IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 18, 2020, accepted June 1, 2020, date of publication June 4, 2020, date of current version June 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000063

Pupil Detection Based on Oblique Projection Using a Binocular Camera

JUNJIE ZHANG, GUANGMIN SUN, KUN ZHENG, AND SARAH MAZHAR
Electrical Engineering Department, Beijing University of Technology, Beijing 100124, China

Corresponding author: Kun Zheng (zhengkun@bjut.edu.cn)

This work was supported in part by the Beijing Education Science Planning, China, under Project CADA18069.

ABSTRACT The real visual attention areas of students during the learning process are vital data. They represent the students' visual concentration and provide strong support for the analysis of learning effects. It is necessary to use professional equipment to collect the visual attention areas of students on a screen but this equipment is expensive, so it cannot be solely depended upon. A binocular camera can directly obtain the depth information under static conditions; thus, detected pupil and iris sizes do not change as the user-to-screen distance changes. Robust pupil detection is an important prerequisite for gaze detection in a real-world setting. Hence, an accurate algorithm to calculate the position of pupils and the size of irises under natural light is proposed in this paper. It is based on the positive and negative oblique projection of relative linear density. The MPIIGaze dataset was used to test the algorithm, and the experimental results show that the algorithm is resistant to changes in illumination and the presence of glasses. Moreover, compared with a system based on a monocular camera, the error in the distance between label points and gaze location is decreased by 5 pixels and dispersion is decreased by 3 pixels. Finally, the line of sight is concentrated and the dispersion is low. With the 'user-screen' distance between 60cm and 80cm, the accuracy can reach up to 1.5° to 2.2°.

INDEX TERMS Binocular camera, positive oblique projection, negative oblique projection, relative linear density, eye vector.

I. INTRODUCTION

The line of sight of an observer can be reflected by the movement of the pupil in the eye. Moreover, statistics on gaze points can be utilized to derive a user's visual attention area. Education and learning data analysis not only can help students with personalized learning but also support teacher personalization [1]. The true visual attention area can reflect a student's interest during the learning process, which can reveal a student's concentration and form a strong basis for learning effect analysis [2]. Furthermore, the change in pupil size reflects fluctuations in emotions and mental state during the learning process. The use of pupil and gaze detection technology in the classroom enables class content and guidance strategies to be adjusted according to an analysis of these data, which can improve students' understanding and acquisition of new knowledge [3]. In the process of student learning, it is helpful to design a personalized learning style that can improve the ability of the student to grasp new knowledge. Through personalized learning, the same knowledge is presented in different ways, and then different attention values are assigned. Researchers [4], [5] found that personalized learning methods are more attractive to students and can improve learning outcomes. For best gaze tracking the pupil should be locate accuracy. Reflections on glasses, illumination changes, eyelashes covering the pupil may effect on pupil locate, especially in real-world scenarios. Several algorithms address pupil detection under laboratory and real environments. For example, Jan make use of Hough transform algorithm with near-infrared or visible wavelength illumination for pupil detection under laboratory environments [6]. Other proposals such as [7]–[10] use ellipse fitting, regression and integral projection method for pupil detection under laboratory conditions respectively. ExCuSe [11] is proposed in 2016 that is well suited for real-world eye-tracking applications. ExCuSe is an algorithm that is based on edge detection and morphologic operations. Canny edge detector was used to detect the edges of the image and then

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Napoletano.

several morphological operations was applied to clean noise and erase straight lines. For all remaining curved lines, it calculates their enclosed mean intensity, selecting the curve with the lowest value as the pupil. Finally, an Ellipse is fitted to this selected curve [12]. With the development of deep learning there have been several improvements in the last year. DeepEye was proposed in 2019. It is able to handle varying illumination, blurring and reflections problems under real environment. DeepEye outperforms previous eye-tracking methods tested with data sets and it improves the results of the current state of the art in a 26% in 2019 [12]. After obtaining the precise position of a pupil, an eye movement vector is constructed to represent the change in gaze. Arar *et al.* [13] presented a comprehensive analysis of regression-based user calibration techniques, and proposed a novel weighted least squares regression-based user calibration method together with a real-time cross-ratio based gaze estimation framework. However, LEDs in four corners were used as an auxiliary light source. Cheung and Peng [14] reduce gaze error by combining the inner corner point of the eye with the center point of the eye in an image to form a feature vector. A polynomial method is used for the mapping function. However, the inner eye point is affected by illumination and the head can only move within a small range. General regression neural networks have been used [15] to map label points with gaze points. However, pupil center and outer eye points are used to create the eye vector, and the accuracy of the method is influenced by this input vector. Convolutional neural networks have been used for gaze detection by Li *et al.* [16]. This approach could simplify the processing of feature extraction, but a large amount of training data must be collected to enable a convolutional neural network to be used. Shih *et al.* [17] used a multi-camera system and external light source to detect the line of sight. The accuracy of this approach is influenced by changes in illumination. Moreover, the equipment is costly and only suitable for detecting the line of sight in a specific environment, which is not conducive to widespread use. To detect gaze points without fixing the head, the depth images of the Kinect can be used [18]. Choi *et al.* [19] used a Kinect to obtain eye regions and head posture. The ASM [20] and KLT [21] algorithms were used to extract eye appearances for the eye vector and Gaussian regression was used to map the eye vectors and screen points.

In all related studies, eye trackers, monocular cameras, and binocular cameras were used to obtain gaze points. For various such devices, Table 1 lists the details of price, usable range, precision, invasiveness (intrusive and non-intrusive), depth, and imaging type. As binocular camera has some functions of stereo camera, a simple hardware structure, low price and more scalability, binocular camera with fixed baseline is used. Different from stereo camera for 3D reconstruction, binocular camera is used for measuring 'user-screen' distance only as per requirement of proposed work. Although the use of commercial products is common, but accuracy of the same level as of commercial products can be achieved using binocular camera only. Binocular camera can also provide support for data collection in E-learning and is cheap and convenient as well. Commercial products have high accuracy but are not affordable. Furthermore, cheap products cannot measure pupil size as well.

