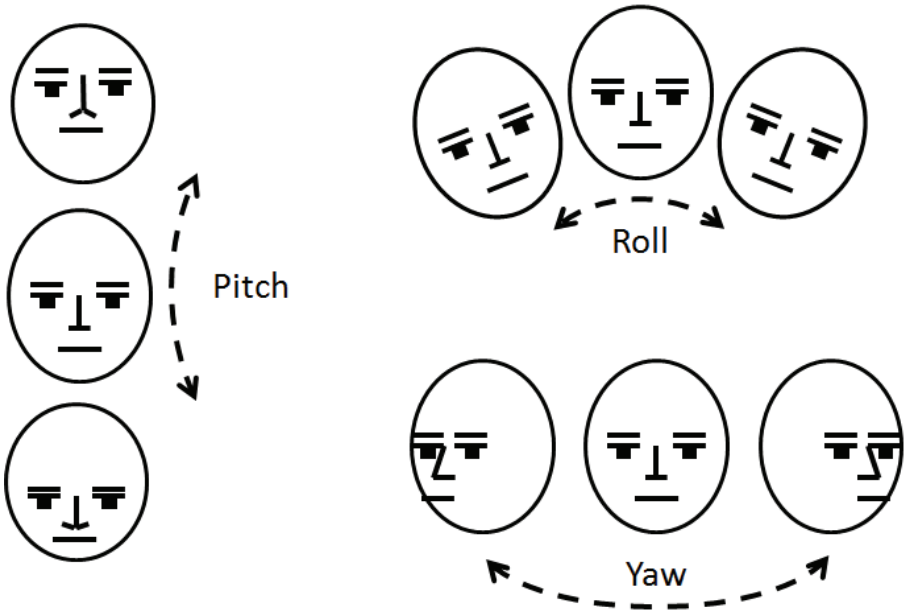


Fig. 4.5: Head pose.

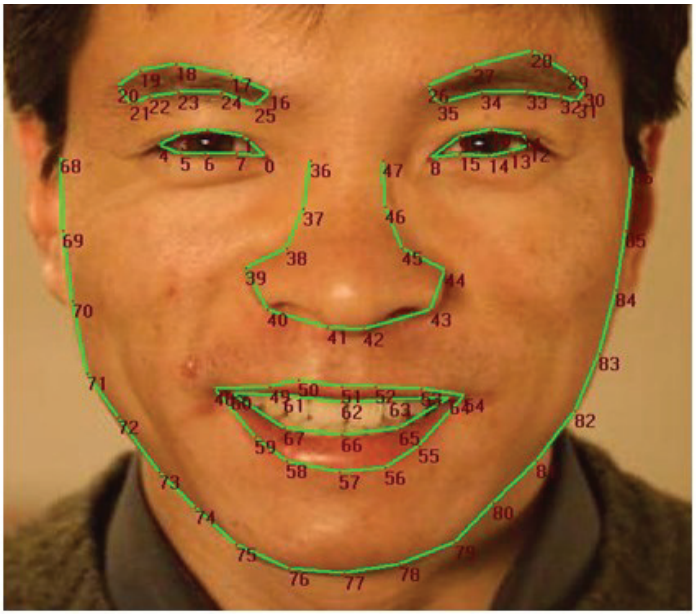


Fig. 4.6: Face tracking.

4.2 Coordinate systems

There are four coordinate systems in our method, which correspond to Kinect, head, eyeball and iris, as shown in Fig. 4.7. The coordinate system of the Kinect RGB camera is selected as the Kinect coordinate system, and the Kinect RGB camera is calibrated in advance.

Kinect coordinates are supplied by the Microsoft Face Tracking SDK for Kinect, which enables applications that track human faces in real time. The face-tracking engine of the Face Tracking SDK analyzes input from a Kinect camera, deduces head pose and facial expressions, and provides that information to an application in real time. Our method uses the Kinect face-tracking algorithm because of its reliability and convenience.

The Kinect coordinate system is based on a right-handed coordinate system whose origin is at the RGB camera sensor, with Z_C pointing toward the user and Y_C pointing upward. The Kinect SDK can supply the head pose, which is captured by three angles: pitch, roll, and yaw. We can calculate the rotation matrix from these angles. As we know the translation and rotation matrices $\boldsymbol{T}$ and $\boldsymbol{R}$, respectively, we can build a head coordinate system that is also based on a right-handed principle. Z_H points to the back of the head, while Y_H points upward. The origin of the system is inside the head, as defined by the SDK. The other two coordinates are in the eyeball coordinate and iris coordinate systems.

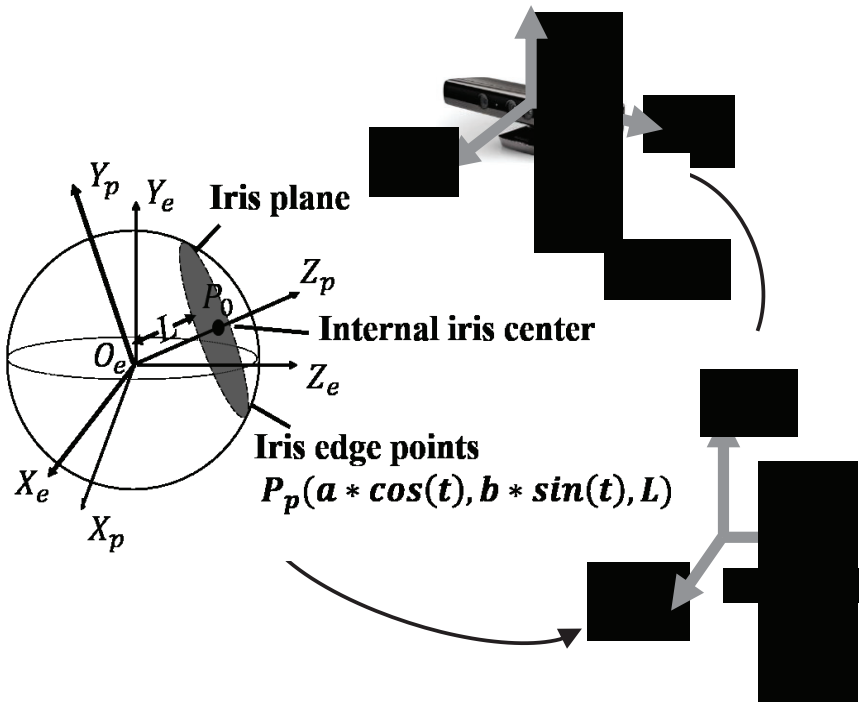


Fig 4.7: Four coordinate systems: Kinect coordinate, head coordinate, eyeball coordinate, and iris coordinate systems.

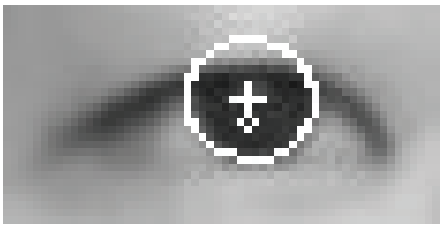
4.3 Initial iris center detection

Before fitting the iris, we use the algorithm of [29] to achieve the initial iris center position on the image. As the face-tracking algorithm offered by Kinect can detect eye contours, we can obtain an initial eye region. Based on that region, the iris center-detecting algorithm is used, and an initial center is achieved. The accuracy of the algorithm in [29] decreases when the iris is partially occluded by eyelids. As the small white circle in Fig. 4.8(a) shows, the algorithm failed in locating the iris center accurately.

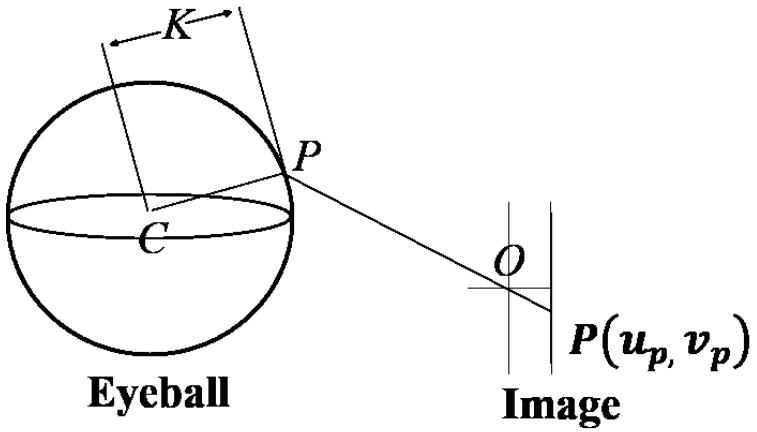
As the initial iris center is $\boldsymbol{p} = (\boldsymbol{u_p}, \boldsymbol{v_p})$ on the image, its 3D position $\boldsymbol{P} = (\boldsymbol{x_p}, \boldsymbol{y_p}, \boldsymbol{z_p})$ can be estimated using a geometric principle, assuming that the distance between iris center $\boldsymbol{P}$ and eyeball center $\boldsymbol{C}$ is a constant $\boldsymbol{K}$ because we can approximately take the eyeball to be a standard ball, with eyeball center $\boldsymbol{C}$ as its center and iris center $\boldsymbol{P}$ on its surface. On the other hand, if we know the internal parameters of the camera, a ray in 3D space passing through the 2D iris center point of the image also passes through the 3D iris center $\boldsymbol{P}$, and the ray intersects the 3D eyeball (Fig. 4.8(b)). Related equations are as follows:

$$\begin{cases} \frac{x_p}{u_p - u_0} = \frac{y_p}{v_p - v_0} = \frac{z_p}{f} \\ \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2 + (z_p - z_c)^2} = K \end{cases} \tag{4.1}$$

where $(\boldsymbol{u_0}, \boldsymbol{v_0})$ is the center of the image and f is the focal length. Eyeball center $\boldsymbol{C} = (\boldsymbol{x_c}, \boldsymbol{y_c}, \boldsymbol{z_c})$ can be estimated using calibration.



(a)



(b)

Fig 4.8: a) The small white circle inside the iris is the iris center detecting result of [29], with the detecting region determined by Kinect. It is the initial value, when fitting the iris. The white cross is the fitting result obtained using our method. b) According to the imaging principle, the ray passing through the 2D iris center on the image from the camera center also passes through the 3D iris center in space, and the 3D iris center is located on the surface of the eyeball. If the eyeball center is known, the intersection can be calculated. The lower right corner is the image on the Kinect camera screen.

4.4 Fitting the iris

We calculate the iris center by fitting the iris because it is more accurate and stable. After projection, the iris, whose shape is nearly a circle in 3D space, can present as different ellipses on images. Here, we show that by using cues provided by Kinect, the ellipse corresponding to the iris contour in a Kinect RGB image can be described using two parameters.

Let the iris and eyeball coordinate systems $\boldsymbol{O_p - X_pY_pZ_p}$ and $\boldsymbol{O_e - X_eY_eZ_e}$, respectively, be constructed as in Fig. 4.7. Note that these coordinate systems have the same origin.

