



# A Monocular Reflection-Free Head-Mounted 3D Eye Tracking System

Shihao Cao<sup>1</sup>, Xinbo Zhao<sup>1,3</sup>(✉), Beibei Qin<sup>1</sup>, Junjie Li<sup>1</sup>, and Zheng Xiang<sup>2</sup>

<sup>1</sup> School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China

{caoshihao, 2019262385, 2019202143}@mail.nwpu.edu.cn

<sup>2</sup> School of Software, Northwestern Polytechnical University, Xi'an 710129, China  
zxixiang@mail.nwpu.edu.cn

<sup>3</sup> Ningbo Institute of Northwestern Polytechnical University, Ningbo 315103, China  
xbozhao@nwpu.edu.cn

**Abstract.** Head-mounted eye tracking has significant potential for gaze based application such as consumer attention monitoring, human-computer interaction, or virtual reality (VR). Existing methods, however, either use pupil center-corneal reflection (PCCR) vectors as gaze directions or require complex hardware setups and use average physiological parameters of the eye to obtain gaze directions. In view of this situation, we propose a novel method which uses only a single camera to obtain gaze direction by fitting a 3D eye model based on the motion trajectory of pupil contour. Then a 3D to 2D mapping model is proposed based on the fitting model, so the complex structure of hardware and the use of average parameters for the eyes are avoided. The experimental results show that the method can improve the gaze accuracy and simplify the hardware structure.

**Keywords:** 3D gaze estimation · Single camera · Head-mounted device · Pupil contour · Mapping model

## 1 Introduction

Over the past decades, eye tracking systems have become a widely used tool in fields such as marketing research [1], psychological studying [2, 3] and human-computer interaction [4, 5]. Recently, eye tracking has also been applied to virtual reality and augmented reality devices for control [6] and panoramic rendering [7, 8]. Currently, commercially available head-mounted eye-tracking devices are expensive such as tobii and Google Glasses, thus designing a head-mounted eye-tracking system with simple hardware structure and low cost is of great significance to researchers for research in related fields.

Typically, the methods of head-mounted eye tracking systems are divided into 2D and 3D according to the features of eye movement changes used, and the 2D methods are simpler in hardware structure compared to the 3D methods. The

2D approaches use 2D eye movement features as input to construct a mapping model to obtain the location of the human eye gaze point. Takemura et al. [9] and Carlos et al. [10] use a camera and an infrared light source to obtain the pupil center and spot to form a pupil center-corneal reflection vector, which was fitted by a polynomial to obtain the gaze position. Their method is the most common method for head-mounted eye tracking systems because of its good accuracy and relatively simple hardware structure, but their method has poor accuracy at non-calibrated points. Arar et al. [11] used four fixed infrared lights to extract the pupil center as well as four light spots to obtain the location of the gaze point based on the geometric principle of cross-ratio invariance. Their method requires four infrared light sources in order to obtain the reflected position of the corneal spot, which makes the hardware structure complex and has poor practicality. Moreover, the common disadvantage of the 2D methods is that the 2D features used do not make full use of the information of gaze direction changes, and therefore have poor accuracy at non-calibrated points.

In contrast to the 2D methods, the 3D methods directly obtain gaze direction and calculate the intersection with the scene to estimate the gaze point location based on the structural characteristics of the eye. However, most 3D methods rely on measurement information that is calibrated in advance such as light source position coordinates, camera position coordinates, screen position coordinates, and other information, causing great inconvenience to the use of eye tracking systems. Shih et al. [12] used two cameras and two light sources to calculate the human optics axis direction directly, avoiding the system calibration process and calibration errors. Nevertheless, this method requires multiple cameras to be calibrated and the position of the light sources need to be pre-set. Roma et al. [13] constructed a 3D eye model by considering the pupil and the radius of the eye as known quantities, and the direction of the line from the center of the eye to the center of the pupil as the visual axis. Their method ignores the physiological differences between different users. Zhu Z et al. [14] used two cameras and two infrared light sources to calculate corneal and pupillary parameters to obtain the direction of human eye gaze. Their method has the same disadvantages with Shih's [12].

In summary, for head-mounted eye tracking systems, existing methods are usually 2D methods using pupil center-corneal reflection vectors for interpolation or 3D methods using three-dimensional eye models. The disadvantage of the 2D method is that pupil center-corneal reflection vectors as a feature does not take full advantage of the information on the change in line of sight, resulting in poor accuracy on non-calibration. The disadvantage of the 3D method is that it usually requires advance calibration of the position relationship of the camera, IR light source or uniform modeling of the eye ignoring individual differences. Moreover, it has a complex hardware structure and high production cost. Hence, we need a lightweight head-mounted eye tracker. Swirski et al. [15] proposed a method to recover 3D eyeball from a monocular, but they only evaluated the model for synthetic eye images in a simulation environment, and the realistic performance of eye-to-scene camera has never been quantified. In order to solve

the problems of existing head-mounted eye tracking systems, this paper proposes a monocular reflection-free head-mounted 3D eye tracking system. Compared with existing methods, our method requires only one camera, does not use the average physiological parameters of the eye, and is able to improve accuracy at non-calibrated points.