TABLE 1. Comparison of different gaze-point detection devices.

Device	Eye tracker	Wearable device	Monocular camera	Binocular camera
Price	High	High	Low	Low
Usage range	Small	Small	Big	Big
Precision	High	High	Low	Low
Invasive	Weak	String	Weak	Weak
Depth	Yes	No	No	Yes
Picture type	Infrared	Infrared/ RGB	RGB	RGB

The current problems with gaze detection include the following: 1) professional equipment from manufacturers such as Tobii must be used to detect eye regions for gaze detection, 2) detection is sensitive to changes in illumination if an ordinary camera is used to detect gaze points and an auxiliary light source is needed if there is insufficient illumination, 3) complex eye position algorithms are needed, and 4) accuracy decreases if glasses are worn.

To improve the above deficiencies, the positive and negative oblique projection based on relative linear density or PNOP for short that is proposed in this study. This method uses a monocular camera to locate the pupil position. The algorithm 1) detects pupils accurately, 2) calculates pupil and iris size according to the relationship between positive and negative oblique projection peaks, 3) is robust to ambient light changes and interference from glasses, and 4) detects irises even if there are slight head movements.

II. METHOD

The algorithm consists of Three parts. First image processing. Second, the candidate pupil regions are detected using multiple positive and negative oblique projections. Third, according to the relative linear densities of each projection line, the lines with the minimum values are selected, and the intersection of the positive and negative oblique projection lines determines the pupil. The algorithm can locate pupil positions accurately; meanwhile, iris size can be calculated even if the user-to-screen distance changes. The algorithm is robust to changes in illumination and the interference caused by glasses, in contrast to other algorithms. Furthermore, the algorithm has low computational complexity. The process of pupil location is shown in Figure 1.

A. PREPROCESSING

The color is similar between pupil(black) and iris(brown) when images captured by ordinary camera. Thus, it is important to distinguish pupil. First, the binocular camera is calibrated by Zhang's calibration method [22] to eliminate the

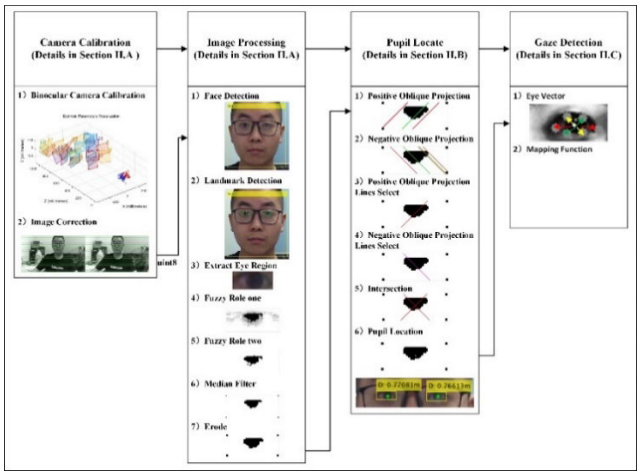


FIGURE 1. Pupil location process.

image distortion, and then the binocular correction is performed on the acquired image. After the binocular camera is calibrated, face detection is performed using the histogram of oriented gradients (HoG) features of the corrected image.

Finally, the 300-W facial landmarks dataset [23]–[25] was used to train the model to recognize 12 feature points around the eyes, and six feature points are extracted from the images of each eye. The six feature points are the outer corner of the eye, the inner corner of the eye, the left and right points on the upper eyelid, and the left and right points on the lower eyelid, as shown in Figure 1. The eye region is extracted according to the position of landmarks. As we can see in the Image processing step (3), there are four parts in the eye region i.e. skin, sclera, iris and pupil. It is easy to distinguish skin and sclera, whereas pupil (black) and iris (brown) are difficult to distinguish because of similarity in color. In order to locate pupil accurately, we should remove other parts first. As black and white color are easy to distinguish, we set low pixel value to be lower and high pixel value to be higher. Thus, two fuzzy algorithms are used. The general idea is that we increase the contrast. The mean intensity and variance are calculated, we separate the pixel of image to three part, less than mean subtraction variance, above mean add variance and between mean subtraction variance and mean add variance.

$$Image\,(x,y) = \begin{cases} \dfrac{Image\,(x,y)}{A}, Image\,(x,y) \le \text{mean-std} \\ B \times [Image\,(x,y) - C] + D, \\ mean - std < Image\,(x,y) \le mean + std \\ E \times [Image\,(x,y) - F] + G, Image > mean + std \end{cases} \tag{1}$$

where A to G is constant. In the paper, A to G represents 10,10/7,5,50,105/180,75 and 150 respectively. Pupil and iris area can be extracted as seen in Figure 1(Image Processing (4)). As pupil and iris are similar in color, in order to extract pupil area, the second fuzzy algorithm is used.

$$Image\,(x,y) = \begin{cases} H, Image\,(x,y) > th \\ I, Image\,(x,y) \le th \end{cases} \tag{2}$$

where H, I and th is constant. In the paper, H equal 255, I equal 0 and th equal 15 respectively. Pupil area can be extracted as seen in Figure 1(Image Processing (5)). Small area is removed in the last step of image processing using median filter and erode. The result can be seen in Figure 1(Image Processing (6) and (7)).