In the iris coordinate system, the Z_P axis passes through the iris center perpendicular to the iris plane. According to prior knowledge, the iris in 3D space is nearly a circle. For 3D iris edge points $\boldsymbol{P_p(a * cos(t), b * sin(t), L)}$, where $\boldsymbol{a}$ is the transverse radius, $\boldsymbol{b}$ is the longitudinal radius, $\boldsymbol{L}$ is the distance between eyeball center and $\boldsymbol{P_0}$ is the internal iris center. In Fig. 3.1, $\boldsymbol{L}$ is not the same as $\boldsymbol{K}$, and $\boldsymbol{t}$ is a parameter. Transforming $\boldsymbol{P_p}$ to iris edge points in the eyeball coordinate system $\boldsymbol{P_e}$:

$$P_e = R_p * P_p \tag{4.2}$$

When the iris is rotating on the eyeball surface, the roll angle will not change relative to the eyeball; hence we can express $\boldsymbol{R_p}$ with roll, pitch, and yaw, while the roll is zero.

Then, we transform them to the head coordinate system. The axis of the eyeball coordinate system is set to the same directions as the head coordinate system, meaning that relative to the head coordinate system, the eyeball coordinate system will have translation but no rotation.

$$P_h = P_e + T_e \tag{4.3}$$

where $\boldsymbol{T_e}$ is actually the eyeball position, which can be calibrated to the head coordinate system.

Next, we transform points to the Kinect coordinate system.

$$P_c = R * P_h + T \tag{4.4}$$

After obtaining the 3D points $\boldsymbol{P_c}(\boldsymbol{x_c}, \boldsymbol{y_c}, \boldsymbol{z_c})$, we project them back to the 2D image. With Equations (4.2-4.4), the image points $\boldsymbol{I_p}(\boldsymbol{u}, \boldsymbol{v})$ can be expressed as:

$$I_p = M * P_c \tag{4.5}$$

where M is the internal camera parameters determined by chessboard calibration. Taking all of these equations and parameters, we can express Equation (4.5) as:

$$\begin{cases} u = f(\sin(t), \cos(t), pitch, yaw, R, T, a, b, L, M, T_e) \\ v = g(\sin(t), \cos(t), pitch, yaw, R, T, a, b, L, M, T_e) \end{cases} \tag{4.6}$$

And from Equation (4.6), we can obtain:

$$\begin{cases} \sin(t) = h(u, v, pitch, yaw, R, T, a, b, L, M, T_e) \\ \cos(t) = k(u, v, pitch, yaw, R, T, a, b, L, M, T_e) \end{cases} \tag{4.7}$$

Since $sin^2(t) + cos^2(t) = 1$, the final objective function is

$$\Psi(u, v, pitch, yaw, R, T, a, b, L, M, T_e) = 0 \tag{4.8}$$

Note that among the parameters of Equation (4.8), only pitch and yaw are unknowns, thanks to the head pose detection function of Kinect. This is a much simpler representation than the conventional ellipse representation with five parameters (The mathematic description can be found in the Appendix of this chapter).

Since Kinect SDK can offer the position of eye corners, we can obtain an eye mask image without eyelids. Then, a set of edge points can be detected from the image by the Canny edge detector. Moreover, the initial 3D iris center $\boldsymbol{P}$ introduced in

Section 4.3 can be obtained; hence, a credible initial pitch and yaw are available for iterations. This function can be solved using the Levenberg–Marquardt algorithm (LMA).

For the known edge points, there are outliers existed. To eliminate these outliers and increase calibration accuracy, random sample consensus (RANSAC) is used prior to LMA. After pitch and yaw are determined, the iris center ***P*** in the Kinect coordinate system can be obtained gradually from ***(0, 0, K)*** in the iris coordinate system using Equations (4.2-4.4). In this way, the accuracy of the iris center, the white cross shown in Fig. 4.8(a), can be improved.

4.5 Eyeball center calibration

The eyeball center can be regarded as fixed in the biological structure of the human head. In the proposed method, we calibrate the eyeball center position in the head coordinate system first and need to do that only once. Then, we transform it to the Kinect coordinate in real time.

To conduct the calibration, we use the 3D eye model. As Fig. 3.1 shows, when people are gazing at the target, there is a fixed angle $\boldsymbol{\theta}$ between visual and optical axes [5]. The visual and optical axes are expressed as the vectors $\overrightarrow{\boldsymbol{C_0G}}$ and $\overrightarrow{\boldsymbol{C_0P}}$, respectively. The two vectors are related to the position $\boldsymbol{C_0}$, which can be expressed by the eyeball center $\boldsymbol{C}$. By using the iris fitting method which is introduced in section 4.4, the 2D iris center can be obtained. Next we obtain an accurate $\boldsymbol{G}$ point. Figure 4.1b illustrates how to calibrate the eyeball center. During the calibration, the observer keeps gazing at the Kinect RGB camera from different directions. In this case, the coordinate of the target point $\boldsymbol{G}$ must be the origin in the Kinect coordinate system.

For the equation, first assume that the eyeball center in the head coordinate system is $\boldsymbol{T_e(x, y, z)}$. As a result of the calculation performed in the Kinect coordinate system, $\mathbf{T_e}$ must also be transformed to Kinect coordinates using Equation (4.9):

$$C = R * T_e + T \tag{4.9}$$

As Equation (4.1) describes, the 3D iris center $\boldsymbol{P}$ can be deduced from the 2D iris center from unknown $\boldsymbol{C}$:

$$P = f(C) \tag{4.10}$$

Since $\boldsymbol{K}$ and $\boldsymbol{K_0}$ are constants [5], $\boldsymbol{C_0}$ can be estimated as follows:

$$C_0 = C + \frac{K_0}{K}(P - C) \tag{4.11}$$

Then, according to a relationship between the two vectors, we obtain the following equation:

$$\frac{\overrightarrow{C_0G} \cdot \overrightarrow{C_0P}}{\| \overrightarrow{C_0G} \| \| \overrightarrow{C_0P} \|} = \cos\theta \tag{4.12}$$

Finally, the only unknown parameter is $\boldsymbol{T_e}$. To solve the nonlinear equation, we use LMA and RANSAC.

4.6 Gaze estimation

After calibration, the gaze direction can be estimated automatically in real time. First, we transform the calculated eyeball center to the Kinect coordinate system using the calculated rotation and translation matrices. The 3D eyeball position can be obtained in the Kinect coordinate system at this time and frame. The 3D position of the iris center $\boldsymbol{P}$ is obtained by fitting. Then, the eyeball center $\boldsymbol{T_e}$ is calibrated. Thus, we can calculate the optical axis frame by frame. The direction of the gaze $\boldsymbol{g}$ can be estimated and expressed as horizontal and vertical angles $(\boldsymbol{\delta}, \boldsymbol{\varphi})$. Finally, the visual axis can be obtained by adding the constant angle values $(\boldsymbol{\delta_e}, \boldsymbol{\varphi_e})$ (Fig. 4.9).

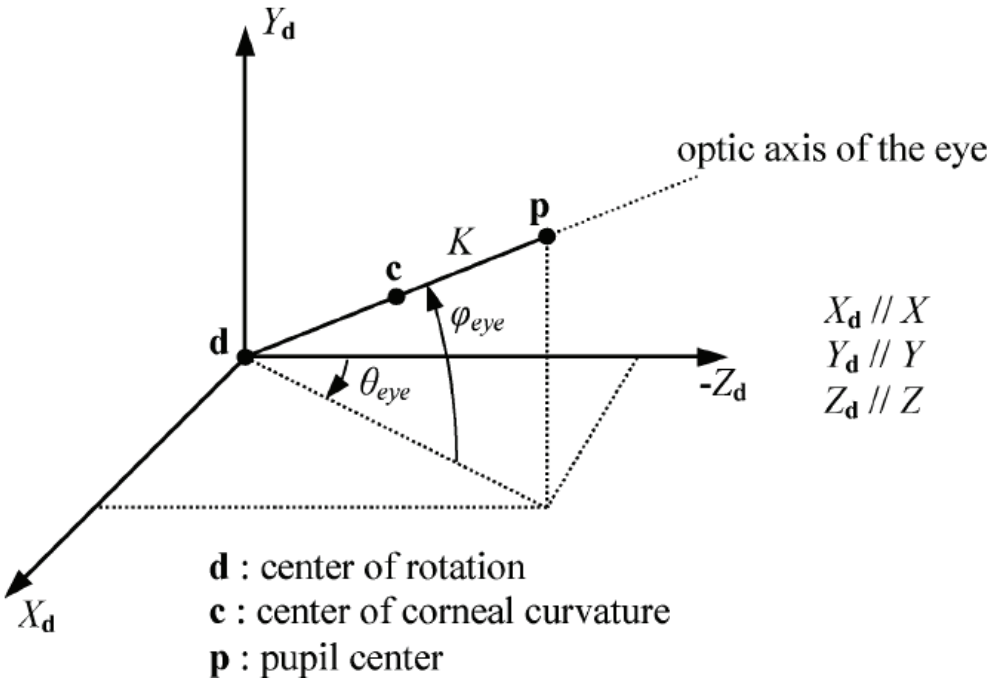


Fig. 4.9: Orientation of the optic axis of the eye (defined in Fig. 2 as in [5]).

4.7 Error analysis

To determine the gaze estimation errors corresponding to iris center errors, we first need to know the real distance that one pixel indicates. Assuming that the eyeball center is accurately estimated, the distance between Kinect and the person is $\boldsymbol{d}$, and the focal length is $\boldsymbol{f}$, one pixel indicates a distance of $\boldsymbol{d/f}$ in the real situation resulting from the proportional relation. If the detecting result has an n-pixel error, then the real distance error is $\boldsymbol{n * d/f}$. As the eyeball radius is $\boldsymbol{K}$, the gaze estimation angle errors can be expressed provably as follows:

$$\theta = arctan\,(n * d/f * K) \tag{4.13}$$

The gaze estimation errors with pixel errors at different distances are shown in Fig. 4.10: $f = 1033$, $K = 1.31$ cm. Since Kinect cannot detect the human structure if the distance between the human and Kinect is less than 50 cm, the analysis range is from 60 cm to 100 cm. Taking 60 cm as an example, when the iris detecting error is one pixel, the final gaze estimation has a 2.5-degree error. If the final gaze estimation error increases rapidly with increasing pixel error, then it would have a 12-degree error, when there was a five-pixel error in iris detection.

The error analysis yields three conclusions about the gaze estimation error:

In terms of (4.13), it is proportional to the iris center detecting error, as n increases,

In terms of (4.13), it is proportional to the distance between the Kinect and human, as d increases,

In terms of (4.13), it is inversely proportional to the image resolution, as f decreases.

With respect to how gaze estimation depends upon head pose estimation error, Reference [30] reported a position accuracy of 3-6 mm at a distance of 1-2.5 m when tracking the face by Kinect SDK. Thus, the additional error due to head estimation accuracy when the user is at 60cm would be 0.57 degrees or less.

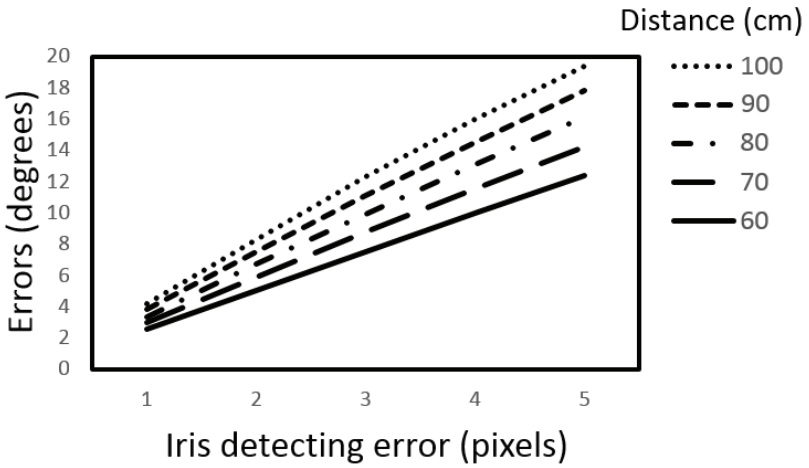


Fig. 4.10: Gaze estimation errors when the distance changed or iris detecting errors occurred.

