3D Eye Model-Based Gaze Estimation from A Depth Sensor*

Xiaolong Zhou[1], Haibin Cai[2], Zhanpeng Shao[1], Hui Yu[3] and Honghai Liu[2]

Abstract— **In this paper, we address the 3D eye gaze estimation problem using a low-cost, simple-setup, and non-intrusive consumer depth sensor (Kinect sensor). We present an effective and accurate method based on 3D eye model to estimate the point of gaze of a subject with the tolerance of free head movement. To determine the parameters involved in the proposed eye model, we propose i) an improved convolution-based means of gradients iris center localization method to accurately and efficiently locate the iris center in 3D space; ii) a geometric constraints-based method to estimate the eyeball center under the constraints that all the iris center points are distributed on a sphere originated from the eyeball center and the sizes of two eyeballs of a subject are identical; iii) an effective *Kappa* angle calculation method based on the fact that the visual axes of both eyes intersect at a same point with the screen plane. The final point of gaze is calculated by using the estimated eye model parameters. We experimentally evaluate our gaze estimation method on five subjects. The experimental results show the good performance of the proposed method with an average estimation accuracy of 3.78°, which outperforms several state-of-the-arts.**

I. INTRODUCTION

Gaze estimation is a technique to determine the visual attention of a person, which has been widely explored in human-computer and human-robot interaction systems. Recently, Kinect-based 3D gaze estimation [1], [2], [3], [4], [5], [6], [7], [8] has attracted increasing attention since it is low-cost, non-intrusive, simple-setup and it allows free head movements.

Typically, Kinect-based gaze estimation methods can be roughly classified into non-eye model-based methods and 3D eye model-based methods. Non-eye model-based methods are typically appearance-based or regression-based. The main benefit of non-eye model-based methods is specific personal calibration free. For example, Mora and Odobez [1] estimated 3D gaze under free-head movements from multimodal Kinect data. They estimated the head pose from the depth data and estimated the gaze vectors based on the estimated head pose. The method achieved a low estimation accuracy around $7.6°-12.6°$. Recently, they proposed a geometric generative 3D gaze estimation method [2] based on an appearance generative process that modeled head-pose rectified eye images recovered by using of RGB-D cameras. The estimation accuracy was improved to 6.3°. Reale et al. [3] used the Kinect to obtain the coarse 3D position of the head, and drove the active camera to estimate the 3D head pose. They estimated the eye gaze according to the detected iris and extracted contour points based on the 3D head pose. However, their eye gaze method performed best only when the camera system was fully calibrated, which meant the accurate orientation and position of the camera and screen should be known ahead of time. Similarly, Cazzato et al. [4] investigated a 3D gaze estimation method based on the head pose information extracted by an RGB-D sensor. They first used the RGB image to face detection and tracking, and then applied depth information to match the tracked feature points with a 3D point cloud. The estimated 3D head pose was incorporated to estimate the final gaze direction according to the geometric relations among the sensor, observer, and target. They reported the estimation errors for unaware users with 6.9° while for informed users with 3.6°.

Different from the non-eye model-based methods that estimate the gaze using appearance or regression technique, 3D eye model-based methods directly determine the gaze using the geometric relationship among human eyes, sensors and gazing points. In contrast, 3D eye model-based gaze estimation methods can achieve a higher accuracy. For example, Li and Li [5] proposed an eye-model-based 3D gaze estimation method by Kinect sensor. They built a head model based on the Kinect sensor and calibrated the eyeball center by gazing at a target in 3D space. The gaze direction can finally be estimated after the calibration and the reported average error of estimation was around 6°. Recently, they estimated the gaze from color image based on an eye model with known head pose [6]. They first determined the 3D eyeball center in calibration manner by gazing at the center of the color image camera, and then estimated the 3D iris center using the contour and projection information of the iris. They reported the average estimation errors for seven subjects with 5.9° vertically and 4.4° horizontally. Sun et al. [7] estimated the gaze direction based on a 3D geometric eye model by considering the head movement and deviation of the visual axis from the optical axis. They reported a high estimation accuracy of 1.4°-2.7°. However, the proposed method involved many calibration procedures like screen-camera calibration and personal calibration.

To leverage the accuracy and automation, we present an accurate 3D eye model-based method to estimate the point of gaze from a single Kinect sensor by calculating the iris

*This work was supported in part by the EU Seventh Framework Programme (No. 611391, Development of Robot-Enhanced therapy for children with AutisM spectrum disorders (DREAM)), the National Natural Science Foundation of China (No. 61403342 and 61273286), and Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering (No. 2014KLA09).

[1]Xiaolong Zhou and Zhanpeng Shao are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. They are also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen, China. zxl@zjut.edu.cn

[2]Haibin Cai and Honghai Liu are with the School of Computing, University of Portsmouth, Portsmouth, UK.