B. PUPIL LOCATION

Iris size is influenced by the user-to-screen distance when a monocular camera is used. Different distances will cause the size of the iris in the image to change. The region of the iris in the image contains the pupil; thus, it is essential to detect the iris region. To calculate an accurate iris size, the relationship between the user-to-screen distance and the area of eye region is established by regression. A monocular camera cannot obtain depth information under static conditions [26]. Hence, the HNY-CV-001 binocular camera is used to calculate the user-to-screen distance. The baseline of HNY-CV-001 is 6 cm and its focal length is 3.6 mm. Assuming the distance between the iris and camera is D, the disparity map is calculated using a semi-global block matching algorithm [27]. The depth can be obtained from the disparity map, which is shown in Figure 2.

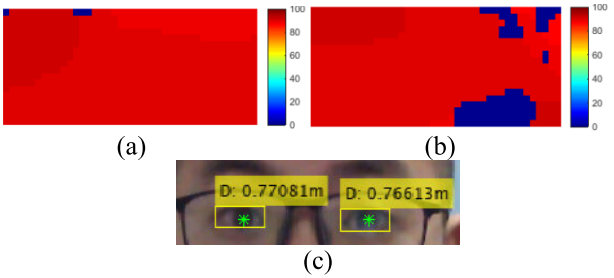


FIGURE 2. Disparity map for the (a) left eye and (b) right eye. (c) User-to-screen distance for each eye.

Studies have shown that constriction and dilation of the pupils reflect emotion change. The students cannot sit in same posture for a long time during electronic learning on the internet. As 'user-screen' distance changes, the pupil size in image will also change even if the actual size of the pupil has not changed. Expression (3) is used to eliminate the impact caused by the distance change as much as possible. Before experience, the user sits in front of the camera and changes position within the effective measurement distance of the binocular camera, at that time, Y_{reala} is measured and then we fit the relationship between the pupil area and distance, as shown in Figure 3. During experiment, binocular camera can measure 'user-screen' distance as D, Y_{reala}(D) is used to calculate pupil size as Y, and $Y_{calculate}$ can be calculated using

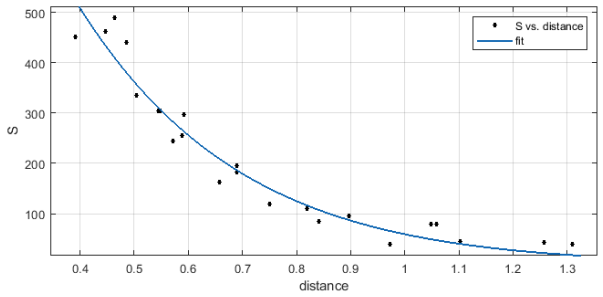


FIGURE 3. Relationship between user-to-screen distance and eye area.

the proposed method.

$$Y_{real} = Y_{reala} + Y - Y_{calculate} \tag{3}$$

The result shows that $Y \propto 1/D$. Thus, $\pi r^2 = P_1/(D + q_1)$, where P_1 and q_1 are constants. It is obvious that the grayscale values change around the human eye and, considering that the human eye is a symmetrical structure, the eyebrows and the upper and lower eyelids are symmetrically distributed around it. Hence, an oblique projection method can effectively remove the interference of the eyebrows and eyelids, and thus the positive and negative oblique projection algorithm is proposed in this study. This algorithm can locate the pupil accurately. An example of positive and negative oblique projection is shown in Figure 4.

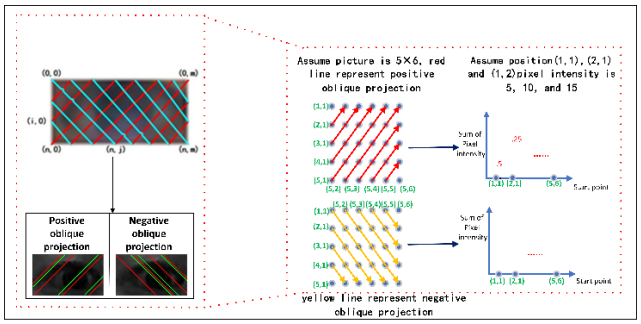


FIGURE 4. Example of positive and negative oblique projection.

The red lines in Figure 4 show positive oblique projection and the cyan lines indicate the negative oblique projection. The relative size of the pixels will change according to the user-to-screen distance; thus, the following two equations are used to separately sum up the line values for the positive and negative oblique projection. Equations (4) is used to calculate Positive oblique projection and Equations (5) is used to calculate Negative oblique projection.

$$S(x, y) = \sum_{x=x-1, y=y+1}^{x\geq 1, y\leq col} data\,(Mx, My) \tag{4}$$

$$S(x, y) = \sum_{x=x+1, y=y+1}^{y\leq col, x\leq row} data(Mx, My) \tag{5}$$

$$M = \sqrt{\frac{P_1}{\pi} \times \frac{1}{S + q_1}} \tag{6}$$

Here, *data* (*Mx*, *My*) is the grayscale value of location (*x*,*y*) and *M* is the ratio between the distance and pixel size. Further, *row* and *col* are the numbers of rows and columns of the image, respectively. Images under different situations (i.e., people with glasses, people without glasses, and people under different illuminations) were selected from the FDDB [31] and BioID [32] face datasets. The regression tree algorithm is used to detect feature points around the eyes. According to the relationship between the inner and outer eye corners, the right and left eye regions are segmented separately, examples of segmented eye regions are shown in Figure 5.

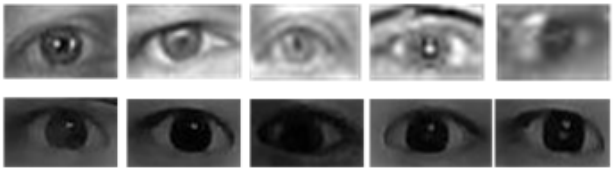


FIGURE 5. Example of segmented eye regions.

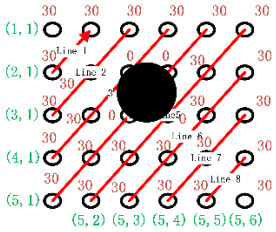


FIGURE 6. An explanation of RLD.