4.8 Evaluation

A series of experiments was conducted to verify the effects of the proposed method. The known eye parameters used in these experiments are the average human eye values [5] shown in Table 4.1.

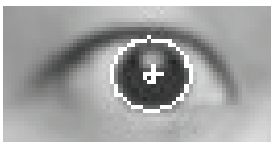
The experiments are conducted on a 1,280 × 960 RGB image from Kinect. The distance from camera to the subjects is approximately 60 cm. We realized this method using Visual C++ on a 2.5GHz Inter(R) Core(TM) i5-2400S processor and a 8GB RAM.

TABLE 4.1
VALUES OF THE EYE PARAMETERS

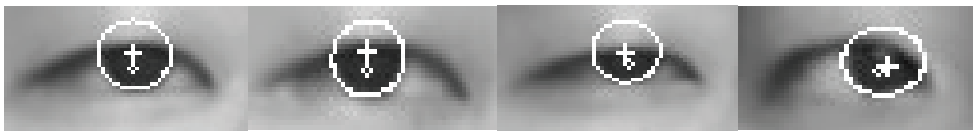
Parameter	Description	Value
K	Distance between the center of the eyeball and the center of iris.	13.1 mm
L	Distance between the center of the eyeball and the internal center of iris.	10.5 mm
K_0	Distance between the center of the eyeball and the center of corneal.	5.3 mm
a	Transverse radius	6 mm
b	Longitudinal radius	5.5 mm
δ_e	Horizontal angle between visual and optical axis of the eye	−5° for the right eye, 5° for the left eye
φ_e	Vertical angle between visual and optical axis of the eye	1.5°

4.8.1 Iris fitting result

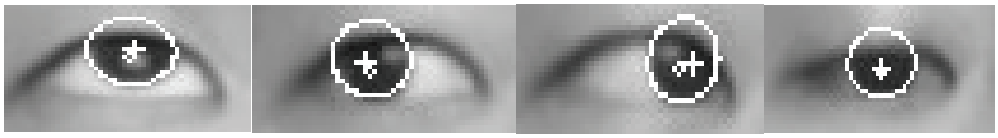
Following the fitting procedure introduced in Section 4.4, to have a direct sense after determining the yaw and pitch, we project the iris edges in the iris coordinate system to the 2D space of the image. The results show that our algorithm performs well in different conditions (Fig. 4.11). The white circle inside the iris is the initial iris center point as determined by the detecting algorithm [29], while the white cross is the final iris center point as determined by fitting. The white cross represents iris center $\boldsymbol{P}$, not the internal iris center point; hence the point is not on the iris plane but outside in 3D space. After projection of 3D iris center to the 2D image, it might not be in the very center of the iris fitting, when the person is not looking directly into the camera as Fig. 4.11 shows. The big white circle is the iris fitted using our method. If an entire iris is shown on images, then it is easy to fit (Fig. 4.11a). However, our method can fit successfully even if an iris that is not complete (Fig. 4.11b) and also irises looking upward, downward, leftward, and rightward (Fig. 4.11c). This shows that fitting performs well, if the iris edge points are detected well, and that it is more accurate and reliable than the method proposed by [29].



(a)



(b)



(c)

Fig. 4.11: Iris fitting result in different conditions

4.8.2 Eyeball center calibration

The eyeball center is fixed in the head coordinate system regardless of rotation; hence the center position must be calibrated first. We ask the subject to gaze at the RGB camera while the head is in different positions. Thus the gaze point $\boldsymbol{G}$ coincides with the RGB camera. This has the advantage that because the calibration is calculated in the Kinect coordinate system, the coordinate of $\boldsymbol{G}$ is zero.

Ten sets of data were collected. To ensure accurate calibration, the subject was provided with an iris image which is as complete as possible. Then, the data were collected with gazing directed at the camera center from different directions at different depths. After calibration, the eyeball center $\boldsymbol{T}_e$ in head coordinate system can be obtained.

4.8.3 Gaze test

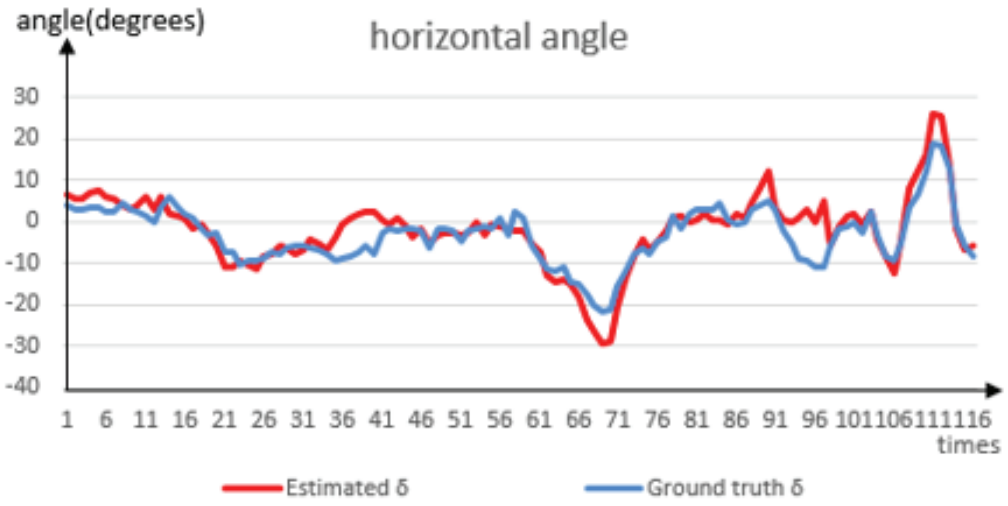
To test the accuracy of the proposed method, the subject continued to gaze at the RGB camera, while moving the head. Then, we determined two vectors, one is connecting the eyeball center and gaze point (the origin of the Kinect coordinate system) directly. The other vector is determined by the eyeball and iris center. Because the position error of eyeball center calibration is very small relative to the distance between Kinect and the subject, we take this vector as ground truth. A comparison of estimated gaze and ground truth is shown in Fig. 4.12. Gaze estimation is expressed in terms of horizontal and vertical angles($\boldsymbol{\delta}, \boldsymbol{\varphi}$). 116 frames are recorded in total, with horizontal and vertical estimations ranging from −20° to 20° and from −10° to 15° respectively (Fig. 4.12). The final average results are 3.0° and 4.5° for horizontal and vertical estimation, respectively, and most of the frames are close to ground truth.

We collected ten samples from ten people to test our algorithm. The subjects were asked to keep gazing at the camera while moving their head. The average gaze direction estimation errors are shown in Table 4.2. Sample No. 8 is for testing how the error depends on the yaw angle (looking rightward and leftward); Sample No. 9 is for testing how the error depends on the pitch angle (looking downward and upward); sample No. 10 is for testing free head movement. Large head poses are included in last three samples. As shown in Fig. 4.13a, the horizontal gaze direction error increases with yaw angle. In Fig. 4.13b, the vertical gaze direction error increases with pitch angle. With respect to head tracking accuracy with Microsoft Face Tracking SDK: “Face Tracking tracks when the user’s head yaw is less than 45

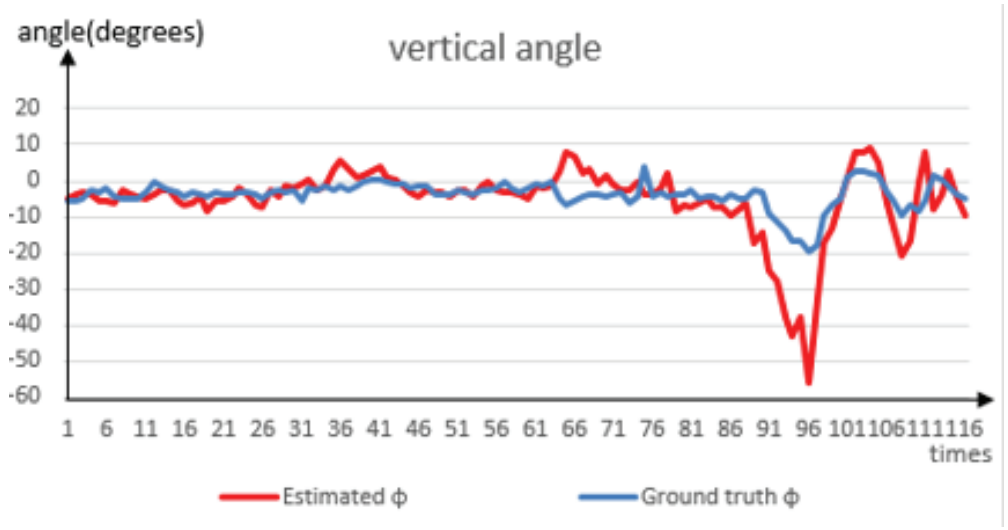
degrees, but works best when less than 30 degrees. Face Tracking tracks when the user's head pitch is less than 20 degrees, but works best when less than 10 degrees." Our experiments are roughly consistent with the above statement.

To ensure accurate calibration, the subject had no head rotation and was provided with as a complete iris image as possible. Then, the data were collected with gazing directed at the camera center from different directions at different depths. After calibration, the eyeball center $\boldsymbol{T}_e$ in head coordinate system can be obtained.

The experimental environment is almost the same as that of [31], with the only difference that subjects gazed at a moving target without their head moving, whereas our test involved gazing at a stable target (camera center) with head moving. The movement difference is relative, hence comparable in our view. The average comparison results of the first seven samples are shown in Table 4.3. Because our iris center detection is more robust to eyelid occlusion than that of [31], our method achieves 27 percent improvement in the vertical direction. The average running time for one frame during our experiment is 330 ms.