The contributions of this work are threefold. First, an eye model is proposed that is more applicable to real-time human eye videos captured by eye cameras rather than just synthetic images in a simulation environment. Secondly, a mapping model from the 3D gaze direction vectors to the 2D plane is proposed, using the gaze direction angle instead of the PCCR to do the interpolation. Experimental results show that the method proposed in this paper has better accuracy. Finally, this paper designs a low-cost, simple hardware structure head-mounted eye-tracking system, which provides great convenience for research in related fields.

## 2 3D Eye Model

### 2.1 Computational Model of Eye Center

The model proposed by Swirski et al. [15] is based on two assumptions: (1) the apparent pupil contour in 2D eye image is a perspective projection of a 3D pupil circle  $P$  which is tangent to the eyeball of fixed radius,  $R$ . (2) the center of the eyeball is stationary over time. In their model, the gaze direction varies with the motion of the 3D pupil circle  $P$  on the eyeball surface. At each time point, the state of the eye model is determined by eye center  $c$  and the 3D pupil circle  $P$ .

Given a set of  $N$  eye images, recorded over a period of time, pupil contours are extracted from each image by means of an automatic pupil extraction algorithm [16–18], leading to sets of two-dimensional contour edges  $\varepsilon_i = \{e_{ij}, j = 1, \dots, M_i\}$ . Firstly, the edges  $\varepsilon_i$  of the contours on each image are fitted to ellipses  $l_i$ . Next, assuming a pinhole camera model for perspective projection, the inverse projection of the pupil ellipse produces two 3D circles when fixing an arbitrary size of radius  $r$  [19]. Two 3D circles can be obtained by unprojection, and these two circles are denoted as:

$$(\mathbf{p}^+, \mathbf{n}^+, r), (\mathbf{p}^-, \mathbf{n}^-, r) \quad (1)$$

where  $\mathbf{p}^+$  and  $\mathbf{p}^-$  denote the centers of the circles and  $\mathbf{n}^+$  and  $\mathbf{n}^-$  denote the normals of the circles. For the two circles obtained by unprojection of each pupil ellipse, Swirski et al. [15] removed the ambiguity by projecting the 3D vectors into the 2D image space. Because the normal of two circles in the image space are parallel:

$$\tilde{\mathbf{n}}_i^+ \propto \tilde{\mathbf{n}}_i^- \quad (2)$$

Similarly, the line between  $\tilde{\mathbf{p}}_i^+$  and  $\tilde{\mathbf{p}}_i^-$  is parallel to  $\tilde{\mathbf{n}}_i^\pm$ . The Eq. (3) can be derived:

$$\exists s, t \in R \cdot \tilde{\mathbf{p}}_i^+ = \tilde{\mathbf{p}}_i^- + s\tilde{\mathbf{n}}_i^+ = \tilde{\mathbf{p}}_i^- + t\tilde{\mathbf{n}}_i^- \quad (3)$$

which means that you can choose either one of the two circles for this stage and calculate the projection of eyeball center  $\tilde{\mathbf{c}}$  by computing the intersection of the normal vectors. These vectors may have numerical or measurement errors and therefore will almost never intersect at a single point. Thus we can find the point with the smallest sum of distances from each line by the least squares method.

$$\tilde{\mathbf{c}} = \left( \sum_i (\mathbf{I} - \tilde{\mathbf{n}}_i \tilde{\mathbf{n}}_i^T) \right)^{-1} \cdot \left( \sum_i (\mathbf{I} - \tilde{\mathbf{n}}_i \tilde{\mathbf{n}}_i^T) \tilde{\mathbf{p}}_i \right) \quad (4)$$

The limitation of this approach is that only the eye model is fitted on the synthetic image sequence, while the situation in the real-time video will be more complex compared to the synthetic image.

There are two differences between performing pupil detection on video frames captured by an eye camera and pupil detection on a synthetic image: (1) The pupil outline on the synthetic image is distinct, while in the video frame the pupil outline may be blurred due to motion blur. (2) The pupil contour on the synthetic image is complete, whereas on the video frame it may be incomplete due to blinking, eyelash occlusion, or excessive eye rotation. In this case, the pupil contour may be partially or even completely obscured, resulting in low accuracy of the fitted ellipse.

Swirski et al. [17] obtained the projection of eyeball center by projecting the normal vector of the circle into the image space and then solving for the intersection of clusters of lines using least squares. In their method all projections of the normal vector are used to calculate  $\tilde{\mathbf{c}}$ . However, when excessive eye rotation or incomplete pupil contours are encountered, the distance between the normal vector of the ellipse and  $\tilde{\mathbf{c}}$  may be too large, as in Fig. 1. Thus, this paper proposes an optimization algorithm to calculate the position of the center of the eye  $\mathbf{c}$ .