[3]Hui Yu is with the School of Creative Technologies, University of Portsmouth, Portsmouth, UK.

center, eyeball center, and *Kappa* angle in an effective way. The main contributions of this paper are those given here.

1) We propose an effective 3D eye model-based gaze estimation method that can achieve a relative low average estimation error (about 3.78°) with free head movements. The experimental results demonstrate that our method outperforms many state-of-the-art Kinect-based gaze estimation methods.

2) We improve the conventional means of gradients iris center localization method in a convolution way. The improved method either improve the accuracy or dramatically reduce the computational cost.

3) We employ a geometric constraints-based method to estimate the eyeball center. The constraints assume that all the iris center points are distributed on a sphere originated from the eyeball center and the sizes of two eyeballs of a subject are identical.

4) We calculate the *Kappa* angle using the constraint that the visual axes of both eyes intersect at a same point with the screen plane when a subject gazes a point on the screen.

The rest of this paper is organized as follows. Section II introduces the overview of 3D eye model-based gaze estimation. Section III details the eye model parameters determination. Experimental results are discussed in Section IV, and followed by concluding remarks in Section V.

II. 3D Eye Model-Based Gaze Estimation

Fig. 1 illustrates the 3D model of two eyes of a subject. $\mathbf{O}_e^L$ (or $\mathbf{O}_e^R$), $\mathbf{O}_c^L$ (or $\mathbf{O}_c^R$), and $\mathbf{P}_i^L$ (or $\mathbf{P}_i^R$) denote the centers of eyeball, cornea, and pupil of left eye (or right eye), respectively. The dash lines through the centers of eyeball, cornea and pupil represent the optical axes for both eyes. $\mathbf{V}_o^L$ (or $\mathbf{V}_o^R$) is a unit vector of optical axis of left eye (or right eye). The lines from the corneal center to the point of gaze on the screen plane denote the visual axes for both eyes. $\mathbf{V}_g^L$ (or $\mathbf{V}_g^R$) is a unit vector of visual axis of left eye (or right eye). The angle of deviation of the visual axis from the optical axis is known as the *Kappa*, which almost keeps constant for each subject. It consists of two components, namely, the horizontal angle α^L (or α^R) and the vertical angle β^L (or β^R). $X_W Y_W Z_W$ represents for the world coordinate system, which is built at the center of the Kinect sensor. $X_H Y_H Z_H$ represents for the head coordinate system, which is built at the middle of a subject's face. When a subject looks at a point on the screen plane, the points of gaze of left and right eyes are represented as $\mathbf{P}_g^L$ and $\mathbf{P}_g^R$, respectively. In this paper, both eyes are modeled to effectively estimate the point of gaze. Without explicit description, all the parameters involved are relative to the world coordinate system.

Take left eye as an example, the point of gaze can be calculated by

$$\mathbf{P}_g^L = \mathbf{O}_e^L + c \cdot \mathbf{V}_o^L + \lambda^L \cdot \mathbf{V}_g^L \tag{1}$$

where $c = \left\|\mathbf{O}_e^L \mathbf{O}_c^L\right\|_2$ is a constant, and normally it is approximately 5.3mm [9]. $\lambda^L = \left\|\mathbf{O}_c^L \mathbf{P}_g^L\right\|_2$ can be obtained by

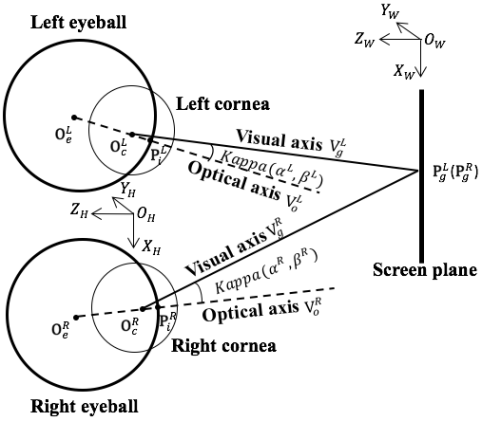

Fig. 1: An illustration of 3D eye model of left and right eyes.

$$\lambda^L = -\frac{(\mathbf{O}_e^L + c \cdot \mathbf{V}_o^L) \cdot \mathbf{V}_s + n}{\mathbf{V}_g^L \cdot \mathbf{V}_s} \tag{2}$$

where $\mathbf{V}_s$ and n are parameters of screen plane function and can be determined from the camera-screen calibration [7]. For any point $\mathbf{P}_g$ on the screen plane, we have $\mathbf{P}_g \cdot \mathbf{V}_s = -n$.