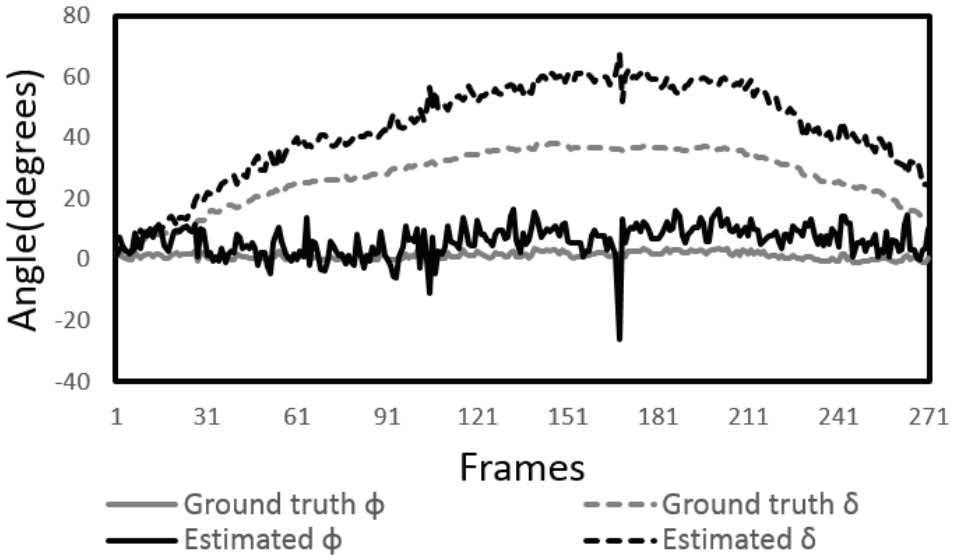


(a)

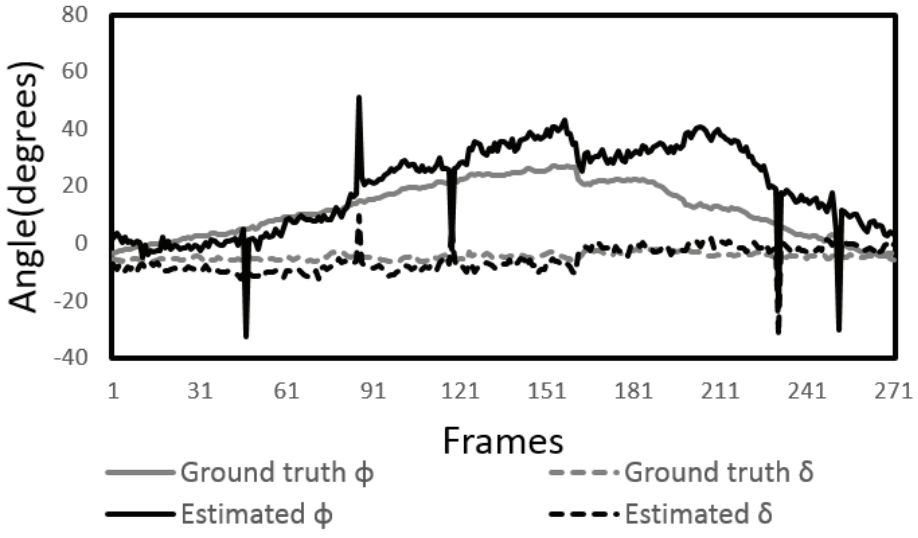


(b)

Fig. 4.12: Comparison of estimated gaze and ground truth. a) Horizontal gaze estimation. b) Vertical gaze estimation.



(a)



(b)

Fig. 4.13: The error depends on the angle. a) Yaw angle b) Pitch angle.

TABLE 4.2
GAZE ESTIMATION ERRORS

Index	Frames	Vertical angles (degrees)	Horizontal angles (degrees)	Average (degrees)
Sample 1	1000	5.5	4.3	4.9
Sample 2	1000	5.2	3.1	4.2
Sample 3	1000	9.1	3.9	6.5
Sample 4	1000	4.8	4.3	4.4
Sample 5	1000	6.9	4.1	5.5
Sample 6	1000	5.0	4.9	5.0
Sample 7	1000	5.1	6.1	5.6
*Sample 8	1000	7.4	11.4	9.4
*Sample 9	1000	8.1	3.0	5.6
*Sample 10	1000	14.1	9.5	11.8

*: These samples are testing for large head poses

TABLE 4.3
GAZE ESTIMATION COMPARING RESULT

Vertical (degrees)	Horizontal (degrees)	Vertical [31] (degrees)	Horizontal [31] (degrees)
5.9	4.4	7.5	4.4

4.8.4 Database test

We tested our method on the public database EYEDIAP [32]. One EYEDIAP session involves continuous screen target testing, with fourteen participants from different areas. A small circle was drawn on the computer screen and programmed to move along a trajectory parameterized by a quadratic Bezier curve. The control points of the curve were drawn from a uniform distribution defined within a smaller window of the screen of which position was drawn randomly. A new trajectory was redefined every 2 s, with a resolution of 640 × 480 from Kinect. The database also offers 3D eyeball center and screen target coordinates. Since the experimental subject gazes continuously at the screen target, we can take the line through the 3D eyeball center and the target as the ground truth of gaze estimation. During iris detection validation, we use the eyeball center and head pose provided by the database.

However, there are some problems we need to illustrate. These participants are completely free of head movement; hence the eye is sometimes occluded by the nose or the head pose is missing. Moreover, our iris fitting result relies on the result of iris edge detection. Due to the low illumination and resolution of the data, the iris fitting result was poor for some frames. The results for the left eye of fourteen participants are shown in Table 4.4. The second column shows video frames. The third column shows detected frames. As sometimes the head pose data are missing, the eye is occluded by the nose, no edge points are found, or eye zone location failed, our method could not perform iris edge detection on these kinds of frames. Next, the vertical and horizontal angles (all) are the average gaze estimation angle errors of detected frames. The sixth column shows filtered frames. Because we did not

eliminate eyelid interference in this database and both illumination and resolution are low, iris edge detection is challenging. As Equation (13) describes, if the iris center has a two-pixel error when the distance is 70 cm and the resolution is 640 × 480, the final gaze estimation will have an error of nearly 12-degree. Therefore, if the horizontal or vertical gaze estimation is greater than 15 degrees, we consider it an iris fitting failure and discount the frame. The vertical and horizontal angles (F) are the average angle errors of filtered frames. Iris fitting fails for most of the frames because our iris fitting method strongly relies on iris edge detection. Considering the low resolution, inaccurate iris edges, and comprehensive test cases, we believe that the average vertical and horizontal gaze errors, 7.6 and 6.7 degrees, are acceptable. Even after calculating all the detected frames, the average gaze estimation is 12.5 and 13.5 degrees for the vertical and horizontal directions, respectively. The performance of the proposed method is inferior to the state-of-the-art appearance-based method of [33], which reported average gaze error of 8.1 degrees for the same database. Considering the low resolution and illumination, this result supports the fact that an eye-model-based method is suitable for the situation where iris contours are available in contrast with the appearance-based method, which can cope with low resolution images by learning 2D eye patterns.

To validate the eyeball calibration method, we take the initial values from the eyeball center provided by the database. We choose fifteen frames that fit the iris well to calibrate the eyeball center using our method. Although the participants are not gazing at the camera center, screen target positions are provided; hence the calibration principle is nearly the same. The fourteen participant calibration results are shown in

Table 4.5. The first row of each participant shows the ground truth from the database, while the second row shows the calibration results using our method. The average errors from the ground truth are 0.54, 1.65, and 16.25 mm for the X, Y, and Z coordinates, respectively. In our opinion, the reason for inaccurate Z coordinate calibration is that during the calibration, the distance between the screen and participants is 70 cm, the free movement of the head in these videos shows no obvious depth variation, and the position of the iris center point on the image was not sensitive to the depth change of the eyeball. In other words, the estimation error of the depth of the eyeball has little influence on gaze estimation in such an experimental setup. An alternative method is to use depth cues and head pose in terms of the statistical physiological data, as the dataset did, only X and Y coordinates are obtained by calibration.

Note that the proposed method is a feature-based gaze estimation method, and theoretically, the accuracy of iris contour estimation depends on the image quality for feature-based gaze estimation methods. The results show that the proposed method works even with a small number of iris edge points.

TABLE 4.4
GAZE ESTIMATION ERRORS FOR EYEDIAP DATABASE

Participant	Frames	Detected frames	Vertical angles (all) (degrees)	Horizontal angles (all) (degrees)	Filtered frames	Vertical angles (F) (degrees)	Horizontal angles (F) (degrees)
1	4464	4151	10.9	14.4	1854	7.5	8.2
2	4456	2388	9.9	18.7	1356	5.2	7.6
3	4457	4172	10.7	12.1	2356	7.0	7.0
4	4493	2207	13.6	12.8	897	7.2	6.3
5	4457	4287	19.1	14.6	837	9.8	6.1
6	4457	4346	8.3	9.5	3189	6.0	6.5
7	4457	2782	15.4	26.8	551	7.3	7.3
8	4457	4030	9.5	11.7	2461	5.5	6.9
9	4456	3884	18.8	17.0	918	8.8	6.7
10	4491	4381	10.7	11.7	2513	7.2	6.2
11	4457	4388	15.2	9.6	1693	10.1	5.9
12	4457	4446	11.0	14.9	2009	8.4	7.9
13	4457	4449	8.9	8.0	3409	7.0	6.3
14	4457	4400	12.7	6.7	2830	9.7	5.4
Average	4462	3879	12.5	13.5	1919	7.6	6.7

TABLE 4.5
EYEBALL CENTER ERRORS FOR EYEDIAP DATABASE
(1ST ROW SHOWS GROUND TRUTH; 2ND ROW SHOWS CALIBRATION RESULTS)

PARTICIPANT	X(mm)	Y(mm)	Z(mm)
1	31.98	34.58	85.39
	31.04	41.12	93.96
2	32.02	37.19	91.37
	31.17	37.36	106.42
3	31.62	35.07	89.30
	31.21	36.30	103.49
4	30.52	34.10	84.33
	31.61	32.43	102.45
5	32.24	33.53	87.28
	31.66	31.67	105.16
6	32.36	33.10	84.76
	32.82	31.49	101.36
7	32.22	34.00	86.65
	33.01	30.64	112.01
8	30.83	32.81	82.27
	31.17	31.31	100.09
9	33.43	38.51	96.57
	32.41	38.01	110.28
10	30.31	35.05	86.61
	30.00	34.77	104.04
11	32.15	34.19	85.94
	32.20	35.52	101.63
12	31.17	33.65	85.94
	31.73	33.78	99.11
13	34.19	34.94	93.25
	34.15	35.38	107.99
14	32.17	34.73	89.02
	32.12	32.29	108.24
Average errors	0.54	1.65	16.25

4.9 Conclusions

This chapter proposed a novel method of estimating gaze direction using color information based on an eye model-given head pose. We introduced a simple calibration method for the eyeball center by gazing at the camera center and estimated the iris center by projecting the 3D contour of the iris to the RGB image using the eye model and RGB cues. In theory, we simplified the elliptic contour estimation of the iris as a representation of two unknowns. This simplified representation results in faster and more accurate gaze estimation. Moreover, our method is simple and reliable, requiring no training stage, and it can also run automatically in real time with a high-resolution image.

Finally, we summarize our work relative to the state of the art gaze estimation methods. In contrast to the eye-model-based gaze estimation method [31], which also uses a RGB-D camera as an input, the proposed two parameter iris boundary estimation method achieves much more accurate results, as shown in our subjects experiment and the public database EYEDIAP experiment. In contrast to the appearance-based gaze estimation method [33], the performance of the proposed method is inferior for lower resolution and poor illumination images, as tested on the public database EYEDIAP. This is because the eye-model-based method necessitates a certain number of correctly detected edge points of iris contour. In practice, eye-model-based or appearance-based approaches, should be used depending upon the setting.

4.10 Appendix

This part is a supplement to the functions in this chapter.

4.10.1 Eyeball center calibration problem

Assuming $\boldsymbol{C}_0(\boldsymbol{x_{c0}}, \boldsymbol{y_{c0}}, \boldsymbol{z_{c0}})$, $P(\boldsymbol{x_p}, \boldsymbol{y_p}, \boldsymbol{z_p})$, and $C(\boldsymbol{x_c}, \boldsymbol{y_c}, \boldsymbol{z_c})$.

