Eye Gaze Tracking Under Natural Head Movements

Zhiwei Zhu and Qiang Ji
Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY,12180
{zhuz,jiq}@rpi.edu

Abstract

Most available remote eye gaze trackers based on Pupil Center Corneal Reflection (PCCR) technique have two characteristics that prevent them from being widely used as an important computer input device for human computer interaction. First, they must often be calibrated repeatedly for each individual; Second, they have low tolerance for head movements and require the user to hold the head uncomfortably still. In this paper, we propose a novel solution for the classical PCCR technique that will simplify the calibration procedure and allow free head movements. The core of our method is to analytically obtain a head mapping function to compensate head movement. Specifically, the head mapping function allows to automatically map the eye movement measurement under an arbitrary head position to a reference head position so that the gaze can be estimated from the mapped eye measurement with respect to the reference head position. Furthermore, our method minimizes the calibration procedure to only one time for each individual. Our proposed method will significantly improve the usability of the eye gaze tracking technology, which is a major step for eye tracker to be accepted as a natural computer input device.

1 Introduction

Eye Gaze is defined as the line of sight of a person. It represents a person's focus of attention. Eye gaze tracking has been an active research topic for many decades because of its potential usages in various applications such as Human Computer Interaction, Eye Disease Diagnosis, Human Behavior Study, etc. Earlier eye gaze trackers were fairly intrusive in that they require physical contacts with the user, such as placing a reflective white dot directly onto the eye [1] or attaching a number of electrodes around the eye [2]. Except the intrusive properties, most of these technologies also require the viewer's head to be motionless during eye tracking.

With the rapid technological advancements in both video cameras and microcomputers, eye gaze tracking technology based on the digital video analysis of eye movements has been widely explored. Since it does not require anything attached to the user, the video technology opens the most promising direction to build a non-intrusive eye gaze tracker. Based on the eye images captured by the video cameras, various techniques [3], [4], [5], [6], [7], [8], [9] have been proposed to do the eye gaze estimation. Yet most of them suffer from two problems, the need of calibration per user session and the large restriction on head motion [10].

Among them, the PCCR technique is the most commonly used approach to perform the video-based eye gaze tracking. The angle of the visual axis (or the location of the fixation point on the display surface) is calculated by tracking the relative position of the pupil center and a speck of light reflected from the cornea, technically known as "glint" as shown in Figure 1. The accuracy of the system can be further enhanced by illuminating the eyes with low-level Infrared light, which produces the "bright pupil" effect as shown Figure 1. It makes the video image easier to process.

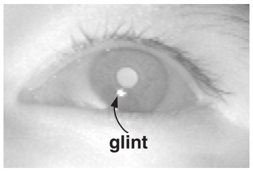


Fig. 1. Eye image with corneal reflection (or glint).

Several systems [11], [12], [3], [4], [10], [8] have been built based on the PCCR technique. Most of these systems show that if the user has the ability to keep his head fixed or via the help of a chin rest or bite bar to restrict the head motion, very high accuracy of the eye gaze tracking results can be achieved. Specifically, the average error can be less than 1° visual angle, which corresponds to less than 10 mm in the computer screen when the subject is sitting around 550 mm from the computer screen. But as the head moves away from the original position where the user performed the eye gaze calibration, the accuracy of these eye gaze tracking systems drops dramatically. Some detailed data [10] are reported about how the calibration mapping function decays as the head moves away from its original position. The current solution to overcome this problem is performing another new calibration whenever the head moves. This, however, is practically impossible since a person's head constantly moves. Therefore, the limited head movement is one of the worst issues in the current remote eye tracking systems. Jacob reports a similar fact in [12], and tried to solve the problem by giving the user the possibility of making local manual re-calibration, which brings numerous troubles for the user.

It is obvious that most of the existing gaze tracking systems based on PCCR technique share two common drawbacks: first, the user must perform certain experiments in calibrating the user-dependent parameters before using the gaze tracking system; second, the user must keep his head uncomfortably still, no significant head movements allowed.