The eyeball center $\mathbf{O}_e^L$ is variable for different head poses in the world coordinate system, while it ($\mathbf{O}_e^{L,H}$) keeps constant related to the head coordinate system. They can be transformed with the estimated head pose $\{\mathbf{R}_t, \mathbf{t}_t\}$.

$$\mathbf{O}_e^L = \mathbf{O}_e^{L,H} \cdot \{\mathbf{R}_t, \mathbf{t}_t\} \tag{3}$$

where $\mathbf{R}_t$ and $\mathbf{t}_t$ are the rotation matrix and translation matrix of head pose at time t, respectively. The details of head pose estimation are stated in section III. A.

The unit vector of optical axis $\mathbf{V}_o^L$ is calculated according to the eyeball center $\mathbf{O}_e^L$ and iris center $\mathbf{P}_i^L$.

$$\mathbf{V}_o^L = \frac{\mathbf{P}_i^L - \mathbf{O}_e^L}{r_e^L} \tag{4}$$

where $r_e^L = \left\|\mathbf{O}_e^L \mathbf{P}_i^L\right\|_2$ denotes the radius of an eyeball.

The unit vector of visual axis $\mathbf{V}_g^L$ is calculated by rotating the optical axis with the *Kappa* angle (α^L, β^L).

$$\mathbf{V}_g^L = \mathbf{V}_o^L \cdot f(\alpha^L, \beta^L) \tag{5}$$

So far, we can estimate the point of gaze of left eye $\mathbf{P}_g^L$ according to the abovementioned equations. Similarly, the point of gaze of right eye $\mathbf{P}_g^R$ can be estimated. To further improve the estimation accuracy, we calculate the final point of gaze $\mathbf{P}_g$ based on the fact that both eyes gaze at a same point on the screen.

$$\mathbf{P}_g = \frac{1}{2}(\mathbf{P}_g^L + \mathbf{P}_g^R) \tag{6}$$

However, to accurately estimate the 3D point of gaze of a subject, the involved unknown head pose $\{\mathbf{R}_t, \mathbf{t}_t\}$ and eye model parameters that include $\mathbf{P}_i^L$ ($\mathbf{P}_i^R$), $\mathbf{O}_e^L$ ($\mathbf{O}_e^R$), r_e^L (r_e^R), α^L (α^R) and β^L (β^R) should be determined beforehand. In Section III, we will introduce the specific methods for eye model parameters determination in detail.

III. Eye Model Parameters Determination

Firstly, we estimate the head pose $\{\mathbf{R}_t, \mathbf{t}_t\}$ of a subject at time t according to the detected facial features and modeled 3D face. Secondly, we present a fast and accurate method based on the convolution of means of gradients to locate the 3D position of iris center $\mathbf{P}_i^L$ (or $\mathbf{P}_i^R$). Thirdly, we calculate the eyeball center $\mathbf{O}_e^L$ (or $\mathbf{O}_e^R$) and radius r_e^L (or r_e^R) based on three geometric constraints. Finally, we employ the constraint that both eyes of a subject share a same point of gaze to determine the *Kappa* angle $\{\alpha^L, \beta^L\}$ (or $\{\alpha^R, \beta^R\}$).

A. Head Pose Estimation

We first detect the face region in the RGB image using an appearance-based boosted cascade face detector [10] with default parameters. After the face region has been identified, we employ a fast and accurate supervised descent method (SDM) [11] to detect and track the facial features. After the facial features of a subject at time t have been detected, the corresponding 3D coordinates of the features can be obtained by the calibrated Kinect sensor. We then model the 3D face of this subject at time t using the obtained 3D features as $\mathbf{X}_t$. Typically, the face model of each subject is person-specific.

The goal of head pose estimation is to determine the head rotation matrix $\mathbf{R}_t$ in terms of yaw, pitch and roll, and translation vector $\mathbf{t}_t$, at time t. To calculate the head pose, a reference face model of the subject should be built first. We require each subject to keep frontal to the Kinect sensor for a certain time and calculate the average to form a reference model $\mathbf{X}^r$. Then, the head pose of a subject at time t can be determined by minimizing the following equation.

$$\arg\min_{\mathbf{R}_t, \mathbf{t}_t} \|\mathbf{R}_t \mathbf{X}^r + \mathbf{1}_{1\times n} \otimes \mathbf{t}_t - \mathbf{X}_t\| \tag{7}$$

where $\mathbf{1}_{1\times n}$ denotes a row vector of ones of size n (n is the number of feature points), $\otimes$ represents the Kronecker produce. We can solve Eq. (7) using Singular Value Decomposition [12] and then obtain the head pose $\{\mathbf{R}_t, \mathbf{t}_t\}$ of the subject at time t.