The results of the PNOP algorithm to segment eye regions are shown in Figure 6. The results of the positive oblique projection are shown in Figure 6(a) and the results of negative oblique projection are shown in Figure 6(b). The result indicates that positive and negative oblique projections can roughly locate pupil position. However, there are multiple lines in the image; thus, pupil location cannot be obtained precisely. To increase pupil location accuracy, RLD is proposed to remove extra oblique projection lines. RLD is in units of pixels and is calculated as follows:

$$RLD = \frac{S}{\sqrt{(x_{end} - x_{start})^2 + (y_{end} - y_{start})^2}} \tag{7}$$

where S is the sum of the positive and negative oblique projections, (xend, yend) is the end position of the image, and (xstart, ystart) is the start position of the image. A more intuitive explanation can be seen in Figure 6. We assume the intensity of line 1 to line 4 is 60, 90, 90 and 90 respectively. RLD of line 1 to line 4 is 30, 30, 22.5 and 18 respectively. Thus, line 4 is selected.

The pixel value of the pupil is lowest in the eye region; thus, the sum of oblique projection is lowest here. Because the eye region is a rectangle, difference start points may lead

to different Euclidean distances between the start and end points; thus, the RLD algorithm can be used to locate pupil position efficiently. The line with the minimal RLD value may cross the pupil position. When a light illuminates the face, the uneven distribution of light will affect the accuracy of the detection. The eye region is only a small part of the entire face; thus, the eye region is extracted. When a change in illumination occurs, the eye region is also affected. Assuming the light intensity is C, the oblique projections are calculated as follows.

$$S = \sum_{x=x-1,y=y+1}^{x\geq 1,y\leq col} data\,(Mx, My) + C \tag{8}$$

$$S = \sum_{x=x+1,y=y+1}^{y\leq col,x\leq row} data\,(Mx, My) + C \tag{9}$$

The influence of illumination is linear, i.e., $S = F + C$. Thus, the proposed algorithm is robust to changes in illumination. Glasses all have the same refractive index, so the influence of glasses is linear as well. Assuming the influence of glasses is G, $S = F + G$. Thus, the proposed algorithm is robust with respect to the presence of glasses. In the next section, we prove these two assumptions. The results of the positive and negative oblique projections based on *RLD* are shown in Figures 7(c) and 7(d). After the projection results have been filtered by the *RLD*, the positive and negative oblique projection results will have only one intersection point in the image. This intersection point is the position of the pupil, as shown in Figures 7(e) and 7(f). An analysis of the positive and negative oblique projection curves (as shown in Figures 7(a) and 7(b), respectively) reveals that the number of projection lines is odd regardless of the number of positive oblique projection lines or negative oblique projection lines. Eye region consists of sclera(white), iris(brown) and pupil (black). As shown in Figure 7(h), there is a boundary between the sclera and the iris (L1 and L5) and there is also a boundary between the iris and the pupil (L2 and L4). The pixel values on each projection line are accumulated and the peaks and troughs will appear due to the color jump. Eye structure is symmetrical. Thus, each side of the pupil has the same projection lines and an extra projection line through the pupil. So, the number of projection lines is odd. The result in Figure 7(a) shows that the numbers of positive projection lines are 3, 5, 7, and 7, respectively. They respectively contain 2, 3, 4, and 4 peak points as well as 1, 2, 3, and 3 wave points. The result in Figure 7(b) shows that the numbers of negative projection lines are 3, 5, 7, and 3. They respectively contain 2, 3, 4, and 2 peak points as well as 1, 2, 3, and 1 wave points. The positive and negative oblique projection results are consistent with the symmetrical structure of the eye; that is, there is a difference in grayscale values between the sclera (white) and the iris (gray) regions. Moreover, there is a difference in grayscale value between the iris and the pupil (black). Because the eye is a symmetrical structure, the difference exists on both sides of the eye, so the number of projection lines is an odd number. In contrast to other algorithms, the proposed algorithm is interpretable. Therefore, in the projection results, the area between the first peak and the last