In this paper, a solution is proposed to handle these two issues. First, before using the system, a calibration procedure

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

is performed for each user to obtain a gaze mapping function. After the calibration, the user does not need to calibrate the system again. Second, completely different from previous eye gaze trackers, the user can move his head freely in front of the camera, which will make the communications between the computer and user more naturally. Therefore, by using our gaze tracking technique, a more robust, accurate, comfortable and useful system can be built.

2 Classical PCCR Technique

The PCCR based technique consists of two major components: pupil-glint vector extraction and gaze mapping function acquisition.

1. **Pupil-glint Vector Extraction**

Gaze estimation starts with the pupil-glint vector extraction. After grabbing the eye image from the camera, computer vision techniques [9], [13] are proposed to extract the pupil center and the glint center robustly and accurately. The pupil center and the glint center is connected to form a 2D pupil-glint vector v as shown in Figure 3.

2. **Specific Gaze Mapping Function Acquisition**

After obtaining the pupil-glint vectors, a calibration procedure is proposed to acquire a specific gaze mapping function that will map the extracted pupil-glint vector to the user's fixation point in the screen for current head position. The extracted pupil-glint vector v is represented as (v_x, v_y) and the screen gaze point S_s is represented by (x_{gaze}, y_{gaze}) in the screen coordinate system. The specific gaze mapping function $S_s = f(v)$ can be modelled by the following nonlinear equations [8]:

$$\left.\begin{array}{lcl} x_{gaze} & = & a_0 + a_1 * v_x + a_2 * v_y + a_3 * v_x * v_y \\ y_{gaze} & = & b_0 + b_1 * v_x + b_2 * v_y + b_3 * {v_y}^2 \end{array}\right\} \quad (1)$$

The coefficients a_0, a_1, a_2, a_3 and b_0, b_1, b_2, b_3 are estimated from a set of pairs of pupil-glint vectors and the corresponding screen gaze points. These pairs are collected in a calibration procedure. During the calibration, the user is required to visually follow a shining dot as it displays at several predefined locations on the computer screen. In addition, he must keep his head as still as possible.

If the user does not move his head significantly after gaze calibration, the specific gaze mapping function can be used to estimate the user's gaze point in the screen accurately based on the extracted pupil-glint vector. But when the user moves his head away from the position where the specific gaze calibration is performed, the specific gaze mapping function will fail to estimate the gaze point accurately because of the pupil-glint vector changes caused by the head movement. In the following, the head movement effects on the pupil-glint vector will be illustrated.

2.1 Head Motion Effects on Pupil-glint Vector

Figure 2 shows the ray diagram of the pupil-glint vector generation in the image when an eye is located at two different 3D positions O_1 and O_2 in front of the camera due to head movement. For simplicity, the eye is represented by a cornea, the cornea is modelled as a convex mirror and the IR light source used to generate the glint is located at O, which are applicable to all the subsequent figures. Assume that the origin of the camera is located at O, p_1 and p_2 are the pupil centers and g_1 and g_2 are the glint centers generated in the image. Further, at both positions, the user is looking at the same point of the computer screen S. According to the light ray diagram shown in Figure 2, the generated pupil-glint vectors $\overrightarrow{g_1 p_1}$ and $\overrightarrow{g_2 p_2}$ will be significantly different in the images as shown in Figure 3. Two factors are responsible for this pupil-glint vector difference: first, the eyes are at different positions in front of the camera; second, in order to look at the same screen point, eyes at different positions rotate themselves differently.

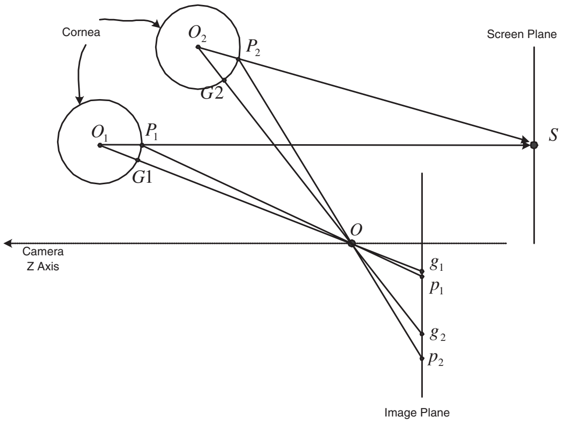
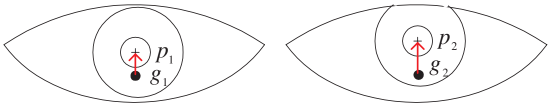


Fig. 2. Pupil and glint image formations when eyes are located at different positions while gazing at the same screen point (side view).