B. 3D Iris Center Localization

The means of gradients method [13] for iris center localization has attracted considerable attention due to its easy implementation and high accuracy. It makes use of the relationship between a possible iris center and the vector field of all the image gradients.

$$\begin{aligned} c' &= \arg\max_c \left\{ \frac{1}{N} \sum_{i=1}^{N} (\mathbf{d}_i^T \cdot \mathbf{g}_i)^2 \right\} \\ \mathbf{d}_i &= \frac{x_i - c}{\|x_i - c\|_2} \qquad \forall i : \|\mathbf{g}_i\|_2 = 1 \end{aligned} \tag{8}$$

where c' denote the located iris center position and c is the possible iris center. The dot product will reach the biggest if the displacement vector d_i and the gradient vector g_i have the same orientation which will happen if the point x_i lies on the boundary of the circle whose center point is c. The displacement vector d_i and gradient vector g_i are scaled to unit length to obtain an equal weight for all pixel positions. N is the number of pixels of the image. The algorithm calculates dot products of the normalised displacement vectors and the gradient vectors for every possible iris center. Each pixel in the image is a potential iris center. The pixel that has the maximum value of mean of dot products is regarded as the final iris center.

Although the means of gradients method can locate iris center accurately, the heavy computational cost hampers its real time applications. The computational complexity of this method is $O(N^2)$, where N stands for the number of pixels of the eye area. The algorithm calculates the dot product of all the displacement vectors d_i and the gradient vectors g_i. Thus for a possible iris center, all the pixels in the eye image are used for the dot product. Although the computational complexity can be decreased by considering only the same orientation displacement vectors and the gradient vectors that have a significant magnitude, the accuracy will drop dramatically. To remedy this, we propose a convolution-based means of gradients method which is capable of reducing the computational cost while at the same time improving the accuracy. In the proposed method only the pixels on the circular boundary of a possible iris center are used to calculate the dot product. So the computational complexity can be greatly reduced. Meanwhile, the negative influence of other points such as eyelids and eye corners in the dot product can also be eliminated. Different sizes of masks are built to convolute the eye images. Each mask contains a circle whose center point is at the center of the masks and the pixels value on the boundary of the circle are normalised. Assuming the radius of the circle is r, then both the width and height of the built mask will be $2r+1$.

We propose to apply convolution to the dot product and the corresponding equation can be further extended to the following.

$$\begin{aligned} &\sum_{i=1}^{n} \mathbf{d}_i^T \cdot \mathbf{g}_i = \sum_{i=1}^{n} (x_{di} x_{gi} + y_{di} y_{gi}) \\ &= \sum_{i=1}^{n} \big((x_i - x_c) x_{gi} + (y_i - y_c) y_{gi} \big) \\ &= \sum_{i=1}^{n} (x_i x_{gi}) - x_c \sum_{i=1}^{n} x_{gi} + \sum_{i}^{n} (y_i y_{gi}) - y_c \sum_{i}^{n} y_{gi} \end{aligned} \tag{9}$$

where (x_{di}, y_{di}) is the coordinate of d_i, which can be calculated by the difference between the circular boundary point (x_i, y_i) and the possible iris center (x_c, y_c). (x_{gi}, y_{gi}) is the coordinate of g_i, which can be calculated through partial derivatives or other methods by computing image gradients. We build two position images I_{px} and I_{py} of pixels for x and y positions, respectively. The size of the position image is the same as the size of eye region image. Similarly, two gradient images I_{gx} and I_{gy} of pixels for x and y directions also can be obtained. By doing so, the $\sum_{i=1}^{n} (x_i x_{gi})$ can be calculated by firstly multiplying position image I_{px} with gradient image I_{gx} and then using the former designed mask to convolute the result. The $x_c \sum_{i=1}^{n} x_{gi}$, similarly, can be calculated by firstly

convoluting the designed mask with the gradient image I_{gx} and then multiplying the result with the position image I_{py}.

Since only the pixels on the boundary of the circles are used to calculate the dot products, the other pixels of the eye image cannot affect the result. Thus we propose to directly use the sum of the dot products rather than the square of the dot products. The final position of iris center is determined by searching the maximum of the following equation.

$$\max_{(r,x_0,y_0)} (\frac{1}{r}\sum_{i=1}^{n} \mathbf{d}_i^T \cdot \mathbf{g}_i) \tag{10}$$

where (x_0, y_0) represents the coordinate of iris center. To locate the iris center, the proposed method searches the maximum of Eq. (10) by changing the values of radius and center points. The FFT is employed in the realisation of convolution where only 2 cycles DFT and 1 cycle IDFT are performed. So the computational complexity of the convolution-based means of gradients is $O(P\log_2(P)N)$, where P satisfies $P \leq X+Y+C$. X, Y are the number of rows and columns of the center coordinate and C is a constant number. Compared to the computational complexity $O(N^2)$ of the conventional means of gradients method, the proposed method significantly improves the processing speed.