Equation (4.12) can be expressed as follows:

$$\left(x_{c0}^2 + y_{c0}^2 + z_{c0}^2 - x_{c0} * x_p - y_{c0} * y_p - z_{c0} * z_p\right) - \sqrt{x_{c0}^2 + y_{c0}^2 + z_{c0}^2} * \sqrt{\left(x_p - x_{c0}\right)^2 + \left(y_p - y_{c0}\right)^2 + \left(z_p - z_{c0}\right)^2} * \cos(\theta) = 0 \tag{4.14}$$

According to the first formula of Equation (4.1),

$$\begin{cases} x_p = k * (u_p - u_0) \\ y_p = k * (v_p - v_0) \\ z_p = k * f \end{cases} \tag{4.15}$$

Substituting Equation (4.15) into the second formula of Equation (4.2), we take k as the unknown.

$$\left[(u_p - u_0)^2 + (v_p - v_0)^2 + f^2\right] * k^2 - 2 * k * \left[\left(u_p - u_0\right) * x_c + \left(v_p - v_0\right) * y_c + f * z_c\right] + x_c^2 + y_c^2 + z_c^2 - K^2 = 0 \tag{4.16}$$

From Equation (4.15) and Equation (4.16), we can express $\boldsymbol{P}$ as $\boldsymbol{C}$.

According to Equation (4.11),

$$\begin{cases} x_{c0} = x_c + \frac{K_0}{K} * (x_p - x_c) \\ y_{c0} = y_c + \frac{K_0}{K} * (y_p - y_c) \\ z_{c0} = z_c + \frac{K_0}{K} * (z_p - z_c) \end{cases} \tag{4.17}$$

From Equation (4.17), we can express C_0 as $\boldsymbol{C}$.

According to Equation (4.9),

$$\begin{cases} x_c = R_1 * T_e + T_1 \\ y_c = R_2 * T_e + T_2 \\ z_c = R_3 * T_e + T_3 \end{cases} \tag{4.18}$$

From Equation (4.18), we can express $\boldsymbol{C}$ as T_e.

Next by substituting Equations (4.16-4.18) into Equation (4.14), the unknown parameter in Equation (4.14) is only T_e. Since T_e is constant in the head coordinate system independent of head pose, we can solve it by solving a set of equations obtained from the observation of different head poses.

4.10.2 Iris fitting problem

Assuming $\boldsymbol{P}_h(\boldsymbol{x_{ph}}, \boldsymbol{y_{ph}}, \boldsymbol{z_{ph}})$, $T_e(\boldsymbol{x_{te}}, \boldsymbol{y_{te}}, \boldsymbol{z_{te}})$, $T(T_1, T_2, T_3)$, $\boldsymbol{P_c}(\boldsymbol{x_{pc}}, \boldsymbol{y_{pc}}, \boldsymbol{z_{pc}})$,

$R = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}$, and $R_p = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$.

According to Equation (4.2) and Equation (4.3), $\boldsymbol{P}_h$ can be expressed as follows:

$$\begin{cases} x_{ph} = r_1 * a * \cos(t) + r_2 * b * \sin(t) + r_3 * L + x_{te} \\ y_{ph} = r_4 * a * \cos(t) + r_5 * b * \sin(t) + r_6 * L + y_{te} \\ z_{ph} = r_7 * a * \cos(t) + r_8 * b * \sin(t) + r_9 * L + z_{te} \end{cases} \tag{4.19}$$

According to Equation (4.4), $\boldsymbol{P_c}$ can be expressed as follows:

$$\begin{cases} x_{pc} = R_1 * x_{ph} + R_2 * y_{ph} + R_3 * z_{ph} + T_1 \\ y_{pc} = R_4 * x_{ph} + R_5 * y_{ph} + R_6 * z_{ph} + T_2 \\ z_{pc} = R_7 * x_{ph} + R_8 * y_{ph} + R_9 * z_{ph} + T_3 \end{cases} \tag{4.20}$$

According to Equation (4.5), $\boldsymbol{I_p}$ can be expressed as follows:

$$\begin{cases} u = x_{pc} * \frac{f}{z_{pc}} + u_0 \\ v = y_{pc} * \frac{f}{z_{pc}} + v_0 \end{cases} \tag{4.21}$$

Substituting Equation (4.19) and Equation (4.20) into Equation (4.21), we see that they are linear equations. Then, $\sin(t)$ and $\cos(t)$ can be obtained from Equation (4.21). Using $sin^2(t) + cos^2(t) = 1$, we can achieve a final objective Equation (4.8). In this function, only R_p is unknown, and the rotation matrix R_p can be expressed in terms of roll, pitch, and yaw, while roll is zero, as we mentioned previously. Thus, there are only two unknowns in this problem.

CHAPTER 5

Gaze Estimation from Head-mounted Camera

The use of an eye model results in more accurate gaze estimation. Based on an eye model, the gaze detection from a head-mounted camera can be formulated as the problem of estimating the pose of the optical axis of eyeballs, including position and direction [12].

In this chapter we argue that the above processing contains with much of futility for gaze estimation system of a head-mounted eye camera. Since the eye camera is mounted on the head, as shown in Fig. 1.2, the position of eye center at the coordinate system of head-mounted eye camera does not change unless the position of eye camera is changed. Once the position of eye center is known, what we need to do is only to determine the direction vector of optical axis with two unknown parameters. In this case, the ellipse of iris contour can be represented by these two parameters, that is, we do not need to fit the ellipse of iris contour with five unknown parameters.

In this chapter, we first present a schematic block diagram of the proposed method. We then describe the algorithm used for fitting the iris contour in an image to an ellipse using two unknown parameters. Finally, we provide the algorithm employed for calibrating eyeball center position in relation to the coordinate system of the head-mounted camera.

5.1 Block diagram of the proposed method

The block diagram of the proposed method is shown in Fig. 5.1. For an image captured using a head-mounted camera, a change in the camera position was first detected. In case of a change in the camera position, the calibration of the eyeball center position is performed. Otherwise, the iris contour fitting is conducted.

The change in the camera position can be detected by finding the eye corners [34]. In this chapter, we focus on the iris contour fitting and the calibration of eyeball center position.

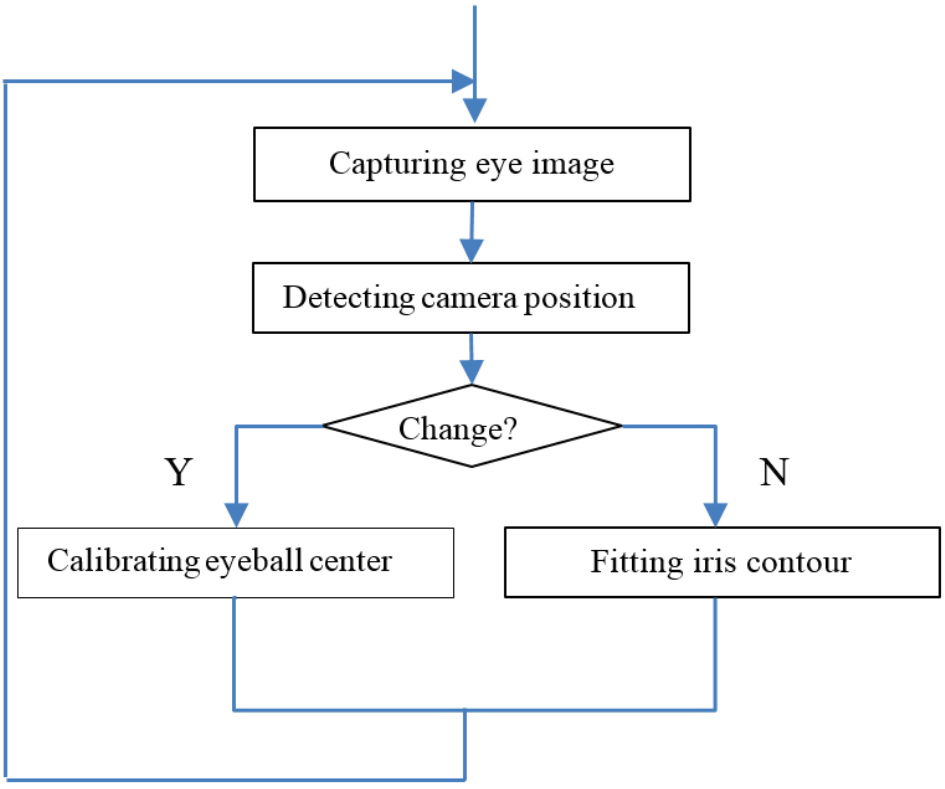


Fig 5.1: Block diagram of the proposed method.

5.2 Fitting of iris contour

Figure 5.2 shows the sketch of our computation model for gaze estimation from a head-mounted camera based on an eye-model. The big circle indicates an eyeball and the small circle on eye image indicates the iris contour. The optical axis starts from the eyeball center O_c, and passes through the internal iris center P_0. Since the position of eyeball center has three unknown parameters and the direction vector of optical axis have two unknown parameters, the problem of gaze estimation from head-mounted camera is to determine these five parameters from an eye image at the coordinate system of head-mounted camera, $O_C - X_C Y_C Z_C$.

Our algorithm for fitting the iris contour is very different from the existing fitting algorithms. In 3D space, the iris is nearly a circle. Based on the 3D iris contour, the iris coordinate system is built. In the iris coordinate system, the origin is at the 3D eyeball center, the Z axis passes through the iris center perpendicular to the iris plane (Fig. 5.2). Given the 3D iris edge points $P_p\,(r\cdot\cos(t), r\cdot\sin(t), L)$, $\boldsymbol{r}$ is the real iris radius, $\boldsymbol{L}$ is the distance between eyeball center and internal iris center P_0, and $\boldsymbol{t}$ is a parameter. The iris edge points (P_p) are then transformed to the camera coordinate system (P_c).

Assuming, $T(T_1, T_2, T_3)$, $P_c(x_{pc}, y_{pc}, z_{pc})$, $R = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}$.

$$\begin{cases} x_{pc} = R_1 * r * \cos(t) + R_2 * r * \sin(t) + R_3 * L + T_1 \\ y_{pc} = R_4 * r * \cos(t) + R_5 * r * \sin(t) + R_6 * L + T_2 \\ z_{pc} = R_7 * r * \cos(t) + R_8 * r * \sin(t) + R_9 * L + T_3 \end{cases}, \tag{5.1}$$

where $\boldsymbol{T}$ is the 3D eyeball center position and it can be calibrated in advance. The calibration procedure is presented in Section 5.3. As the iris rotates on the eyeball surface, the roll angle does not change relative to the eye camera pose. Hence, $\boldsymbol{R}$ can be expressed using roll, pitch, and yaw, when the roll is zero.