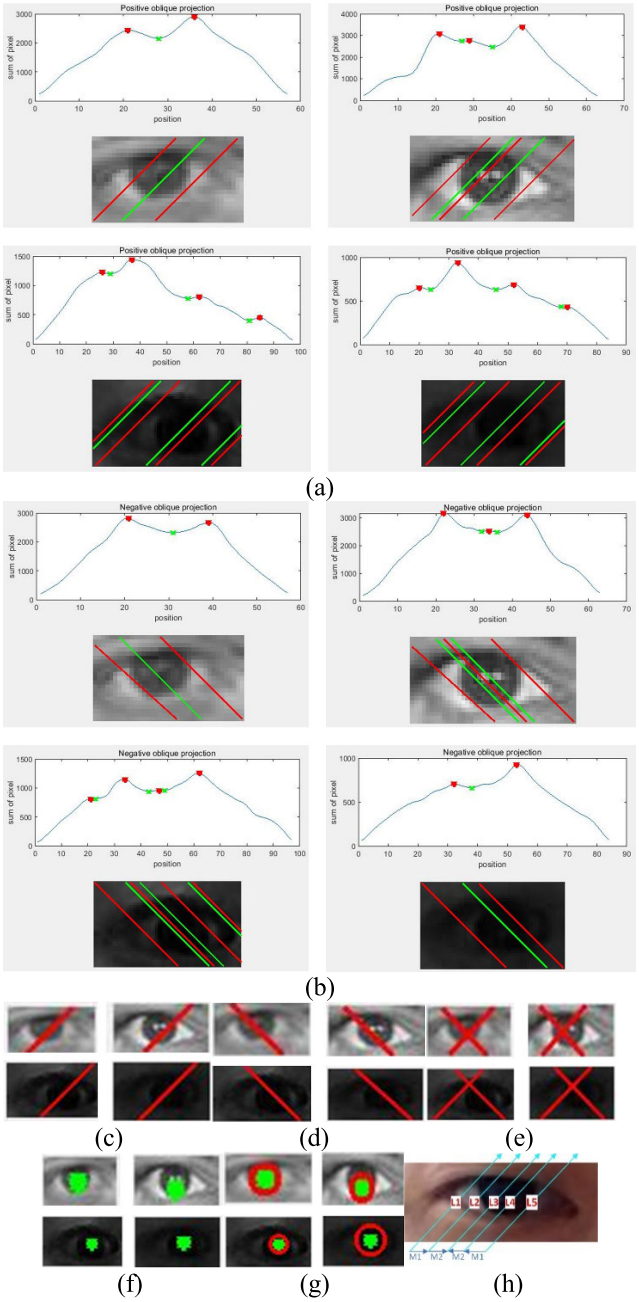


FIGURE 7. Results of iris location: (a) positive oblique projection, (b) negative oblique projection, (c) selected result of positive oblique projection, (d) selected result of negative oblique projection, (e) intersection of the lines, (f) location of the pupil, (g) iris and pupil sizes and (h) an explanation of projection lines are odd.

peak is the iris area, and the area between the adjacent peaks changes in color, that is, it is the border between the iris and pupil. Because the projection lines are parallel to each other and the eyes are approximately circular, the distance between the projection lines is calculated by the following equation and the result is the size of the iris in this image.

$$R = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}} \tag{10}$$

where (x_0, y_0) is the start point of the projection line and A, B, and C are the coefficients of Ax+ By + C = 0. The results of the iris size are shown in Figure 7(g). Glasses interfere with the precise positioning of the pupil, but the interference of the glasses can be excluded using the ratio of the peak to the valley in the positive and negative oblique projections.

C. CONSTRUCTION OF THE EYE VECTOR

It is necessary to obtain gaze detection in real time to estimate the region of interest. Therefore, the movement of the pupil is mapped with respect to the screen area to estimate the gaze point. The eye area only accounts for a small part of the whole face. It is difficult to detect eyes and extract eye vectors in the original image, so reducing the detection area is an effective way to improve accuracy here. In contrast to HoG features, Harr wavelets are good at detecting texture features, but they cannot detect direction features effectively [31]. In short, HoG features are suitable for describing shapes [32], and it is hence better to use them to describe a pedestrian or the shape of a target. After the face has been detected, a regression tree [33] is used to detect 12 feature points around the eyes.

A polynomial is used to map the eye vector and screen area. The elements of eye vector $V = [E_{inner}, E_{ul}, E_{ur}, E_{outer}, E_{lr}, E_{ll}]$ represent the distance from the inner eye point to the pupil, the distance from the left upper eyelid point to the pupil, the distance from the right upper eyelid point to the pupil, the distance from the outer eye point to the pupil, the distance from the right lower eyelid point to the pupil, and the distance from the left lower eyelid point to the pupil. The eye vector can be seen in Figure 1(Gaze Detection (1)). Thus, the precise position of the pupil is important for the construction of the eye movement vector. The changes in gaze direction may influence the distance from the pupil to other feature points. Therefore, eye vector V is used to represent the change of gaze direction and the following polynomial function is used to map the eye vector to screen point (S_x, S_y).

$$S_x = a_0 + a_1 V_{x_0} + a_2 V_{x_0}^2 + a_3 V_{y_0} \tag{11}$$

$$S_y = b_0 + b_1 V_{x_0}^2 + b_2 V_{x_0} + b_3 V_{x_0} V_{y_0} + b_4 V_{x_0}^2 V_{y_0} \tag{12}$$

III. RESULT

A binocular camera with a resolution of 1, 280 × 480 pixels was used to locate the pupil position without an additional light source. The system was implemented on a desktop PC (see Figure 8). The experimental hardware is listed in Table 2. The region of interest can be detected using the gaze detection.

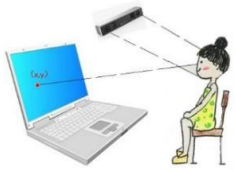


FIGURE 8. Gaze direction detection system with a 1280 × 480 pixel screen.

TABLE 2. Hardware environment.

HARDWARE	DELL G3 3779
CPU	Intel(R) Core (TM) i7-8750H @2.20GHz
GPU	Nvidia GeForce GTX 1060 with Max-Q
RAM	8GB (DDR4 2666MHz)
OS	Windows 10 64bit (DirectX 12)
MONITOR	CMN1738(17.2 inch) 1920*1080
CAMERA	HNY-CV-001 OV9714 1280*480@30fps
LANGUAGE	Python

The accuracy of pupil location is influenced by the user-to-screen distance. The optimal distance from the computer screen to the eyes is 60 cm [39], and a user cannot see the content on a screen clearly at distances of over 150 cm or less than 30 cm. Thus, the true user-to-screen distance is measured using an LDM-100 rangefinder and a binocular camera is used to calculate the user-to-screen distance using a disparity map. The deviation between these distances was then computed.

TABLE 3. Deviation between the true distance and calculated distance in the presence of glasses.