So far, iris center coordinate $\mathbf{p}_i^L$ (or $\mathbf{p}_i^R$) in the image plane can be effectively and efficiently determined according to the proposed method. Then, its coordinate in 3D space $\mathbf{P}_i^L$ (or $\mathbf{P}_i^R$) can be obtained by incorporating the depth information captured by the Kinect sensor.

C. Eyeball Centers Estimation

We estimate the eyeball centers $\mathbf{O}_e^{L,H}$ and $\mathbf{O}_e^{R,H}$ in the head coordinate system based on the following constraints.

Constraint1 : the eyeball is regarded as a sphere with the origin is $\mathbf{O}_e^L$ (or $\mathbf{O}_e^R$) and the radius is r_e^L (or r_e^R), and all the estimated iris centers are distributed on the eyeball sphere.

Constraint2 : two eyeballs of the subject are identical, which means the radius of left eyeball is equal to the radius of right eyeball. Namely, $r_e^L = r_e^R$.

Constraint3 : the eyeball centers $\mathbf{O}_e^{L,H}$ and $\mathbf{O}_e^{R,H}$ are fixed relative to the head coordinate system and independent of gaze direction and head movement, where $\mathbf{O}_e^{L,H}$ and $\mathbf{O}_e^{R,H}$ are the coordinates of eyeball centers in the head coordinate system.

Based on the **Constraint1** and given the estimated iris centers, the eyeball center can be estimated by fitting these iris center points to a sphere. The subject is required to look at M points randomly distributed on the screen. When looking at these points, the head of subject keeps static while the eye direction changes. Noted that the subject does not need to look at the specific defined points but has to change the gaze directions to capture enough iris images. By doing so, all the iris centers $\{\mathbf{P}_{i,j}^L\}_{j=1}^M$ (or $\{\mathbf{P}_{i,j}^R\}_{j=1}^M$) for gazing these points can be regarded distributing on a same sphere. Then, we have

$$r_e^L = \left\|\mathbf{P}_{i,1}^L - \mathbf{O}_e^L\right\|_2 = \cdots = \left\|\mathbf{P}_{i,M}^L - \mathbf{O}_e^L\right\|_2 \tag{11}$$

$$r_e^R = \left\|\mathbf{P}_{i,1}^R - \mathbf{O}_e^R\right\|_2 = \cdots = \left\|\mathbf{P}_{i,M}^R - \mathbf{O}_e^R\right\|_2 \tag{12}$$

We use a series of iris center points $\{\mathbf{P}_{i,j}^L\}_{j=1}^M$ (or $\{\mathbf{P}_{i,j}^R\}_{j=1}^M$) to fit the sphere function of left eyeball (or right eyeball), as shown in Eq. (13) and Eq. (14). To estimate the sphere parameters, the least squares fitting algorithm is used to fit those iris center points distributed on the sphere. More than ten iris center points ($M > 10$) are used in our experiments to ensure the fitting accuracy.

$$(x-a^L)^2 + (y-b^L)^2 + (z-c^L)^2 = (r_e^L)^2 \tag{13}$$

$$(x-a^R)^2 + (y-b^R)^2 + (z-c^R)^2 = (r_e^R)^2 \tag{14}$$

where (a^L, b^L, c^L) and (a^R, b^R, c^R) are the 3D coordinates of $\mathbf{O}_e^L$ and $\mathbf{O}_e^R$, respectively. Based on the **Constraint2** and **Constraint3**, we have

$$r_e^L = r_e^R \tag{15}$$

$$\mathbf{O}_e^{L,H} = \mathbf{O}_e^L \cdot \{\mathbf{R}_t, \mathbf{t}_t\} \tag{16}$$

$$\mathbf{O}_e^{R,H} = \mathbf{O}_e^R \cdot \{\mathbf{R}_t, \mathbf{t}_t\} \tag{17}$$

where $\{\mathbf{R}_t, \mathbf{t}_t\}$ is the head pose that can be determined by the method stated in Section III.A.

Note that once the $\mathbf{O}_e^{L,H}$ and $\mathbf{O}_e^{R,H}$ have been estimated, they remain constant no matter head moves or gaze changes because they are fixed with respect to the head coordinate system. Then, $\mathbf{O}_e^L$ and $\mathbf{O}_e^R$ in the world coordinate system can be calculated according to Eq. (16) and Eq. (17), respectively.

D. Kappa Angle Calculation

To accurately and explicitly calculate the *Kappa* angle, the following constraint is employed.