We can calculate  $N$  lines from  $N$  images by Eq. ((3)), denoted as  $L^N$ . Then  $M$  line are randomly selected from  $L^N$  to obtain  $L^M$ , Eq. ((4)) can be rewritten as ((8)) for this stage.

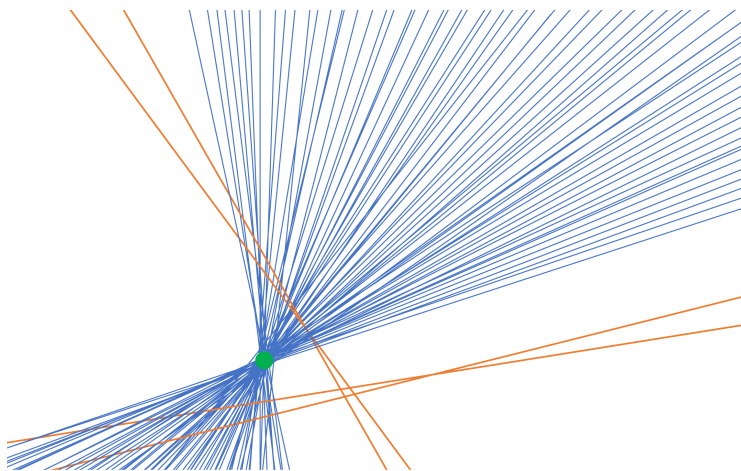
$$L^N = \{\tilde{\mathbf{n}}_i, i = 1, \dots, N\} \quad (5)$$

$$\{M\} = \text{random}(\{N\}) \quad (6)$$

$$L^M = \{\tilde{\mathbf{n}}_j, j = 1, \dots, M\} \quad (7)$$

$$\tilde{\mathbf{c}}_m = \left( \sum_j (\mathbf{I} - \tilde{\mathbf{n}}_j \tilde{\mathbf{n}}_j^T) \right)^{-1} \cdot \left( \sum_j (\mathbf{I} - \tilde{\mathbf{n}}_j \tilde{\mathbf{n}}_j^T) \tilde{\mathbf{p}}_j \right) \quad (8)$$

where  $\tilde{\mathbf{c}}_m$  is the coordinates of eye center in the image space calculated by the iterative algorithm. Then we count the number of lines whose distance from  $\tilde{\mathbf{c}}_m$  is within the given threshold, repeat Eqs. (6)–(8) and select the intersection with the largest number of lines among all results. We calculate the intersection points again for those lines whose obtained results are within the threshold condition and compare them with the original results until the results are not changing. We then unproject  $\tilde{\mathbf{c}}_m$  to find 3D eyeball center  $\mathbf{c}$  by fixing the  $z$  coordinate of  $\mathbf{c}$ .



**Fig. 1.** The green point is the theoretical projection position of the center of the eye. The orange line is the normal vector projection when the eye is overrotated or the pupil contour is incomplete. (Color figure online)

## 2.2 Calculating the Radius of the Eye

Once we obtain the coordinates of eyeball center  $\mathbf{c}$ , we note that the normal  $\mathbf{n}_i$  of each pupil has to point away from eyeball center  $\mathbf{c}$ :

$$\mathbf{n}_i \cdot (\mathbf{p}_i - \mathbf{c}) > 0 \quad (9)$$

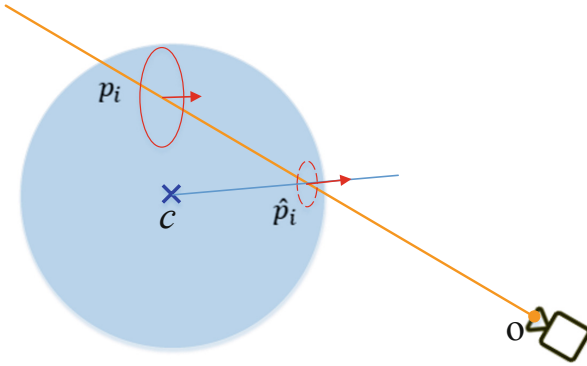
Therefore, when projected into the image space,  $\tilde{\mathbf{n}}_i$  has to point away from the projected center  $\tilde{\mathbf{c}}$ :

$$\tilde{\mathbf{n}}_i \cdot (\tilde{\mathbf{p}}_i - \tilde{\mathbf{c}}) > 0 \quad (10)$$