(a) eye image at location O_1 (b) eye image at location O_2

Fig. 3. The pupil-glint vectors generated in the eye images when the eye is located at O_1 and O_2 in Figure 2.

The eye will move as the head moves. Therefore, when the user is gazing at a fixed point on the screen while moving his head in front of the camera, a set of pupil-glint vectors in the image will be generated. These pupil-glint vectors are significantly different from each other. If uncorrected, inaccurate gaze points will be estimated after inputting them into the specific gaze mapping function obtained at the reference eye position.

Therefore, the head movement effects on these pupil-glint vectors must be eliminated in order to utilize the specific gaze mapping function to estimate the screen gaze points accurately. In the following, a technique is proposed to eliminate the head movement effects on these pupil-glint vectors before inputting them into the specific gaze mapping function. With this technique, accurate gaze screen points can be estimated accurately whether the users has moved his head or not.

3 Dynamic Head Compensation Model

3.1 Approach Overview

The first step of our technique is to find a specific gaze mapping function f_{O_1} between the pupil-glint vector v_1 and the screen coordinate S at a reference 3D eye position O_1. This is usually achieved via a gaze calibration procedure using equations 1. The function f_{O_1} can be expressed as follows:

$$S = f_{O_1}(v_1) \tag{2}$$

Assume that when the eye moves to a new position O_2 as the head moves, a pupil-glint vector v_2 will be created in the image while the user is looking at the same screen point S. When O_2 is significantly different from O_1, v_2 can not be used as the input of the gaze mapping function f_{O_1} to estimate the screen gaze point due to the changes of the pupil-glint vector caused by the head movement. If the changes of the pupil-glint vector v_2 caused by the head movement can be eliminated, then a corrected pupil-glint vector ${v_2}^{'}$ will be obtained. Ideally, this corrected pupil-glint vector ${v_2}^{'}$ is the generated pupil-glint vector v_1 of the eye at the reference position O_1 when gazing at the same screen point S. Therefore, this is equivalent to finding a head mapping function g between two different pupil-glint vectors at two different head positions when still gazing at the same screen point. This mapping function g can be written as follows:

$${v_2}^{'} = g(v_2, O_2, O_1) \tag{3}$$

where ${v_2}^{'}$ is the equivalent measurement of v_1 with respect to the initial reference head position O_1. Therefore, the screen gaze point can be estimated accurately from the pupil-glint vector ${v_2}^{'}$ via the specific gaze mapping function f_{O_1} as follows:

$$S = f_{O_1}(g(v_2, O_2, O_1)) = F(v_2, O_2) \tag{4}$$

where the function F can be called as a generalized gaze mapping function or a computational dynamic head compensation model. It provides the gaze mapping function for a new eye position O_2 dynamically.

With the use of the proposed technique, whenever the head moves, a gaze mapping function at each new 3D eye position will be estimated automatically, therefore, the issue of the head movement can be solved through g.

3.2 Head Movement Compensation

In this section, we show how to find the head mapping function g. Figure 4 shows the process of the pupil-glint vector formation in the image for an eye in front of the camera. When the eye is located at two different positions O_1 and O_2 while still gazing at the same screen point S, two different pupil-glint vectors $\overrightarrow{g_1p_1}$ and $\overrightarrow{g_2p_2}$ are generated in the image. Further, as shown in Figure 4, a parallel plane A of the image plane that goes through the point P_1 will intersect the line O_1O at G_1[1]. Another parallel plane B of the image plane that goes through the point P_2 will intersect the line O_2O at G_2[2]. Therefore, $\overrightarrow{g_1p_1}$ is the projection of the vector $\overrightarrow{G_1P_1}$ and $\overrightarrow{g_2p_2}$ is the projection of the vector $\overrightarrow{G_2P_2}$ in the image plane. Because plane A, plane B and the image plane are parallel, the vectors $\overrightarrow{g_1p_1}$, $\overrightarrow{g_2p_2}$, $\overrightarrow{G_1P_1}$ and $\overrightarrow{G_2P_2}$ can be represented as 2D vectors in the $X - Y$ plane of the camera coordinate system.