The iris edge points in the camera coordinate system (P_c) are then projected back to a 2D image. The image points $P_i(u, v)$ can be expressed as

$$\begin{cases} u = x_{pc} * \dfrac{f}{z_{pc}} + u_0 \\ v = y_{pc} * \dfrac{f}{z_{pc}} + v_0 \end{cases}, \tag{5.2}$$

where $M(u_0, v_0, f)$ is the internal camera parameter determined using a chessboard calibration. Substituting Equation (5.1) into Equation (5.2), we can express Equation (5.2) as

$$\begin{cases} u = f(\sin(t), \cos(t), R, T, r, L, M) \\ v = g(\sin(t), \cos(t), R, T, r, L, M) \end{cases}, \tag{5.3}$$

Obviously, they are linear equations. The values for sin(t) and cos(t) can be easily obtained from Equation (5.3)

$$\begin{cases} \sin(t) = h(u, v, R, T, r, L, M) \\ \cos(t) = k(u, v, R, T, r, L, M) \end{cases}, \tag{5.4}$$

Using the relation, $\sin^2(t) + \cos^2(t) = 1$, from Equation (5.4) we can have the following equation.

$$\Psi(u, v, R, T, r, L, M) = 0, \tag{5.5}$$

where only the rotation matrix $\boldsymbol{R}$ is unknown, and can be expressed in terms of roll, pitch, and yaw, when the roll is zero as mentioned previously. Thus, there are only two unknowns in Equation (5.5). Because a set of 2D iris edge points can be detected on the image, a set of over-constraint equations can be obtained from one image. The Levenberg–Marquardt algorithm can be used to solve the equation. Furthermore, RANSAC is used to cope with outliers caused by noises and edge point detection errors. Hereafter, this approach is named as the two parameters iris fitting method (TIFM).

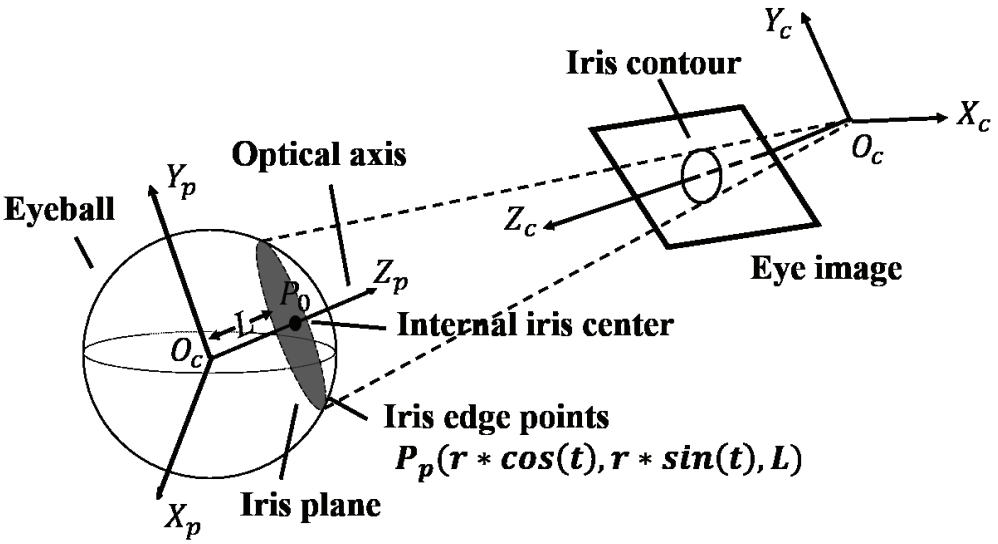


Fig. 5.2: Sketch of our computation model for iris fitting from a head-mounted camera based on an eye-model.

5.3 Calibration of eyeball center position

Because our iris-fitting algorithm is based on a 3D eye model, the estimation of the 3D eyeball center position is necessary. A head-mounted eye camera is adopted in our system, where the eyeball center position is fixed with respect to the camera coordinate system. Thus, the position of the eyeball center is constant with respect to the coordinate system of eye camera once it is calibrated.

Theoretically, the 3D iris center and the normal direction vector of iris plane can be computed as two solutions from a single eye image using the conventional five parameters iris fitting method (FIFM) [2]. For calibrating the eyeball center position, it is sufficient to identify the true one from the two solutions [12, 35]. Practically, because these parameters are computed from iris contour, fitting iris contour accurately is a crucial problem.

In this chapter, we employ the relaxation method to solve this problem. For an eye image, the iris contour is first fitted using edge points by FIFM [2] to compute the 3D iris center and the direction vector. The eyeball center can then be inferred. Next, the proposed TIFM is used to compute the projection of 3D iris contour on the eye image. Then, the 3D iris contour projection is used as cues to cluster edges points. This process is iterated until a stable solution was obtained. Empirically, a satisfactory result can be obtained over 50 iterations.

An example of the iterative solution is shown in Fig. 5.3. The final iris contour is obviously fitted better than the initial estimate. Because the 3D eyeball center position is estimated from the detected iris contour, it results in much better estimation.

The next step is to identify the true one from the two solutions computed using FIFM. Because eyeball center is constant in the eye camera coordinate systems unless we change the eye camera position, a consecutive image sequence is used to determine the truth of eyeball center.

Assume there are a set of solutions $\{P_{O1}, P_{O2}, \cdots, P_{On}\}$, for each P_{Oi}, we count the numbers that satisfy the distance from P_{Oi} to P_{Oj} is smaller than the threshold of 2mm as follows.

$$P_{Ot} = \arg\max \underbrace{\sum_{j=0}^{n} (dist(P_{Oi}, P_{Oj}) < thres)}_{i=0,1,\ldots,n} \quad (i \neq j) \tag{5.6}$$

P_{Ot} is the point, which has the maximum number of points. We then take P_{Ot} and the points around together as a cluster, and compute the average value of this cluster as the final solution of 3D eyeball center position.

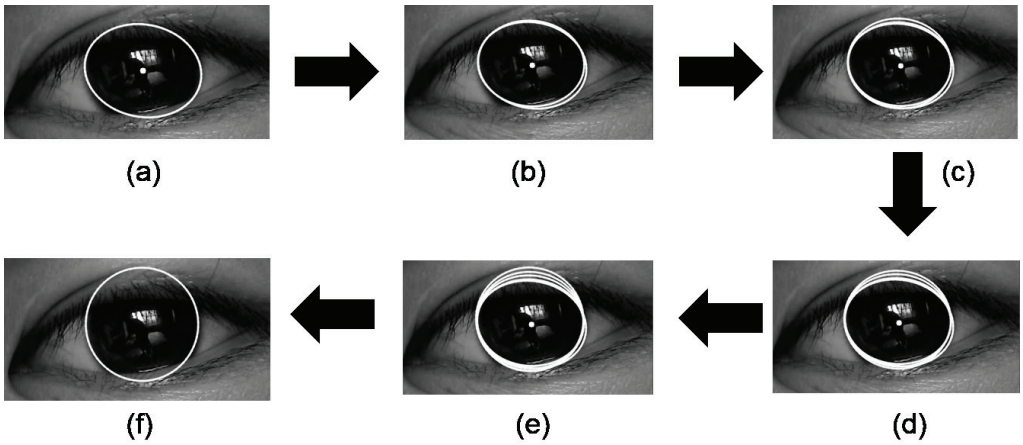


Fig. 5.3: Iris fitting by iterations. a) Initial fitting. b) 1st iteration. c) 10th iteration. d) 20th iteration. e) 30th iteration. f) Final fitting.

5.4 Evaluation

To verify the effects of the proposed method, a series of experiments were conducted. In these experiments, the images of 640×480 pixels were captured using a calibrated head-mounted camera. The eye parameters used in these experiments are derived from known average human eye values [5], iris radius r = 6mm, distance between the center of the eyeball and the internal center of iris L = 10.5mm.

5.4.1 Simulation

This section provides a simulation of the effects of edge point errors and outliers on the accuracy of fitting using FIFM [2] and the proposed TIFM. In the camera coordinate system, assuming that the eyeball center position and the iris pose are known, the iris contour, which is projected onto the image, can also be determined. The contour on the image represents the iris location. Thus, these iris edge points can be considered the ground truth.

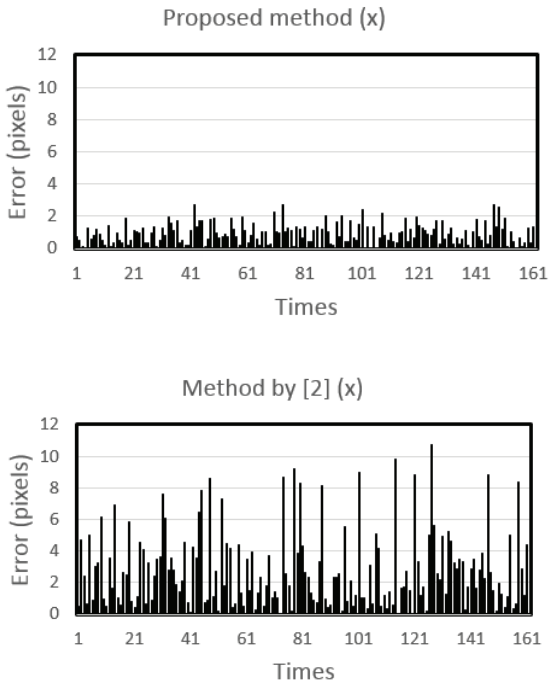
Simulation 1: To validate the robustness against edge point errors, 0–10 random pixel errors were added to the ground truth of edge points. Based on the ground truth, the iris fitting was performed using our method and FIFM, and the results were compared. Considering that different iris poses will have different projections on the image, the simulations were repeated for different situations (Fig. 5.4a). As a result, the average fitted iris center error obtained using our method is one pixel in comparison to the value of three pixels obtained using FIFM (Table 5.1). This implies that our method is more robust to noise.

Simulation 2: To validate the robustness against outliers, 0–40 random pixel errors were added to a quarter of the edge points, such that these points are quite far

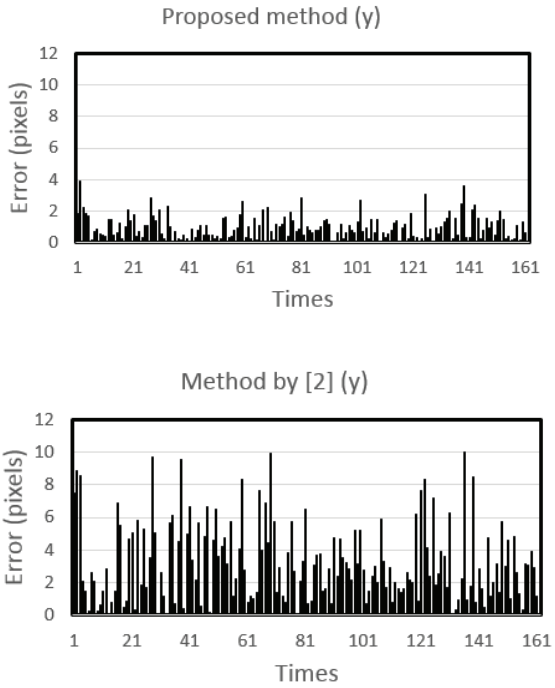
from the iris. To the other edge points, 0–10 random pixel errors were added. After repeat simulations (Fig. 5.4b), the average fitted iris center error obtained using FIFM simulation is 4.7 pixels while the error obtained using our method is 1.8 pixels (Table 5.2). This shows that FIFM is more sensitive to outliers in comparison to our method.

Simulation 3: In the first two simulations, no errors were introduced on the eyeball center. However, our method is based on the eye model and the center of the eyeball is a key factor. In fact, the estimation of the center of the eyeball may not be very accurate. To validate the robustness against eyeball center error, a 1 mm error was added to the on x, y, and 2 mm to the z coordinate of the center of the eyeball, while using the same edge points as simulation 2. The average fitted iris center error for the x and y coordinates are three pixels and two pixels, respectively. The errors observed in our method are still lower than the errors tested by FIFM.