The pupil is tangent to the eye in the assumptions of Sect. 2.1, and the eye radius  $R$  can be estimated after obtaining the correct pupil projection. Since the unprojection of pupil has a distance ambiguity, we cannot use  $\mathbf{p}_i$  directly to calculate  $R$ . Thus, we consider a candidate location  $\hat{\mathbf{p}}_i$  for the pupil center that is different from  $\mathbf{p}_i$ , which is another possible unprojection of  $\tilde{\mathbf{p}}_i$  at a different distance. This means that  $\hat{\mathbf{p}}_i$  is on the line passing through camera center and  $\mathbf{p}_i$ , meanwhile the pupil circle is tangent to the eye, the line passing through  $\mathbf{c}$  and parallel to  $\mathbf{n}_i$  must pass through  $\hat{\mathbf{p}}_i$ . The position of  $\hat{\mathbf{p}}_i$  can be obtained by calculating the intersection of these two lines, as in Fig. 2. Since two lines hardly ever intersect in space, the least squares method is used here to calculate the intersection point.

We then obtain the radius  $R$  of the eye by calculating the mean value of the distance between  $\hat{\mathbf{p}}_i$  and  $\mathbf{c}$ .

$$R = \frac{1}{M} \sum_{i=1}^M (\hat{\mathbf{p}}_i - \mathbf{c}) \quad (11)$$



**Fig. 2.** We find  $\hat{p}_i$  by intersecting gaze line (blue line) from eyeball center with orange line which is passing through  $p_i$  and the camera center  $o$ . (Color figure online)

### 2.3 Calculating Gaze Direction

In the assumptions of Sect. 2.1 each pupil center is on the surface of the eye and its projection is  $\hat{p}_i$ . Due to the ambiguity of the distance, the  $p_i$  obtained by unprojection calculation can hardly lie exactly on the surface of the eye. However, in the normal case, the line passing through the center of the camera and  $p_i$  is intersected by the eye. Therefore, A new pupil center  $p'_i$  can be determined by calculating the intersection of the line passing through the center of the camera and  $p_i$  with the eye ( $c, R$ ). In order to calculate the position of the intersection point, the magnitudes of  $d_1$  and  $L$  are first calculated.

$$\begin{aligned}
 d_1^2 &= R^2 - d_2^2 \\
 &= R^2 - (\|c - o\|^2 - L^2) \\
 &= R^2 + L^2 - \|c - o\|^2
 \end{aligned}
 \tag{12}$$

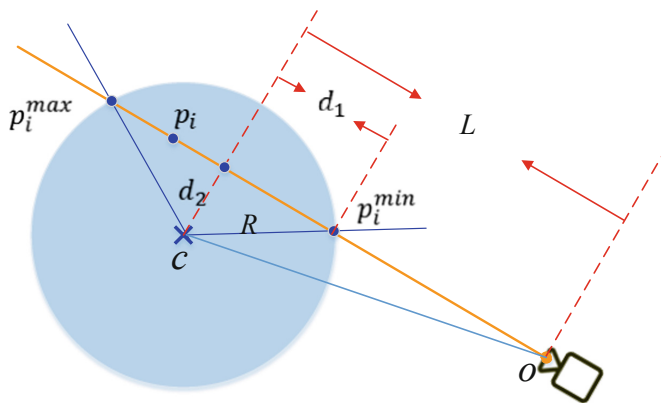
$$L = (c - o) \cdot \frac{(p_i - o)}{\|p_i - o\|}
 \tag{13}$$

As can be seen by Fig. 3, normally, the line  $o - p_i$  will have two intersections with the eye ( $c, R$ ), and the closest intersection is chosen here.

$$d_{\min} = L - d_1
 \tag{14}$$

$$p'_i = d_{\min} \cdot \frac{(p_i - o)}{\|p_i - o\|}
 \tag{15}$$

After obtaining the new pupil center position,  $n_i$  is discarded in favor of using  $n'_i = p'_i - c$  as gaze direction.



**Fig. 3.** We use the average radius of the eye  $R$  to recalculate the spatial position of the pupil  $p_i^{min}$ .

### 3 System Design and Implementation

In any head posture, with the eye looking at different positions, the pupil center will be presented in a different position in the image. Therefore, it is often assumed that the movement of the pupil center can be used as a feature of the change in vision. The most common PCCR method uses the vector between the pupil center and the reflected light spot on the cornea to represent the feature of gaze direction. Assuming that the head-mounted device remains fixed relative to the head, the position of light source’s reflected spot on the cornea is fixed. The spot does not change with eye movement, so the PCCR vector only changes with the movement of the pupil center. Then the pupil corneal vector  $(x, y)$  in the eye camera image is mapped to the pixel point  $(X, Y)$  on the scene image or screen by the interpolation formula.