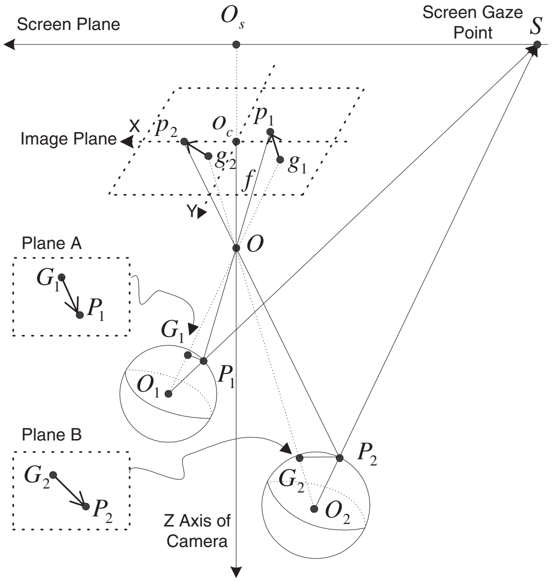


Fig. 4. Pupil and glint image formation when the eye is located at different positions in front of the camera.

3.2.1 Image Projection of Pupil-glint Vector

Assume that in the camera coordinate system, the 3D pupil centers P_1 and P_2 are represented as (x_1, y_1, z_1) and (x_2, y_2, z_2), the glint centers g_1 and g_2 are represented as $(x_{g_1}, y_{g_1}, -f)$ and $(x_{g_2}, y_{g_2}, -f)$, where f is focus length of the camera, and the screen gaze point S is represented by (x_s, y_s, z_s). Via the pinhole camera model, the image projection of the pupil-glint vectors can be expressed as follows:

$$\overrightarrow{g_1p_1} = -\frac{f}{z_1} * \overrightarrow{G_1P_1} \tag{5}$$

$$\overrightarrow{g_2p_2} = -\frac{f}{z_2} * \overrightarrow{G_2P_2} \tag{6}$$

Assume that the pupil-glint vectors $\overrightarrow{g_1p_1}$ and $\overrightarrow{g_2p_2}$ are represented as (v_{x1}, v_{y1}) and (v_{x2}, v_{y2}) respectively, and the vectors $\overrightarrow{G_1P_1}$ and $\overrightarrow{G_2P_2}$ are represented as (V_{x1}, V_{y1}) and (V_{x2}, V_{y2}) respectively. Therefore, the following equation can be derived by combining the equations 5 and 6:

[1] G_1 is not the actual virtual image of the IR light source

[2] G_2 is not the actual virtual image of the IR light source

$$v_{x1} = \frac{V_{x1}}{V_{x2}} * \frac{z_2}{z_1} * v_{x2} \tag{7}$$

$$v_{y1} = \frac{V_{y1}}{V_{y2}} * \frac{z_2}{z_1} * v_{y2} \tag{8}$$

The above two equations describe how the pupil-glint vector changes as the head moves in front of the camera. Also, based on the above equations, it is obvious that each component of the pupil-glint vector can be transformed individually. Therefore, equation 7 for the X component of the pupil-glint vector will be derived first as follows.

3.2.2 First Case: the cornea center and the pupil center lie on the camera's $X - Z$ plane:

Figure 5 shows the ray diagram of the pupil-glint vector formation when the cornea center and pupil center of an eye happen to lie on the $X - Z$ plane of the camera coordinate system. Therefore, either the generated pupil-glint vectors $\overrightarrow{p_1 g_1}$ and $\overrightarrow{p_2 g_2}$ or the vectors $\overrightarrow{P_1 G_1}$ and $\overrightarrow{P_2 G_2}$ can be represented as one dimensional vectors, specifically, $\overrightarrow{p_1 g_1} = v_{x1}$, $\overrightarrow{p_2 g_2} = v_{x2}$, $\overrightarrow{P_1 G_1} = V_{x1}$ and $\overrightarrow{P_2 G_2} = V_{x2}$.