Constraint4 : when a subject gazes at the screen, the visual axes of both eyes would intersect at a same point with the screen plane. That is, $\mathbf{P}_g^L = \mathbf{P}_g^R$ (shown as $\mathbf{P}_g^L$ ($\mathbf{P}_g^R$) in Fig. 1).

The **Constraint4** holds true for the vast majority of subjects and viewing conditions. The *Kappa* angle of both eyes then can be calculated by the following steps.

Step1: the subject is required to look at a given point on the screen. According to Sections III.B and III.C, 3D iris centers ($\mathbf{P}_i^L$ and $\mathbf{P}_i^R$) and eyeball parameters (centers $\mathbf{O}_e^L$ and $\mathbf{O}_e^R$, and radius r_e^L and r_e^R) of both eyes can be estimated.

Step 2: we calculate the unit vector of optical axis $\mathbf{V}_o^L$ of left eye by Eq. (4) with the parameters estimated in *Step 1*, and similarly we can obtain the $\mathbf{V}_o^R$ of right eye. Then, we can obtain the unit vector of visual axis $\mathbf{V}_g^L$ (or $\mathbf{V}_g^R$) by rotating the optical axis with the *Kappa* angle.

$$\mathbf{V}_g^L = \begin{bmatrix} cos(\varphi^L + \beta^L)sin(\theta^L + \alpha^L) \\ sin(\varphi^L + \beta^L) \\ -cos(\varphi^L + \beta^L)cos(\theta^L + \alpha^L) \end{bmatrix} \tag{18}$$

$$\mathbf{V}_g^R = \begin{bmatrix} cos(\varphi^R + \beta^R)sin(\theta^R + \alpha^R) \\ sin(\varphi^R + \beta^R) \\ -cos(\varphi^R + \beta^R)cos(\theta^R + \alpha^R) \end{bmatrix} \tag{19}$$

where θ^L (or θ^R) and φ^L (or φ^R) denote the horizontal angle and vertical angle of the unit vector of optical axis $\mathbf{V}_o^L$ (or

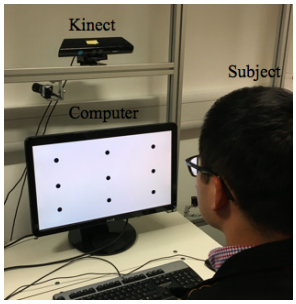

Fig. 2: System setup of the proposed method.

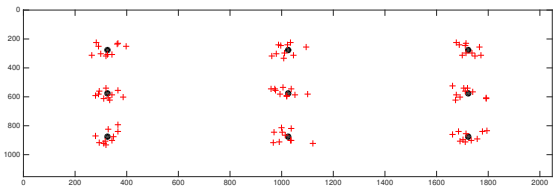

Fig. 3: Estimation of gazes of subject 1 on the screen plane. The black dots and red crosses represent the groundtruth and the estimated gazes, respectively.

$\mathbf{V}_o^R$). α^L (or α^R) and β^L (or β^R) denote the horizontal angle and vertical angle of the *Kappa* angle of left eye (or right eye).

Step 3: we employ Sun's camera-screen calibration method [7] to determine the parameters $\{\mathbf{V}_s, n\}$ of screen plane. Then, we can calculate the λ^L by Eq. (2), and similarly we can obtain λ^R.

Step 4: finally, we determine the point of gaze by Eq. (1) with unknown parameters $\{\alpha^L, \beta^L\}$ (or $\{\alpha^R, \beta^R\}$). Assume that the point of gaze is known, namely, $\mathbf{P}_g^L$ (or $\mathbf{P}_g^R$) is known, then we can calculate the *Kappa* angle $\{\alpha^L, \beta^L\}$ (or $\{\alpha^R, \beta^R\}$) according to Eq. (1).

For a given point with specific coordinate in 3D space, we have 3 equations with only 2 unknown parameters in Eq.(1). Therefore, the *Kappa* angle can be calculated by only a calibrated point. To improve the estimation accuracy, we repeat the abovementioned steps for several times with different head poses or gaze directions from the same gaze point and calculate the average angle as the final *Kappa* angle.

IV. Experiments

Our system only uses a Kinect sensor that is mounted above the computer monitor, as shown in Fig. 2. Each subject is required to sit at front of the monitor and to keep his/her head in the field of view of the Kinect. We test the proposed 3D eye model-based gaze estimation method on the subjects with free head movements.