$$X = \sum_{k=0}^{n-1} a_k x^i y^j, i \in [0, k], j \in [0, k] \tag{16}$$

This method is more accurate when looking at the calibration points, but less accurate when looking at the non-calibrated points. The reason for this phenomenon is that annotating the calibration points corresponds to bringing the interpolated nodes into the interpolation formula, while annotating the non-calibrated points corresponds to bringing the non-interpolated nodes into the interpolation formula. A deeper reason is that the polynomial chosen does not fit well the correspondence between the gaze points and the PCCR, or that there is no corresponding polynomial relationship between the two. The usual solution to this problem is to increase the order of the polynomial to improve the accuracy of the fit, but this results in more parameters and increases the complexity of the calibration procedure. Moreover, when the order is high enough, the Runge Phenomenon may occur. To address this problem this paper proposes

a solution to improve the accuracy of non-calibration points without increasing the polynomial parameters and order. We argue that the PCCR in the ocular image does not make full use of the information about the change in the gaze direction. Therefore, in this paper, we propose to use the angle of the gaze direction  $(\alpha, \beta)$  instead of the PCCR  $(x, y)$  as the feature of gaze direction. In the previous section we have obtained the gaze direction vector  $\mathbf{n}'_i$  and here it is only necessary to transform vector into angle  $(\alpha, \beta)$ .

$$\mathbf{n}'_i = (x_{\text{gaze}}, y_{\text{gaze}}, z_{\text{gaze}}) \quad (17)$$

$$\begin{cases} \alpha = \arctan\left(\frac{|z_{\text{gaze}}|}{x_{\text{gaze}}}\right), x > 0 \\ \alpha = \pi - \arctan\left(\frac{|z_{\text{gaze}}|}{|x_{\text{gaze}}|}\right), x \leq 0 \end{cases} \quad (18)$$

$$\begin{cases} \beta = \arctan\left(\frac{|z_{\text{gaze}}|}{y_{\text{gaze}}}\right), y > 0 \\ \beta = \pi - \arctan\left(\frac{|z_{\text{gaze}}|}{|y_{\text{gaze}}|}\right), y \leq 0 \end{cases} \quad (19)$$

The mapping between the gaze direction angle  $(\alpha, \beta)$  and the scene image coordinates  $(X, Y)$  is then modelled by a polynomial. Then Eq. (20) can be rewritten as

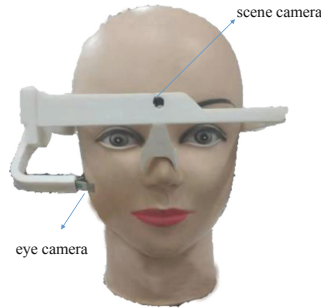
$$X = \sum_{k=0}^{n-1} a_k \alpha^i \beta^j, i \in [0, k], j \in [0, k] \quad (20)$$

A comparison of Eq. (20) with Eq. (16) shows that the system designed in this paper uses the same number of parameters as the traditional pupil-corneal vector method. In contrast to the PCCR method, the system is designed to use the angle of the gaze direction instead of PCCR vector, thus improving the use of the variation in visual information. A 3D to 2D mapping model is used to avoid advance calibration between the camera and the headset and to reduce the hardware architecture requirements.

## 4 Experiments

We use a head-mounted eye-tracking device made in our laboratory, in which the image resolution of the scene camera and the eye infrared camera are both  $640 \times 480$  pixels. The acquisition frame rate is 60 FPS. The development environment is Qt Creator 4.7 + OpenCV 3.0. In order to ensure the fairness of the experimental results, our experiments use the same head-mounted device to test the experimental results of the PCCR method and the method proposed in this paper (Fig. 4).





**Fig. 4.** The head-mounted eye tracker made by our laboratory with a scene camera and a eye camera.

#### 4.1 Calibration

In the experiment, we use a nine-point calibration for both our method and the PCCR method respectively. Then the subjects gazed at the calibrated and non-calibrated points and collected the distribution of their respective gazing points. The polynomial used in the calibration process was the second order polynomial proposed by Cerrolaza et al. [20]:

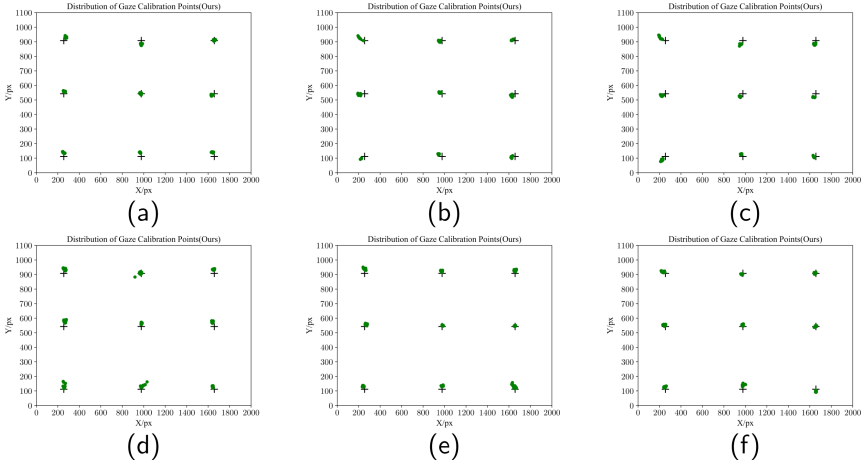
$$X = a_0 + a_1x + a_2x^2 + a_3y + a_4y^2 + a_5xy \quad (21)$$