All the above simulations show that our method is more robust than the traditional five parameters fitting method. The efficiency of our method is enhanced when the estimation of the center of the eyeball is accurate enough.



(a)



(b)

Fig. 5.4: Iris fitting error.

TABLE 5.1
THE AVERAGE FITTED IRIS CENTER ERROR

Method in [2](x)	Method in [2](y)	Our method(x)	Our method(y)
2.8 pixels	3.3 pixels	0.9 pixels	1.0 pixels

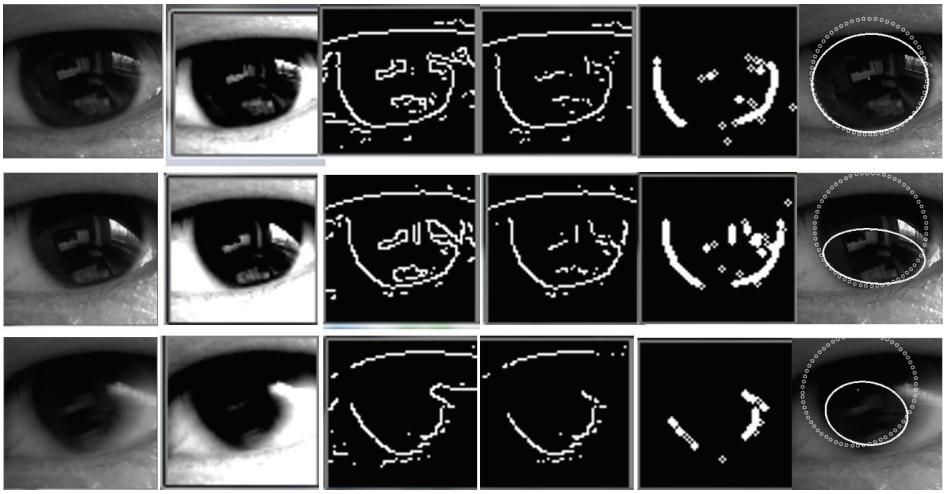
TABLE 5.2
FITTED IRIS CENTER ERROR WITH OUTLIERS

Method in [2](x)	Method in [2](y)	Our method(x)	Our method(y)
5.2 pixels	4.3 pixels	1.9 pixels	1.7 pixels

5.4.2 Method validation

To validate our method and to demonstrate that it is more robust and accurate, we also performed experiments employing conditions not favorable to our method. Sometimes due to poor lighting, eyelid occlusion, and variable illumination, the distinction between the iris region and background is often ambiguous. Figure. 5.5a shows examples that include blurring, iris reflection, and occlusion. First, to increase the contrast, we compute mean intensity in region S1 and S2 (Fig. 5.6). In general, pixels in these two regions can represent iris and eyelid color respectively, and these parts are usually not occluded. Then refer to these two values, the input intensity values, which are lower than S1, would be changed to 0. The input intensity values, which are higher than S2, would be changed to 255. The input intensity values between S1 and S2 would be stretched. The image after stretch is shown as Fig. 5.5b. It can do a large help for segmentation, but at the same time, it also increased the iris reflection. Then Canny algorithm is adopted to obtain the rough edge points. From Fig. 5.5c, you can see that the edge detecting result is really a messy, especially disturbed by iris reflection, eyelids, and eyelashes. We use the vector product strategy to eliminate the noises. An initial 2D iris center is known by Ref. [29], we take it as a possible center, like point C in Fig. 5.6. The normalized displacement to an iris edge point E is d, while the gradient vector at point E is g. Generally, even the point C is not accurate, at least the iris edge points should satisfy the principle that the vector product is positive. To guarantee not eliminate the iris edge point, we only eliminate one third of the rough edge points by vector product intensity. The result is shown as Fig. 5.5d. Although up to now the iris edge is not complete, the edge points are still

there. Because the eye image is captured by head-mounted camera, the eye would not have a rotation movement relative to camera. In other words, the iris edge points would always have a strong horizontal gradient. Therefore, last step; we filter the edge points by a given horizontal gradient threshold. Considering the illumination influence, in our experiments; we only set the threshold as 30. One more reason why we take this tactics is that the left and the right regions of iris are usually not occluded. As you can see from Fig. 5.5e, after these steps, most of the remaining edge points belong to iris. Based on these points, as shown in Fig. 5.5f, our method can fit the iris contour much more accurately (the circle with dotted line) than FIFM (the circle with solid line). This implies that our method is much more robust with outliers caused by eyelids or other disturbances than FIFM.



(a) (b) (c) (d) (e) (f)

Fig. 5.5: The iris detection on the occlusion, iris reflection, and blurred situation.

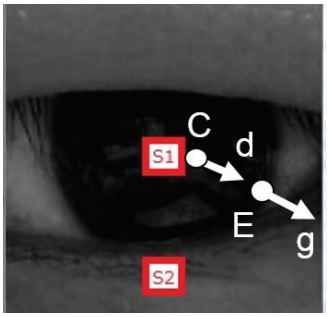


Fig. 5.6: Vector product principle.

5.4.3 Eyeball calibration

To validate our eyeball calibration method, four different samples were acquired for testing. First, in order to have a reference value, the subject was allowed to gaze at the camera center. Then, a scale was manually defined. Thus, the 3D iris position was measured, and subsequently, the 3D center of the eyeball was determined on the basis of the eye model.

Next, the approach used in the FIFM method [2] was used to fit the iris, and the center of the eyeball was calculated using the approach employed in Ref. [12] for five frames. The selected final eyeball center locations using our proposed method with and without relaxation are listed in Table 5.3. As aforementioned, because of the eyelid, the FIFM ellipse fitting method performs poorly in the vertical direction (Fig. 5.3a). As a result, the error without relaxation method on the y axis is 5 mm, which is beyond the range allowed for the eye model. With the relaxation method, the y axis error decreases to 1 mm, while the z axis error is approximately 2 mm. As shown in simulation 3, in this case, our method performs well for eyeball center estimation.

5.4.4 Iris fitting

The calibrated eyeball center positions (Table 5.3) were used to test our fitting method on four subjects. No limitations were imposed during this test, and the subjects were allowed to freely move their eyeballs or blink. A 500 frame video was recorded for each subject and the number of iris fitting failed frames is listed in Table 5.4. The results show that our method is successful in fitting the iris, and failed in very few frames, while the traditional method has a much lower performance (Fig. 5.5f).

TABLE 5.3
EYEBALL CALIBRATION VALIDATION

Samples	Ground truth (cm)	Without iterations (cm)	With iterations (cm)
1	(0.15, 1.24, 7.14)	(0.19, 1.68, 6.96)	(0.04, 1.21, 7.17)
2	(0.15, 1.12, 6.84)	(0.18, 1.60, 6.84)	(0.12, 1.19, 7.05)
3	(0.32,1.30, 7.01)	(0.41, 1.78, 6.89)	(0.33, 1.40, 7.20)
4	(0.33, 0.62, 6.10)	(0.35, 1.00, 5.97)	(0.22, 0.64, 6.10)

TABLE 5.4
IRIS FITTING TEST ON SAMPLES

Sample	Frames	Failed frames
1	500	7
2	500	4
3	500	2
4	500	17

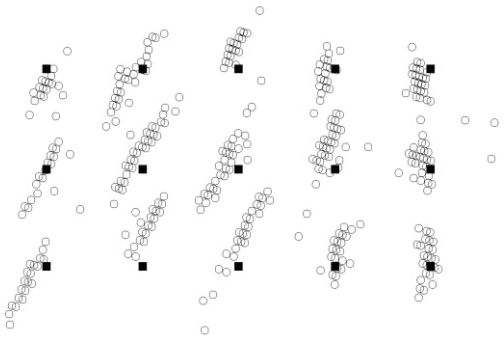
5.4.5 Screen test

The accuracy of our gaze estimation method and other existed methods [5, 27, 36, 37] using targets on a screen was evaluated. The resolution of the screen is 1920 × 1080 pixels, and fifteen points are displayed on the screen. These points are separated from each other at a fixed distance of 6.5 cm. During this validation, the user was asked to gaze at each of these points for a while, and this validation was conducted in a continuous mode without interruption. The distance between the user and the screen was 50 cm. The computed gaze points on the screen are shown in Fig. 5.7a. The vertical and horizontal angle error for each observation is shown in Fig. 5.7b. The average error is 1.49° in the vertical direction and 0.89° in the horizontal direction. Table 5.5 shows the average angle error of each marker. The comparison with some recent research are summarized in Table 5.6.

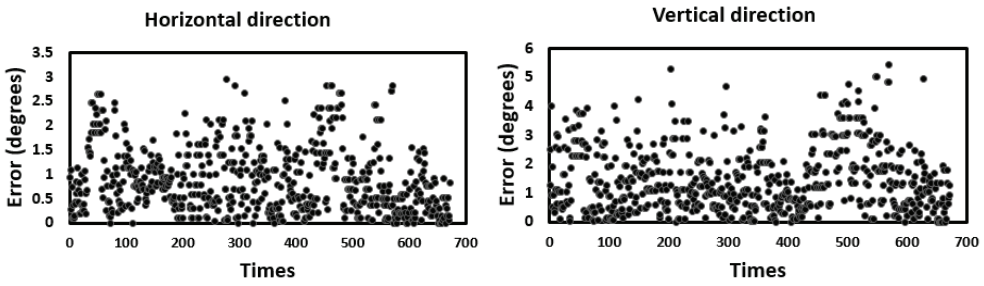
All the above experiments were performed using OPENCV in Visual Studio using a 2.5-GHz Inter(R) Core™ computer with an i5-2400S processor and an 8 GB RAM. The average processing time per frame are 160 ms.

TABLE 5.5
AVERAGE ANGLE ERROR OF EACH MARKER

Marker	Vertical(degrees)	Horizontal(degrees)
1	1.46	0.52
2	1.98	1.54
3	1.28	1.05
4	1.20	0.80
5	1.49	0.86
6	1.58	0.81
7	1.16	1.00
8	1.30	1.08
9	1.33	0.70
10	0.72	0.76
11	1.92	1.68
12	2.63	0.67
13	2.19	0.97
14	1.06	0.43
15	0.98	0.47
Average	1.49	0.89



(a)



(b)

Fig. 5.7: Screen gaze point error.

TABLE 5.6
COMPARISON WITH OTHER WORKS

Item	Error(degrees)
Sugano et al. [36]	2-4
Guestrin et al. [5]	1-3
Chen et al. [37]	1.8-2
Takemura et al. [27]	3
Proposed	0.5-2

5.5 Conclusion

In this chapter, we propose a novel method for gaze tracking of head-mounted eye camera. In the proposed method, we exploit the characteristics of the head-mounted eye camera system, and decouple the estimation of the eyeball center and iris contour. We demonstrate that the iris can be extracted in a more efficient and accurate manner by projecting the 3D iris contour onto a 2D space. Moreover, an automatic eyeball center calibration procedure was introduced. As shown in the comparative experiments, the proposed method achieves higher accuracy than conventional methods

CHAPTER 6

Conclusions

6.1 Summary

In this thesis, we focus on the problem of estimating gaze from a remote camera and a head-mounted camera. And novel methods of gaze estimation based on an eye model respectively for a remote camera and a head-mounted camera are proposed.