As shown in Fig. 2, the subject is sitting at front of the computer monitor and gazing at the points on the monitor plane. To estimate the eyeball centers, the subject is first asked to look at 15 randomly distributed points on the computer screen with a same head pose but various gaze directions. Note that we don't need to known the specific coordinates of these points. Then, 3D iris centers can be determined and thus the eyeball centers can be calculated according to Section III.C. To further improve the estimation accuracy of eyeball centers, we require the subject to repeat the above-mentioned procedure for three times with different distances away from the computer monitor. Due to the limited measure range (80cm-400cm in default mode) of the Kinect sensor, the distances are set as 90cm, 100cm, and 110cm in our experiments. The average estimation is determined as the final eyeball center.

To determine the *Kappa* angle, the subject is required to gaze at a given point (calibration point) with known coordinate. Basically, the *Kappa* angle can be estimated with only one gaze (details are stated in Section III.D). To obtain a more accurate estimation, the subject is asked to gaze at the given point with 10 different head poses or gaze directions. The average result is determined as the final *Kappa* angle.

In our experiments, five subjects (three males and two females) are asked to test the proposed gaze estimation method. Table I shows the estimated eyeball parameters and *Kappa* angle of each subject by the proposed method.

After the required eye model parameters have been determined, nine evenly distributed ground truth points are used to test the proposed gaze estimation method. The subject is required to gaze at each point with ten different head poses or gaze directions. Note that the subject can freely move his heads and change his gaze directions as he/she wants. The only constraint is that his/her heads should not be out of view of the field of the Kinect sensor. Otherwise, the sensor could fail to capture the eye images and thus result in failure of gaze estimation. Fig. 3 illustrates the distribution of estimated points of gaze of one subject (subject 1) when gazing at nine given groundtruth points.

To quantitatively analyze the accuracy of estimated point of gaze, an angular degree α_g [14] is calculated.

$$\alpha_g = arctan(D_{gg}/D_{ss}) \tag{20}$$

where D_{gg} denotes the distance between the estimated point of gaze and the groundtruth point, and D_{ss} is the distance between the subject and the screen plane. The smaller α_g is, the higher of estimation accuracy is.

By using the estimated eye model parameters in Table I, the angular degree of each gaze can be determined according to the proposed method. Table. II shows the average accuracy (angular degree) and tolerance of head movements of gazing nine ground truth points with ten different head poses or gaze directions of each subject under different gazing distances. The head movement tolerance involved in the table denotes the maximum rotation angles for yaw, pitch, and roll, respectively. From the experimental results, we can conclude that gaze estimation accuracy is sensitive to the gazing distance between the subject and the target as well as the degrees of head movements.

To further demonstrate the good performance of the proposed method, we compare the estimation accuracy of our

TABLE I: Estimated Eye Model Parameters of Five Subjects.

Subjects	$\mathbf{O}_e^{L,H}$	$\mathbf{O}_e^{R,H}$	r_e^L	r_e^R	$\{\alpha^L, \beta^L\}$	$\{\alpha^R, \beta^R\}$
1 (male)	$[34.23, 1.84, -9.61]^T$	$[-35.72, 3.56, -12.46]^T$	$10.73mm$	$10.46mm$	$\{-2.00^\circ, 3.17^\circ\}$	$\{-1.26^\circ, -1.58^\circ\}$
2 (male)	$[28.54, 2.60, -8.74]^T$	$[-27.61, 2.03, -9.37]^T$	$10.20mm$	$10.10mm$	$\{-2.27^\circ, 0.16^\circ\}$	$\{-2.97^\circ, -0.5^\circ\}$
3 (male)	$[32.58, 0.95, -10.28]^T$	$[-31.66, 1.24, -10.81]^T$	$10.62mm$	$10.55mm$	$\{-1.85^\circ, 1.56^\circ\}$	$\{-1.90^\circ, -2.28^\circ\}$
4 (female)	$[29.25, 1.37, -10.03]^T$	$[-28.68, 1.98, -11.62]^T$	$10.02mm$	$10.12mm$	$\{-1.79^\circ, 2.28^\circ\}$	$\{-2.13^\circ, -0.76^\circ\}$
5 (female)	$[29.86, 2.84, -9.25]^T$	$[-30.15, 3.17, -10.14]^T$	$9.86mm$	$10.05mm$	$\{-2.52^\circ, 2.70^\circ\}$	$\{-1.46^\circ, -1.59^\circ\}$

TABLE II: Average Estimated Gaze Accuracy and Tolerance of Head Movements.

Subjects	D_{ss}	Average accuracy	Head movements tolerance
1	90cm	3.61°	$14.2^\circ \times 13.7^\circ \times 6.6^\circ$
2	90cm	3.53°	$12.6^\circ \times 12.9^\circ \times 6.3^\circ$
3	100cm	4.12°	$12.8^\circ \times 12.5^\circ \times 6.1^\circ$
4	100cm	3.39°	$13.7^\circ \times 14.3^\circ \times 7.4^\circ$
5	110cm	4.27°	$13.5^\circ \times 13.8^\circ \times 7.2^\circ$
Average	n/a	3.78°	$13.36^\circ \times 13.44^\circ \times 6.72^\circ$

TABLE III: Comparison with the State-of-the-art Kinect-based Gaze Estimation Methods.