$$Y = b_0 + b_1x + b_2x^2 + b_3y + b_4y^2 + b_5xy \quad (22)$$

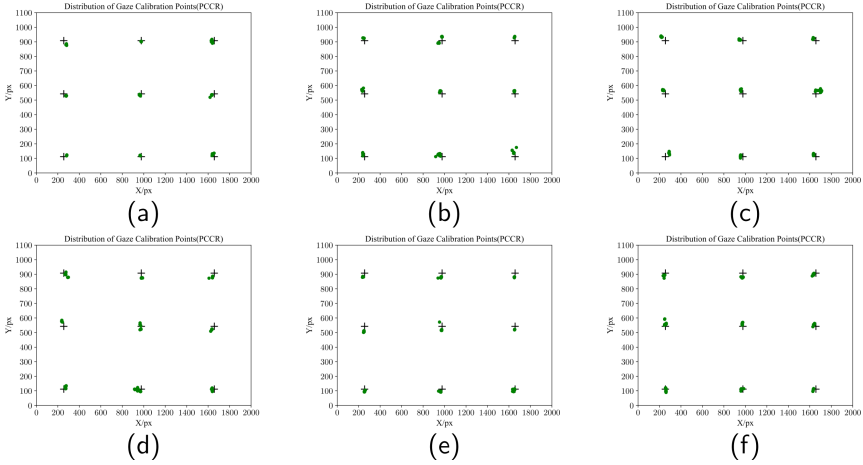
#### 4.2 Data Collection

We invited 6 subjects to participate in this experiment. Every subject sits at a position approximately 0.7 m in front of the screen and adjusts the head posture to ensure that all calibration points on the screen are present in the scene image. The head posture is kept fixed during the calibration. Firstly, the markers on the screen are looked at in turn and the angle of vision or pupil-corneal vector is recorded for each calibration point. Secondly, the subject looks at a set of dots on the calibration target after the calibration is completed, and for each dot we collect 20 consecutive frames of data as a result of the experiment. Finally, significant shifts caused by involuntary eye movements such as nystagmus are removed. The distribution of the results of our method and PCCR method on calibration points are shown in Fig. 5 and Fig. 6.

In order to further evaluate the accuracy of the proposed method and the PCCR method at non-calibrated points, 16 test points different from the calibrated points were experimentally fixed on the target for evaluation. The distribution of the results of our method and PCCR method on test points are shown in Fig. 7 and Fig. 8.



**Fig. 5.** Distribution of gaze calibration points (Ours).



**Fig. 6.** Distribution of gaze calibration points (PCCR).

The crosses in Fig. 5, 6, 7 and Fig. 8 represent the calibration points on the calibration target and the cluster of points represent the real gaze points collected during the experiment. Once the data for the gaze points have been collected, Eq. (23) is used to calculate the angular error.

$$\bar{\alpha}_l = \frac{\sum_{j=0}^N \arctan \left( \sqrt{(x_{ij} - X_i)^2 + (y_{ij} - Y_i)^2} / L \right)}{N} \quad (23)$$

where  $N$  is the number of qualified samples,  $(x_{ij}, y_{ij})$  is the position of the  $j$ -th data corresponding to the  $i$ -th gaze point collected, and  $(X_i, Y_i)$  is the position

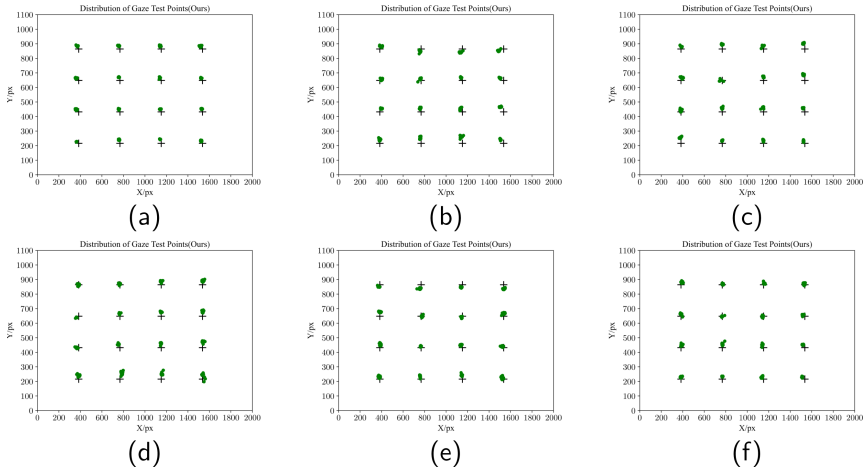


Fig. 7. Distribution of gaze test points (Ours).