D_r(cm)	D_C(cm)	Error (%)	D_r	D_C	Error
37.7	31.12	17.5%	86.5	86.85	-0.4%
39.3	30.93	21.3%	98.2	98.47	-0.3%
41.9	35.73	14.7%	113.5	111.47	1.8%
47.0	42.19	10.2%	125.3	125.41	-0.09%
49.8	47.43	4.8%	128.4	128.04	0.3%
58.7	57.26	2.5%	137.8	137.05	0.5%
69.0	69.01	-0.01%	140.3	NA	NA
80.0	79.14	1.1%	156.2	NA	NA

As Table 3 shows, when the user-to-screen distance exceeds 140 cm, the image of the acquired eye region is too small and the eyes are obscured by lenses, so the pupil cannot be precisely located. When the user-to-screen distance is less than 50 cm, the errors between the measured and actual distances are large. Because the measurement range of the binocular camera is related to the binocular baseline and the focal length of the lens, when the distance is less than 50 cm, accurate measurement at close range requires the focal length of the camera and the baseline distance to be reduced. In our experiments, when the user was less than 50 cm away from the computer screen for a long time, this strained the eyes and hindered reading. Thus, the binocular camera used in the experiment can meet the requirements in terms of accuracy. If the user does not wear glasses, see Table 4, the measured distance of the pupil is increased because the lenses do not occlude the eyes. When the distance is greater than 150 cm, the user is too far away from the computer screen and cannot see the contents of the computer clearly, so the measurement of long distances is not important. The deviation of real distance and calculate distance with or without glasses can be seen in Figure 9. Hence, the method used in this paper meets the practical requirements of the system.

TABLE 4. Deviation between true distance and calculated distance without glasses.

D_r(cm)	D_C(cm)	Error (%)	D_r	D_C	Error
32.8	29.78	9.2%	93.8	95.11	-1.4%
38.7	33.17	14.3%	114.1	114.79	-0.6%
44.2	40.57	8.2%	120.7	117.60	2.6%
50.1	47.74	4.7%	132.9	132.05	0.6%
62.2	60.16	3.3%	138.1	138.18	-0.06%
71.9	71.22	0.9%	148.4	148.52	-0.08%
84.3	84.10	0.2%	152.8	150.60	1.4%
90.0	90.45	-0.5%	172.0	NA	NA

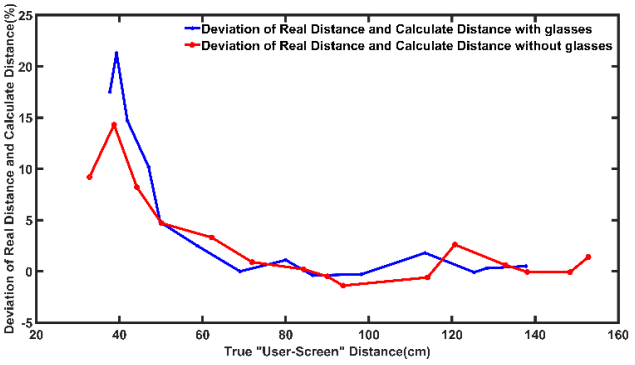


FIGURE 9. Deviation between the true and calculated distance with and without glasses.

Although pupil position is influenced by the presence of glasses, the PNOP algorithm can locate pupil position accurately from the ratio of the peaks to troughs in positive and negative oblique projection lines. Glasses frames influence the result of positive and negative oblique projection lines, which are selected by RLD. Figure 10 shows that the value of the frame is low, and the projection line is across the glasses frame, so the value of RLD is at a minimum and the sum of its value is less than 800. Therefore, the projection line based on the RLD is wrong. An analysis of the positive and negative oblique projection curves in Figure 10 reveals that the difference between the peak and valley points of the projection curve are less than 500, so when the minimum curve of the projection curve is less than a third of the maximum peak value of the projection, the candidate curve is the wrong result. The final eye region detection results are shown in Figure 10.

The BioID [32] dataset was used to test accuracy under changes in illumination and the presence of glasses. An example image as shown in Figure 11(a) and the result can be seen in Figure 11(b). Because of the small area of the eye, the illumination of the eye area is averaged for both flatlit or backlit conditions. When the PNOP algorithm is performed on the eye area, the intensity in the eye region is the same. When glasses are worn, the spectacle lens completely covers the eye area and the refractive index of the ophthalmic lens is uniform. Hence, the effect of the glasses on the projection result of the eye area is uniform. When students browse screen information, the difference in screen brightness and the brightness of the surrounding environment will affect the result of pupil location. To evaluate the robustness of the algorithm considering the different user-to-screen distances during the day and evening, the PNOP algorithm was used to test the location accuracy and iris size with and without glasses. The intensity was measured

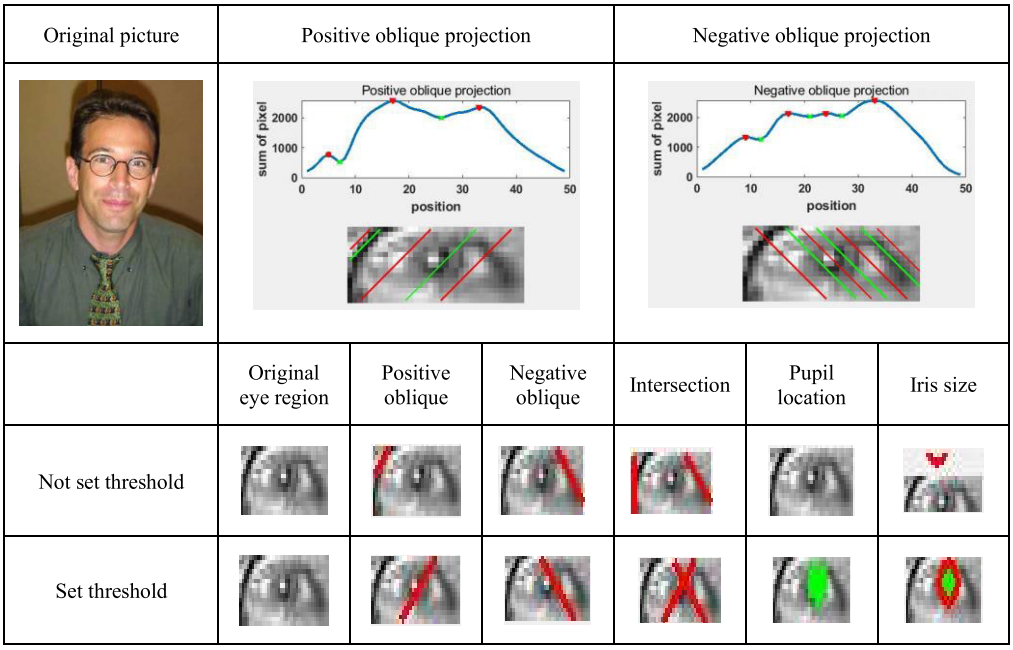
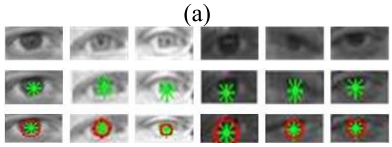


FIGURE 10. Results for the pupil position and iris size with the presence of glasses.