Methods	Reported accuracy	Features
Mora and Odobez[1]	7.6°-12.6°	Regression-based
Jafari and Ziou[8]	7.9°	Regression-based
Mora and Odobez[2]	6.3°	Model-based
Cazzato et al.[4]	6.9°	Model-based
Li and Li[5]	6°	Model-based
Li and Li[6]	Vertical 5.9°, horizontal 4.4°	Model-based
Sun et al.[7]	1.4°-2.7°	Model-based
Ours	3.78°	Model-based

method with the accuracy of the state-of-the-art Kinect-based gaze estimation methods. Results in Table III indicate that our method outperforms all the regression-based methods as well as most of the model-based methods.

V. Conclusion

In this paper, an effective gaze estimation method based on 3D eye model was presented. The proposed method was capable of estimating human's gaze directly from a Kinect sensor with free head movements. A convolution-based means of gradients iris center localization method was developed, which significantly improved the accuracy and speed of the conventional means of gradients method. Several reliable geometric constraints were employed for eye model parameters determination. Based on these constraints, both the eyeball centers and *Kappa* angle could be effectively calculated. The final human's gaze could be directly calculated according to the located iris center and estimated eye model parameters. Experiments conducted on five subjects demonstrated the good performance of the proposed gaze estimation method.

Although the proposed algorithm can simply estimate eyeball center with a series of iris centers, it is unstable and greatly relies on the estimation accuracy of iris centers. In our experiments, we picked up the best estimation results for iris centers for eyeball center fitting. Even though, the fitting results were not accurate enough and finally degraded the gaze estimation accuracy. Our future work will focus on exploring a more accurate eyeball center estimation method and incorporating the human's gaze with visual tracking methods [15], [16] to recognize human's activity.

References

[1] K. A. F. Mora and J. M. Odobez, "Gaze estimation from multimodal Kinect data," *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshop*, pp. 25–30, 2012.

[2] K. A. F. Mora and J. M. Odobez, "Geometric generative gaze estimation (G^3E) for remote RGB-D cameras," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1773–1780, 2014.

[3] M. J. Reale, P. Liu, L. Yin, and S. Canavan, "Art critic: multisignal vision and speech interaction system in a gaming context," *IEEE Trans. Cybern*, vol. 43, no. 6, pp. 1546–1559, 2013.

[4] D. Cazzato, M. Leo, and C. Distante, "An investigation on the feasibility of uncalibrated and unconstrained gaze tracking for human assistive applications by using head pose estimation," *Sensors*, vol. 2014, no. 14, pp. 8363–8379, 2014.

[5] J. Li and S. Li, "Eye-model-based gaze estimation by RGB-D camera," *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshop*, pp. 606–610, 2014.

[6] J. Li and S. Li, "Gaze estimation from color image based on the eye model with known head pose," *IEEE Trans. Human-Machine Systems*, vol. 46, no. 3, pp. 414–423, 2016.

[7] L. Sun, Z. Liu, and M.-T. Sun, "Real time gaze estimation with a consumer depth camera," *Information Sciences*, vol. 320, pp. 346–360, 2015.

[8] R. Jafari and D. Ziou, "Eye-gaze estimation under various head positions and iris states," *Expert Systems with Applications*, vol. 42, no. 1, pp. 510–518, 2015.

[9] E. Guestrin and E. Eizenman, "General theory of remote gaze estimation using the pupil center and corneal reflections," *IEEE Trans. Biomedical Engineering*, vol. 53, no. 6, pp. 1124–1133, 2006.

[10] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[11] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 532–539, 2013.

[12] G. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[13] F. Timm and E. Barth, "Accurate eye centre localisation by means of gradients," *Proc. 6th Int. Conf. Computer Vision Theory and Applications*, pp. 125–130, 2011.

[14] Y.-M. Cheung and Q. Peng, "Eye gaze tracking with a web camera in a desktop environment," *IEEE Trans. Human-Machine Systems*, vol. 45, no. 4, pp. 419–430, 2015.

[15] X. Zhou, H. Yu, H. Liu, and Y. Li, "Tracking multiple video targets with an improved gm-phd tracker," *Sensors*, vol. 15, no. 12, pp. 30 240–30 260, 2015.

[16] X. Zhou, Y. Li, B. He, and T. Bai, "GM-PHD-based multi-target visual tracking using entropy distribution and game theory," *IEEE Trans. Industrial Informatics*, vol. 10, no. 2, pp. 1064–1076, 2014.