For a remote RGB-D camera--Kinect, based on the given head pose, a novel method of estimating gaze direction is proposed. This method is not like the traditional gaze estimation method, by using the 3D eye-model and the knowledge of projection, we first introduced a simple calibration method for the eyeball center. As a result of biological structure, there are two axes in our eye model, one is the visual axis, which passes through the eyeball center to iris center. The other one is the optical axis, which passes through the corneal center to gaze point. In theory, the visual axis is our gaze direction, but the real gaze direction is actually optical axis. Because the visual axis and the optical axis can form a constant angle, we use this principle to calibrate the eyeball center position. And about obtaining the 3D gaze point, we did not use the traditional way that put a target in front of the camera but take the camera center as target to gaze, so the problem of face occlusion can be eliminated, and it can also simplify our objective function. Next we estimate the iris center by projecting the 3D contour of the iris to the RGB image using the eye model and RGB cues. In this procedure, four coordinate systems are involved, and coordinate transformation is widely used. Finally the iris contour estimation would

only rely on only two unknowns, which is better than the common five parameters fitting method. This simplified representation results in simpler and more accurate gaze estimation. Moreover, our method needs no extra training stages and devices, and it can also run automatically in real time with a high-resolution image.

For the head mounted camera, we also proposed a novel method for gaze tracking of head-mounted eye camera. Based on the acquired eye image, we detect whether the position is changed, if the camera has been moved, we proposed an eyeball center calibration method to calibrate the eyeball center position. First of all, from one frame, we use the conventional ellipse fitting method to obtain an iris contour, then based on the principle of circle/ellipse correspondence, we infer an eyeball center position. Next by our proposed iris fitting method, we obtain a more accurate iris contour, which can infer a more accurate eyeball center. This procedure is iterated for a few times, then we can have a rather accurate eyeball center position for this frame. Notice that the eyeball center is fixed in the camera coordinate system whatever the frame passes, so we take a few frames to calculate a set of eyeball center positions and find a credible solution. By our proposed method, we can calibrate the eyeball center automatically. After calibrating the eyeball center, we fit the iris at every frame in real-time. Referring to the 3D iris position and its 2D projection on images, a method that detecting the iris contour with only two parameters is proposed. Even in some situations that the iris is not complete, our proposed iris fitting method can fit the contour as well, while the traditional way could not perform well. Last, we also design a screen experiment to validate our proposed gaze estimation method. As

shown in the comparative experiments, the proposed method achieves higher accuracy than conventional methods.

6.2 Future work

For future work of our proposed gaze estimation methods, we plan to use them in driver support system. In this system, there are two cameras are involved. One camera is set inside the car and let it capture the driver's gaze, the other camera is set in front of the car to capture the view. We calibrate the pose of two cameras, then we can combine the estimated gaze from the first camera and the view from the second camera together, so we can easily understand where the driver is gazing at. Based on this system, we can analysis the driver's behavior during driving. For example, according to the view from the front camera, we can know where the traffic signal is and what the color it is, and also we can know whether the driver is noticing the traffic signal by gaze estimation from the inside camera.

A number of obvious improvements could be made to our current implementation. For example, eyeball center calibration procedure for the head-mounted camera is quite time consuming, we are looking for a better and more efficient way to improve it. There is also room for additional improvement if we can eliminate more outliers of iris feature points as soon as possible. Our research is aimed at developing reliable gaze estimation method that can run on general-purpose hardware and that can be widely employed in every day human-computer interfaces.

REFERENCES

[1] D. W. Hansen and Q. Ji. In the eye of the beholder: a survey of models for eyes and gaze. IEEE Trans. on Pat. Analysis and Machine Intelligence, 32(3):478-500, Mar. 2010.

[2] D. H. Li, D. Winfield. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) -Workshops, 3:79–79, 2005.

[3] X, Xiong, et al. Eye gaze tracking using an RGBD camera: a comparison with a RGB solution. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. ACM, 2014.

[4] J. Chen, Q. Ji. 3D gaze estimation with a single camera without IR illumination. In Proceedings of the 19th International Conference on Pattern Recognition, pp. 1-4, 2008.

[5] E. D. Guestrin, M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. IEEE Transactions on biomedical engineering, 53(6): 1124-1133, 2006.

[6] K. Tan, D. J. Kriegman, and N. Ahuja. Appearance-based eye gaze estimation. In Proceedings of the 6th IEEE Workshop on Applications of Computer Vision, pp. 191-195, 2002.

[7] F. Lu, T. Okabe, Y. Sugano, and Y. Sato. A head pose-free approach for appearance-based gaze estimation. British Machine Vision Conference, pp. 1-11, 2011.

[8] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring Human Gaze from Appearance via Adaptive Linear Regression. In ICCV: International Conference on Computer Vision, Barcelona, Spain, 2011.

[9] B. Noris, J. Keller, and A. Billard. A wearable gaze tracking system for children in unconstrained environments. Computer Vision and Image Understanding, pp. 1-27, 2010.

[10] Z. Zhu, Q. Ji, and K. P. Bennett. Nonlinear eye gaze mapping function estimation via support vector regression. International Conference on Pattern Recognition, pp. 1132-1135, 2006.

[11] R. Valenti, T. Gevers. Accurate eye center location and tracking using isophote curvature. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1-8, 2008.

[12] J. G. Wang, E. Sung, and R. Venkateswarlu. Eye gaze estimation from a single image of one eye. In Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV 2003), pp. 136-143, 2003.

[13] E. D. Guestrin and M. Eizenman. Remote point-of-gaze estimation requiring a single-point calibration for applications with infants. In Proceedings of the 2008 symposium on eye tracking research & applications, pp. 267-274, 2008.

[14] T. Nagamatsu, J. Kamahara, and N. Tanaka. 3D gaze tracking with easy calibration using stereo cameras for robot and human communication. In

Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, pp. 59-64, 2008.

[15] Z. Zhu, Q. Ji. Novel eye gaze tracking techniques under natural head movement. IEEE Transaction on Biomedical Engineering, 54(12):2246-2260, 2007.

[16] T. Ishikawa, S. Baker, and I. Matthews. Passive driver gaze tracking with active appearance models. In proc. World congress on Intelligent Transportation Systems, pp. 1-12, 2004.

[17] R. Jafari, D. Ziou. Gaze estimation using Kinect/PTZ camera. In Proceedings of 2012 IEEE International Symposium on Robotic and Sensors Environments, pp. 16-18, 2012.

[18] Y. Li, D. S. Monaghan, and N. E. Connor. Real-time gaze estimation using a Kinect and a HD webcam. MultiMedia Modeling. 8325:506-517, 2014.

[19] K. A. F Mora, J. Odobez. Gaze estimation from multimodal Kinect data. Computer Vision and Pattern Recognition Workshops, pp. 25-30, 2012.

[20] K. A. F Mora, J. Odobez. Geometric generative gaze estimation for remote RGB-D cameras. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 1773-1780, 2014.

[21] N. K. Mahadeo, A. P. PAPLINSKI, S. RAY. Robust video based iris segmentation system in less constrained environments. In: Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on. IEEE, 2013. p. 1-8.

[22] Y. Du, E. Arslanturk, Z. Zhou. Video-based noncooperative iris image segmentation. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2011, 41.1: 64-74.

[23] D. Zhu, S. Moore, and T. Raphan. Robust pupil center detection using a curvature algorithm. Computer Methods and Programs in Biomedicine, vol. 59, no. 3, pp. 145–157, 1999.

[24] J. G. Wang, E. Sung, and R. Venkateswarlu. Gaze estimation from a single image of one eye. In Proceedings of IEEE International Conference on Computer vision, pp. 136-143, 2003.

[25] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.

[26] L. Zhang, et al. It starts with gaze: Visual attention driven networking with smart glasses. In: Proceedings of the 20th annual international conference on Mobile computing and networking. ACM, 2014. p. 91-102.

[27] K. Takemura, et al. Estimating 3-D point-of-regard in a real environment using a head-mounted eye-tracking system. Human-Machine Systems, IEEE Transactions on, 2014, 44.4: 531-536.

[28] X. Xiong, Q. Cai, Z. Liu, and Z. Zhang. Eye gaze tracking using an RGBD camera: a comparison with a RGB solution. PETMEI 2014.

[29] F. Timm, E. Barth. Accurate eye center localization by means of gradients. In Proceedings of the International Conference on Computer Theory and Applications, volume 1, pp. 125-130, 2011.

[30] N. Smolyanskiy, C. Huitema, L. Liang, and S. Anderson. Real-time 3d face tracking based on active appearance model constrained by depth data. In Image and Vision Computing, 2014. 2.

[31] J. LI, S. LI. Eye-model-based gaze estimation by RGB-D Camera. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 606-610, 2014.

[32] Funes Mora, K. A., Monay, F., and Odobez, J.-M. 2014. EYEDIAP database: Data description and gaze tracking evaluation benchmarks. Tech. Rep. RR-08-2014, Idiap, May 2014.

[33] Funes Mora, K. A., Monay, F., and Odobez, J.-M. EYEDIAP: a database for the development and evaluation of gaze estimation algorithms from RGB and RGB-D cameras. In Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA 14), pp. 255-258, ACM, 2014.

[34] C. Xu, Y. Zheng, Z. Wang. Semantic feature extraction for accurate eye corner detection. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, pp. 1-4. 2008.

[35] H. S. Sawhney, J. Oliensis, and A. R. Hanson. Description and reconstruction from image trajectories of rotational motion. In Proceedings of IEEE International Conference on Computer vision, pp.494-498, 1990.

[36] Y. Sugano, Y. Matsushita, and Y. Sato. Appearance-based gaze estimation using visual saliency. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 35, no. 2, pp. 329-341, 2013.

[37] B. Chen, P. Wu, and S. Chien. Real-time eye localization, blink detection, and gaze estimation system without infrared illumination. In Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, pp. 715-719. 2015.

[38] D. A. Robinson. A method of measuring eye movement using a scleral search coil in a magnetic field. IEEE Transactions on Bio-Medical Electronics, pp. 137-145, October 1963.

[39] A. Bulling, D. Roggen, and G. Troster. Wearable EOG goggles: seamless sensing and context-awareness in everyday environments. Journal of Ambient Intelligence and Smart Environments, vol. 1, no. 2, pp. 157-171, 2009.

[40] http://www.laramyk.com/resources/education/occular-anatomy/major-ocular-structures.

[41] http://www.preventblindness.org/eye-how-we-see.

[42] http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html.

[43] http://www.maa.org/external_archive/joma/Volume8/Kalman/General.html.

LIST OF PUBLICATIONS

Journals	
Jianfeng LI, Shigang LI. Gaze estimation from color image based on the eye model with known head pose. IEEE Transaction on Human-machine Systems, Vol. 46, No. 3, pp. 414-423, 2016.	Related to Chapter 4 in this thesis
International Conferences and Workshops:	
Jianfeng LI, Shigang LI. Eye-model-based gaze estimation by RGB-D Camera. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 606-610, 2014.	Related to Chapter 4 in this thesis
Jianfeng LI, Shigang LI. Two-phase approach – calibration and iris contour estimation – for gaze tracking of head-mounted eye camera. In Proceedings of IEEE International Conference on Image Processing (ICIP), 2016.	Related to Chapter 5 in this thesis