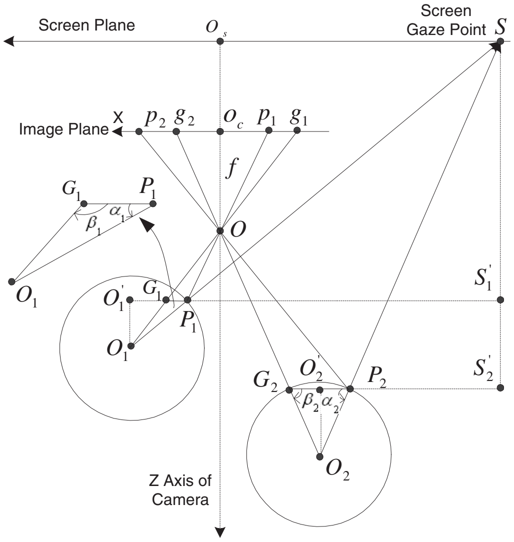


Fig. 5. Pupil and glint image formation when the eye is located at different positions in front of the camera (Top-down view).

According to Figure 5, the vectors $\overrightarrow{G_1 P_1}$ and $\overrightarrow{G_2 P_2}$ can be represented as follows:

$$\overrightarrow{G_1 P_1} = \overrightarrow{G_1 O_1'} + \overrightarrow{O_1' P_1} \tag{9}$$

$$\overrightarrow{G_2 P_2} = \overrightarrow{G_2 O_2'} + \overrightarrow{O_2' P_2} \tag{10}$$

For simplicity, r_1 is used to represent the length of $\overrightarrow{O_1 P_1}$, r_2 is used to represent the length of $\overrightarrow{O_2 P_2}$, $\angle G_1 P_1 O_1$ is represented as $\alpha 1$, $\angle G_2 P_2 O_2$ is represented as $\alpha 2$, $\angle P_1 G_1 O_1$ is represented as $\beta 1$ and $\angle P_2 G_2 O_2$ is represented as $\beta 2$. According to the geometries shown in Figure 5, the vectors $\overrightarrow{G_1 P_1}$ and $\overrightarrow{G_2 P_2}$ can be further achieved as follows:

$$\overrightarrow{G_1 P_1} = -\frac{r_1 * sin(\alpha 1)}{tan(\beta 1)} - r_1 * cos(\alpha 1) \tag{11}$$

$$\overrightarrow{G_2 P_2} = -\frac{r_2 * sin(\alpha 2)}{tan(\beta 2)} - r_2 * cos(\alpha 2) \tag{12}$$

As shown in Figure 5, line $G_1 P_1$ and line $G_2 P_2$ are parallel to the X axis of the camera. Therefore, $tan(\beta 1)$ and $tan(\beta 2)$ can be obtained from the rectangles $g_1 o_c O$ and $g_2 o_c O$ individually as follows:

$$tan(\beta 1) = \frac{f}{\overrightarrow{o_c g_1}} \tag{13}$$

$$tan(\beta 2) = \frac{f}{\overrightarrow{o_c g_2}} \tag{14}$$

In the above equation, g_1 and g_2 are the glints in the image, and o_c is the principal point of the camera. For simplicity, we choose x_{g_1} to represent $\overrightarrow{o_c g_1}$ and x_{g_2} to represent $\overrightarrow{o_c g_2}$. Therefore, after detecting the glints in the image, $tan(\beta 1)$ and $tan(\beta 2)$ can be obtained accurately.