(a)



(b)

Indoor(LUX):140	Screen Luminance	100%				
Outdoor(LUX):60	Eye Luminance	148	166	180	179	179
Light(LUX):327	Distance	60.8	68.2	76.1	90.4	118.3
Without Glasses	Result					
With Glasses	Result					
Indoor(LUX):140	Screen Luminance	50%				
Outdoor(LUX):60	Eye Luminance	165	167	174	172	179
Light(LUX):327	Distance	55.4	68.4	79.4	93.4	108.5
Without Glasses	Result					
With Glasses	Result					
Indoor(LUX):140	Screen Luminance	0%				
Outdoor(LUX):60	Eye Luminance	162	161	179	179	178
Light(LUX):327	Distance	60.1	71.2	87.8	104.9	115.8
Without Glasses	Result					
With Glasses	Result					

(c)

Indoor(LUX):112	Screen Luminance	100%				
Outdoor(LUX):329	Eye Luminance	102	104	104	100	99
Light(LUX):0	Distance	54.3	62.6	71.4	80.7	94.4
Without Glasses	Result					
With Glasses	Result					
Indoor(LUX):112	Screen Luminance	50%				
Outdoor(LUX):329	Eye Luminance	86	86	90	90	92
Light(LUX):0	Distance	47.7	61.4	72.5	81.7	96.5
Without Glasses	Result					
With Glasses	Result					
Indoor(LUX):112	Screen Luminance	0%				
Outdoor(LUX):329	Eye Luminance	81	82	82	82	80
Light(LUX):0	Distance	49.5	64.0	72.1	83.1	92.8
Without Glasses	Result					
With Glasses	Result					

(d)

FIGURE 11. Results for glasses and changes in illumination: (a) example image, (b) result of iris location and pupil size, (c) results from evening measurements, and (d) results from daytime measurements.

using an SW-582 illuminometer and the result can be seen in Figures 11(c) and 11(d).

The illumination in the room, outside the room, and at the light source were measured using the SW-582 illuminometer in the evening, and the intensities were 140 lx, 60 lx, and 327 lx, respectively. The brightness of the computer screen was then adjusted to 0%, 50%, and 100% of the maximum brightness. The results show that the proposed method can locate pupil position accuracy and calculate the iris size in the experimental environment. The results of pupil location and iris size during the day are shown in Figure 11(d). We also measured the illumination in the room, outside the room, and at the light source. Because the ambient light intensity can meet the experimental requirements, there is no need to turn on the light source, and thus its brightness becomes zero. The brightness of the display was adjusted in the same way. The result shows that the PNOP algorithm is robust to changes in illumination around the eyes.

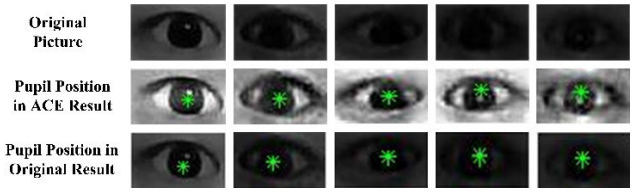


FIGURE 12. Results of pupil location in enhanced and original images.

It is necessary to enhance the image to improve the accuracy of pupil location using the current algorithm. To further evaluate the robustness of the algorithm to illumination changes, images captured under low light were evaluated, as shown in Figure 12. The results show that the brightness of the image decreases from left to right. In this experiment, the image was enhanced by the automatic color enhancement algorithm [38]. Pupil location was detected using the enhanced eye region image, and then the pupil position in the original eye region image was located. The results are shown in Figure 11, which show that pupil location in the enhanced eye region image is the same as that found in the original eye region image, which further proves that the algorithm is robust to brightness changes.

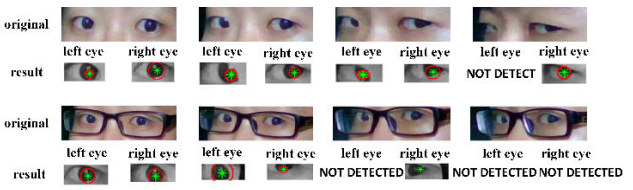


FIGURE 13. Detection results for head rotation.

It is difficult to locate the pupil position when the size of the eye region changes or is blocked by the face because of head movement. To evaluate the accuracy of the pupil position when the head rotates, images were captured with and without glasses. The results are shown in Figure 13. The proposed algorithm determines the position of the pupil by evaluating the magnitude of the positive and negative oblique projections on the RLD. Glasses with black frames hence affect the projection result. To evaluate the robustness of the algorithm, a user wore black-framed glasses to increase the difficulty of detection. The experimental results show that when a head without glasses is rotated from the front view

TABLE 5. Comparison of the accuracy of different algorithms on MPIIGaze.