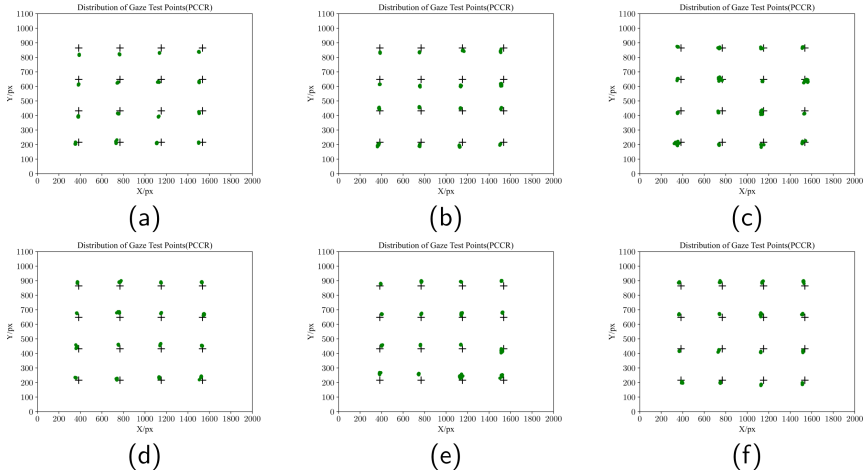
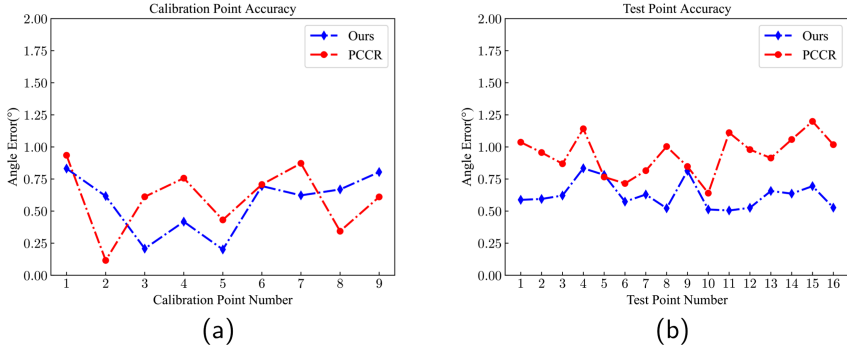


Fig. 8. Distribution of gaze test points (PCCR).

of the  $i$ -th reference gaze point. Figure 9 give the angular errors for each point when observing both calibrated and non-calibrated points for both our methods and the PCCR method.

According to the experimental data in Fig. 9, the errors of our method and the PCCR method are  $0.56^\circ$  and  $0.60^\circ$  respectively for the annotated calibration points, and  $0.63^\circ$  and  $0.94^\circ$  respectively for the annotated non-calibrated points. The experimental results show that the errors of the two methods are close to each other at the calibration points, and the accuracy of our method is improved at the non-calibrated points. Figure 9 show that using gaze direction



**Fig. 9.** Error of calibration point and test point results. (a) is error of calibration point results; (b) is error of test point results.

angles instead of PCCR vectors as features makes better use of the information on gaze direction variation and improves the accuracy of the system in general and at non-calibrated points in particular. Using only a single camera and no eye-averaging parameters, the accuracy of this method remains at the same level compared to other 3D methods (Table 1).

**Table 1.** Comparison of the results of different 3D methods.

Method	RMSE(°)	Hardware configuration
Shih Sheng-Wen [12]	0.8	2 cameras, 1 light sources
Zhu Z [14]	1.6	2 cameras, 1 light sources
Liu [21]	3.7	2 cameras, 2 light sources
Wen Q [22]	3.45	1 cameras
Tobii Pro Glasses 3	0.5	4 cameras, 16 light sources
Diskablis Glasses 3	0.3	2 cameras, 2 light sources
Ours	0.63	1 camera

## 5 Conclusion

Based on the features of the pupil’s motion trajectory, we propose a single-camera head-mounted 3D eye tracking system. The number of cameras is reduced by analyzing the pupil motion trajectory to obtain the 3D gaze direction. By using the mapping model from the gaze direction to the scene, the advance calibration of the hardware structure is avoided. Moreover, the results show that the method in this paper has better accuracy at non-calibrated points compared to the PCCR method when using the same hardware equipment. The complexity of the hardware structure is greatly reduced while ensuring accuracy.

**Acknowledgment.** This work was supported by the National Natural Science Foundation of China under Grants nos. 61871326, and Ningbo Natural Science Foundation under Grants nos. 202003N4367.