Further, $sin(\alpha 1)$, $cos(\alpha 1)$, $sin(\alpha 2)$ and $cos(\alpha 2)$ can be obtained from the geometries of the rectangles $P_1 S S_1'$ and $P_2 S S_2'$ directly. Therefore, equations 11 and 12 can be derived as follows:

$$V_{x1} = r_1 * \frac{(z_s - z_1) * x_{g_1}}{P_1 S * f} + r_1 * \frac{(x_s - x_1)}{P_1 S} \tag{15}$$

$$V_{x2} = r_2 * \frac{(z_s - z_2) * x_{g_2}}{P_2 S * f} + r_2 * \frac{(x_s - x_2)}{P_2 S} \tag{16}$$

3.2.3 Second Case: the cornea center and the pupil center do not lie on the camera's $X - Z$ plane:

However, the cornea center and the pupil center do not always lie on the camera's $X - Z$ plane, we can obtain the ray diagram shown in Figure 5 by projecting the ray diagram in Figure 4 into $X - Z$ plane along the Y axis of the camera's coordinate system. Therefore, as shown in Figure 6, point P_1 is the projection of the pupil center P_{c1}, point O_1 is the projection of the cornea center O_{c1}, and point S is also the projection of the screen gaze point $S^{'}$ in the $X - Z$ plane. Starting from O_{c1}, a parallel line $O_{c1} {P_1}^{'}$ of line $O_1 P_1$ intersects with line $P_{c1} P_1$ at ${P_1}^{'}$. Also starting from P_{c1}, a parallel line $P_{c1} {O_1}^{''}$ of line $P_1 S$ intersects with line $SS^{'}$ at ${O_1}^{''}$.

Because $O_{c1} P_{c1}$ represents the distance r between the pupil center to the cornea center, which will not change as the eyeball rotates, $O_1 P_1$ can be derived as follows:

$$r_1 = O_1 P_1 = r * \frac{P_1 S}{\sqrt{P_1 S^2 + (y_1 - y_s)^2}} \tag{17}$$

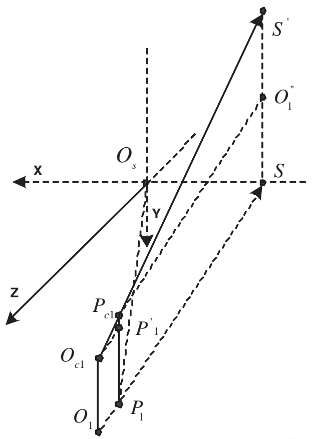


Fig. 6. Projection into camera's $X-Z$ plane.

Therefore, when the eye moves to a new location O_2 as shown in Figure 5, O_2P_2 can be represented as follows:

$$r_2 = O_2P_2 = r * \frac{P_2S}{\sqrt{P_2S^2 + (y_2 - y_s)^2}} \tag{18}$$

After substituting the formulations of r_1 and r_2 into equations 15 and 16, we can obtain $\frac{V_{x1}}{V_{x2}}$ as follows:

$$\frac{V_{x1}}{V_{x2}} = d * \frac{[(z_s - z_1) * x_{g_1} + (x_s - x_1) * f]}{[(z_s - z_2) * x_{g_2} + (x_s - x_2) * f]} \tag{19}$$

where d is set as follows:

$$d = \frac{\sqrt{(z_2 - z_s)^2 + (x_2 - x_s)^2 + (y_2 - y_s)^2}}{\sqrt{(z_1 - z_s)^2 + (x_1 - x_s)^2 + (y_1 - y_s)^2}}$$

As a result, equations 7 and 8 can be finally obtained as follows:

$$v_{x1} = d * \frac{[(z_s - z_1) * x_{g_1} + (x_s - x_1) * f]}{[(z_s - z_2) * x_{g_2} + (x_s - x_2) * f]} * \frac{z_2}{z_1} * v_{x2} \tag{20}$$

$$v_{y1} = d * \frac{[(z_s - z_1) * y_{g_1} + (y_s - y_1) * f]}{[(z_s - z_2) * y_{g_2} + (y_s - y_2) * f]} * \frac{z_2}{z_1} * v_{y2} \tag{21}$$

The above equations constitute the head mapping function g between the pupil-glint vectors of the eyes at different positions in front of the camera, while gazing at the same screen point.

3.3 Iterative Algorithm for Gaze Estimation

Actually, the derived equations 20 and 21 of the head mapping function can not be established unless the gaze point $S = (x_s, y_s, z_s)$ of the screen is known. However, the gaze point S is the one that needs to be estimated. As a result, the gaze point S is also a variable of the head mapping function g, which can be further expressed as follows:

$$v_2{}' = g(v_2, P_2, P_1, S) \tag{22}$$

Assume that a specific gaze mapping function f_{P_1} is known via the calibration procedure described in Section 2. Therefore, after integrating the head mapping function g into the specific gaze mapping function f_{P_1} via equation 4, the generalized gaze mapping function F can be rewritten as follows:

$$S = F(v_2, P_2, S) \tag{23}$$

Given the extracted pupil-glint vector v_2 from the eye image and the new location P_2 that the eye has moved to, equation 23 becomes a recursive function. An iterative solution is proposed to solve it.

First, the screen center S_0 is chosen as an initial gaze point, then a corrected pupil-glint vector $v_2{}'$ can be obtained from the detected pupil-glint vector v_2 via the head mapping function g. By inputting the corrected pupil-glint vector $v_2{}'$ into the specific gaze mapping function f_{P_1}, a new screen gaze point S' can be estimated. S' is further used to compute a new corrected pupil-glint vector $v_2{}'$. The loop continues until the estimated screen gaze point S' does not change any more. Usually, the whole iteration process will converge in less than 5 loops, which is very fast. By the proposed method, the gaze point can be estimated accurately.

4 Experiment Results

4.1 System Setup

Two cameras are mounted under the monitor screen. An IR light illuminator is mounted at the center of the lens of one camera, which will produce the corneal glint in the eye image. The pupil-glint vectors extracted from the eye images captured by this camera will be used for gaze estimation. Since our algorithm needs the 3D eye position to work, another camera is mounted close to the one with IR illuminator. These two cameras form a stereo vision system so that the 3D eye position can be obtained accurately.

One requirement for our eye tracker system is to know the 3D representation of the monitor screen plane in the coordinate system of the camera with the IR illuminator. This is accomplished via the calibration method proposed in [14]. After the calibration, the configuration between the camera and the monitor is fixed during the eye tracking.

4.2 Head Compensation Model Validation

The equations 20 and 21 of the head mapping function g are validated by the following experiment.

A screen point $S_c = (132.75, -226.00, -135.00)$ is chosen as the gaze point. The user was gazing at this point from twenty different locations in front of the camera; at each location, the pupil-glint vector and the 3D pupil center are collected. The 3D pupil centers and the pupil-glint vectors of the first two samples P_1,P_2 are shown in Table I, where P_1 is served as the reference position. The second column indicates the original pupil-glint vectors, while the third column indicates the transformed pupil-glint vectors by the head mapping function. The difference between the transformed pupil-glint vector of P_2 and the reference pupil-glint vector at P_1 is defined as the transformation error. Figure 7 illustrates the transformation errors for all these twenty samples. It's observed that the average transformation error is only around 1 pixel. This shows the validity of the proposed head compensation model.

TABLE I

PUPIL-GLINT VECTOR COMPARISON AT DIFFERENT EYE LOCATIONS

3D pupil position (mm.)	2D Pupil-glint Vector (pixel)	Transformed Pupil-glint Vector (pixel)
P_1(5.25, 15.56, 331.55)	(9.65, -16.62)	(9.65, -16.62)
P_2(-8.13, 32.29, 361.63)	(7.17, -13.33)	(8.75, -16.01)

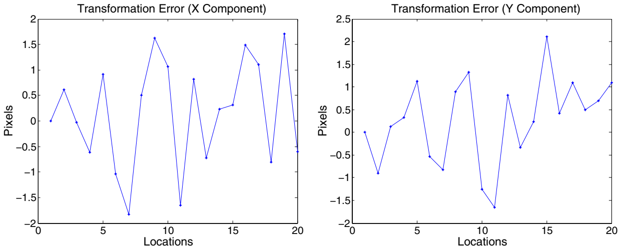


Fig. 7. The pupil-glint vector transformation errors.