## References

1. van Reijmersdal, E.A., Rozendaal, E., Hudders, L., Vanwesenbeeck, I., Cauberghe, V., van Berlo, Z.M.: Effects of disclosing influencer marketing in videos: an eye tracking study among children in early adolescence. *J. Interact. Market.* **49**, 94–106 (2020)
2. Steindorf, L., Rummel, J.: Do your eyes give you away? A validation study of eye-movement measures used as indicators for mindless reading. *Behav. Res. Methods* **52**(1), 162–176 (2020)
3. Bergman, M.A., et al.: Is a negative attentional bias in individuals with autism spectrum disorder explained by comorbid depression? An eye-tracking study. *J. Autism Dev. Disord.* 1–14 (2021). <https://doi.org/10.1007/s10803-021-04880-6>
4. Bozomitu, R.G., Pășărică, A., Tărniceriu, D., Rotariu, C.: Development of an eye tracking-based human-computer interface for real-time applications. *Sensors* **19**(16), 3630 (2019)
5. Rahal, R.M., Fiedler, S.: Understanding cognitive and affective mechanisms in social psychology through eye-tracking. *J. Exp. Soc. Psychol.* **85**, 103842 (2019)
6. Matthews, S., Uribe-Quevedo, A., Theodorou, A.: Rendering optimizations for virtual reality using eye-tracking. In: 2020 22nd Symposium on Virtual and Augmented Reality (SVR), pp. 398–405. IEEE (2020)
7. Pohl, D., Zhang, X., Bulling, A.: Combining eye tracking with optimizations for lens astigmatism in modern wide-angle HMDs. In: 2016 IEEE Virtual Reality (VR), pp. 269–270. IEEE (2016)
8. Mikhailenko, M., Kurushkin, M.: Eye-tracking in immersive virtual reality for education: a review of the current progress and applications (2021)
9. Takemura, K., Takahashi, K., Takamatsu, J., Ogasawara, T.: Estimating 3-D point-of-regard in a real environment using a head-mounted eye-tracking system. *IEEE Trans. Hum. Mach. Syst.* **44**(4), 531–536 (2014)
10. Morimoto, C.H., Mimica, M.R.: Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.* **98**(1), 4–24 (2005)
11. Arar, N.M., Gao, H., Thiran, J.P.: Towards convenient calibration for cross-ratio based gaze estimation. In: 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 642–648. IEEE (2015)
12. Shih, S.W., Liu, J.: A novel approach to 3-D gaze tracking using stereo cameras. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **34**(1), 234–245 (2004)
13. Urano, R., Suzuki, R., Sasaki, T.: Eye gaze estimation based on ellipse fitting and three-dimensional model of eye for “intelligent poster”. In: 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 1157–1162. IEEE (2014)
14. Zhu, Z., Ji, Q.: Novel eye gaze tracking techniques under natural head movement. *IEEE Trans. Biomed. Eng.* **54**(12), 2246–2260 (2007)
15. Swirski, L., Dodgson, N.: A fully-automatic, temporal approach to single camera, glint-free 3D eye model fitting. In: Proceedings of PETMEI, pp. 1–11 (2013)

16. Li, Z., Miao, D., Liang, H., Zhang, H., Liu, J., He, Z.: Efficient and accurate iris detection and segmentation based on multi-scale optimized mask R-CNN. In: Zhao, Y., Barnes, N., Chen, B., Westermann, R., Kong, X., Lin, C. (eds.) ICIG 2019. LNCS, vol. 11902, pp. 715–726. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-34110-7\\_60](https://doi.org/10.1007/978-3-030-34110-7_60)
17. Świrski, L., Bulling, A., Dodgson, N.: Robust real-time pupil tracking in highly off-axis images. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 173–176 (2012)
18. Santini, T., Fuhl, W., Kasneci, E.: Pure: robust pupil detection for real-time pervasive eye tracking. *Comput. Vis. Image Understand.* **170**, 40–50 (2018)
19. Safaee-Rad, R., Tchoukanov, I., Smith, K.C., Benhabib, B.: Three-dimensional location estimation of circular features for machine vision. *IEEE Trans. Robot. Autom.* **8**(5), 624–640 (1992)
20. Cerrolaza, J.J., Villanueva, A., Villanueva, M., Cabeza, R.: Error characterization and compensation in eye tracking systems. In: Proceedings of the Symposium on Eye Tracking Research and Applications, pp. 205–208 (2012)
21. Liu, M., Li, Y., Liu, H.: 3D gaze estimation for head-mounted eye tracking system with auto-calibration method. *IEEE Access* **8**, 104207–104215 (2020)
22. Wen, Q., Bradley, D., Beeler, T., Park, S., Xu, F.: Accurate real-time 3D gaze tracking using a lightweight eyeball calibration. *Comput. Graph. Forum* **39**(2), 475–485 (2020)