4.3 Gaze Estimation Accuracy

When the user moves away from the camera, the eye in the image becomes smaller. Due to the increased pixel measurement error caused by the lower image resolution, the gaze accuracy of the eye gaze tracker will decrease as the user moves away from the camera. In this experiment, the effect of the distance to the camera on the gaze accuracy of our system is analyzed. A user was asked to perform the gaze calibration when he was sitting around 330 mm to the camera. After the calibration, the user was positioned at four different locations, which have different distances to the camera as listed in Table II. At each location, the user was asked to follow the moving objects that will display 12 predefined positions across the screen. Table II lists the computed gaze estimation accuracy at these four different locations. It shows that as the user moves away from the camera, the gaze resolution will decrease. Yet within this space allowed for the head movement, approximately 200 × 200 × 300 mm (width×height×depth) at 450 mm from the camera, the average horizontal angular accuracy is around 1.3° and the average vertical angular accuracy is around 1.7°, which is acceptable for most of the Human Computer Interaction applications. In addition, this space volume allowed for the head movement is large enough for a user to sit comfortably in front of the camera and communicate with the computer naturally.

5 Conclusion

In this paper, a solution is proposed to improve the classical PCCR eye gaze tracking technique. While enjoying the advantages of high accuracy and stability of classical PCCR technique, we make it work under natural head movements. Therefore, the user does not need to keep his head uncomfortably still as most of eye trackers require. Instead, the user can move his head freely while using our eye tracking system. At the same time, the number of calibration procedures is minimized to only once for a new user. Also, the calibration result is long-lasting: if the user repositions his head or he wants to come back and use the system next time, he does not need to do the calibration again. Both achievements are made possible because of the proposed head mapping function, which can automatically accommodate the head movement changes.

TABLE II

GAZE ESTIMATION ACCURACY

Distance to the camera (mm)	Horizontal Accuracy (degrees)	Vertical Accuracy (degrees)
340.26	0.68	0.83
400.05	1.31	1.41
462.23	1.54	1.90
552.51	1.73	2.34

References

[1] S. Milekic, "The more you look the more you get: intention-based interface using gaze-tracking," in *Bearman, D., Trant, J.(des.) Museums and the Web 2002: Selected papers from an international onference, Archives and Mseum Informatics*, 2002.

[2] K. Hyoki, M. Shigeta, N. Tsuno, Y. Kawamuro, and T. Kinoshita, "Quantitative electro-oculography and electroencephalography as indices of alertness," in *Electroencephalography and Clinical Neurophysiology*, 1998, pp. 213–219.

[3] Y.Ebisawa, M. Ohtani, and A. Sugioka, "Proposal of a zoom and focus control method using an ultrasonic distance-meter for video-based eye-gaze detection under free-hand condition," in *Proceedings of the 18th Annual International conference of the IEEE Eng. in Medicine and Biology Society*, 1996.

[4] C.H. Morimoto, D. Koons, A. Amir, and M. Flickner, "Frame-rate pupil detector and gaze tracker," in *IEEE ICCV'99 Frame-rate Workshop*, 1999.

[5] A. T. Duchowski, "Eye tracking methodology: Theory and practice," in *Spring Verlag*, 2002.

[6] R.J.K. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," 2003, Oxford, Elsevier Science.

[7] D. H. Yoo, "Non-contact eye gaze estimation system using robsut feature," *CVIU, special Issue on eye detection and tracking*, 2005.

[8] LC Technologies Inc, "The eyegaze development sytem, http://www.eyegaze.com," .

[9] Z. Zhu and Q. Ji, "Eye and gaze tracking for interactive graphic display," *Machine Vision and Applications*, vol. 15, no. 3, pp. 139–148, 2004.

[10] C.H. Morimoto and M. R.M. Mimica, "Eye gaze tracking techniques for intractive applications," *CVIU, special issue on eye detection and tracking*, 2005.

[11] T.E. Hutchinson, K.P. White Jr., K.C. Reichert, and L.A. Frey, "Human-computer interaction using eye-gaze input," in *IEEE Transactions on Systems, Man, and Cybernetics*, 1989, pp. 1527–1533.

[12] R.J.K. Jacob, "Eye-movement-based human-computer interaction techniques: Towards non-command interfaces," 1993, pp. 151–190, Ablex Publishing corporation, Norwood, NJ.

[13] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *CVIU, special issue on eye detection and tracking*, 2005.

[14] S. W. Shih and J. Liu, "A novel approach to 3-d gaze tracking using stereo cameras," in *IEEE Trans. Syst. Man and Cybern., part B*, 2004, number 1, pp. 234–245.