Parameters			Algorithm									
			DeepEye		Hough		Excuse		VAH		Proposed (PNOP)	
LB	RB	H	LeftEye	RightEye	LeftEye	RightEye	LeftEye	RightEye	LeftEye	RightEye	LeftEye	RightEye
10	10	10	15.12	14.52	35.95	32.76	30.94	30.27	111.06	61.18	**5.89**	**6.50**
10	10	20	11.44	11.45	32.19	27.73	28.92	28.28	32.78	16.36	**7.26**	**6.18**
10	15	20	11.44	12.25	30.54	27.73	30.02	28.19	20.25	16.36	**6.83**	**6.18**
15	15	15	15.25	14.57	33.09	27.18	32.25	31.83	52.02	30.60	**6.48**	**7.00**
15	15	20	12.21	12.25	30.54	25.17	31.43	30.48	20.25	10.11	**6.86**	**6.07**
15	15	30	9.12	9.37	24.90	21.53	16.95	16.26	11.32	9.16	**8.52**	**7.46**
20	20	10	32.55	30.58	34.73	26.17	34.47	34.64	148.07	100.53	**13.54**	**17.87**
20	20	20	16.70	16.30	30.02	23.68	29.59	29.24	29.79	19.63	**7.04**	**7.80**
25	25	25	18.55	17.72	27.66	22.91	29.54	31.04	19.78	18.55	**7.96**	**9.28**
30	30	30	19.43	18.78	24.19	22.53	30.40	33.15	18.17	20.04	**9.36**	**11.72**

by more than 45°, the position of the contralateral eye pupil cannot be located due to the occlusion of one side of the face. When glasses are worn, a rotation of 30° from the front view means that the pupil cannot be accurately located because of occlusion from the face and glasses frame. When a user watches the computer screen, the rotation of the head is less than 20°. Otherwise, the reading of the content will be strongly affected. Therefore, the proposed method meets the needs of practical applications in normal reading situations.

We compare four algorithms to proposed method, ExCuSe, a novel algorithm for robust pupil detection in real-world environments proposed in 2016, which is based on edge filtering and oriented histograms calculated via the Angular Integral Projection Function. DeepEye proposed in 2019, a deep convolutional neural network for eye-tracking based on atrous convolutions and spatial pyramids. DeepEye is able to handle varying illumination, blurring and reflections. Hough and integral projection algorithm which are widely used algorithms. MPIIGaze [33], [34] dataset was used to test the accuracy of pupil position. The MPIIGaze dataset contains 10,848 eye images collected from 15 participants during natural everyday laptop use over more than three months that are manually labeled with the left and right pupil positions. Euclidean distance was used to calculate the deviation of the measurement position from the label position. The dataset is significantly more variable than existing ones with respect to appearance and illumination which is similar with our environment. The dataset was annotated with 6 eye landmarks: the corners of the left and right eye(four) and the pupil centers(two). Whereas in our method, there are 12 landmarks is detected: the corners of the left and right eye(four), the eyelid landmarks of the left and right eye(eight). The width of the eye region and the start pixel are needed in the PNOP algorithm. Thus, the inner, outer, upper eyelid, and lower eyelid points are needed to locate the eye region. This is the main factor that affects the pupil position results, as illuminate in Figure 14. The results are listed in Table 5.

The performance of proposed method is better than other method. It is difficult to collect real environment Image, thus compare with DeepEye and other deep learn method,

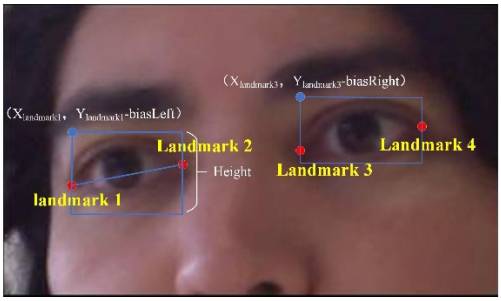


FIGURE 14. An illuminate result of eye region: start point and height.

PNOP algorithm do not need train the model. There are more than eight parameters needed for Excuse algorithm, different application environment needs different parameters, it is difficult to set suit parameters, whereas our method only need 3 parameters, biasLeft for short LB, biasRight for short RB and Height for short H. Hough and relative algorithms are sensitive to the change of illumination, meanwhile this algorithm requires a high quality original image and requires complicated parameter settings when performing detection. Therefore, it has limited use in practice. For vertical integral projection and the horizontal integral projection algorithm (for short VAH). The result shows that there are more candidate projection lines detected using VAH than with the proposed method. Moreover, VAH algorithm is sensitive to light changes. Meanwhile, the number of projection lines are not regular; thus, their result is random (not symmetrical) and is not informative. Meanwhile, iris size can be detected using the relationship between the peak points of the positive and negative oblique projection lines. No additional complicated algorithms are needed to calculate the size of the iris, and the complexity of the computation is low. The time consuming of the method is 71.8ms in our hardware setting. Result can be seen in Figure 15. Figure 16 depicts the performance in terms of the detection rate achieved with less than a certain pixel's distance between the output and the hand labeled ground true.

Pupil accuracy may influence by the glint in glasses. We use MPIIGaze dataset to test the performance rates with and without glasses. We manually divide all images into two

TABLE 6. Comparison of the accuracy with or without glass of different algorithms on MPIIGaze.

Parameters			Glass	Algorithm									
				DeepEye		Hough		Excuse		VAH		Proposed (PNOP)	
LB	RB	H	Y/N	LeftEye	RightEye	LeftEye	RightEye	LeftEye	RightEye	LeftEye	RightEye	LeftEye	RightEye
10	10	10	N	14.3678	14.6376	33.1104	36.7042	30.2181	31.4917	45.1915	115.5855	**5.6373**	**5.5263**
			Y	16.4663	14.3367	32.1220	34.5663	30.4337	29.2422	90.3008	102.8080	**8.1648**	**7.2367**
10	10	20	N	11.1506	10.9751	28.9024	32.5549	28.4568	29.4905	18.9273	42.1628	**5.3447**	**6.9488**
			Y	12.0020	12.3174	25.5913	31.5171	27.7208	27.1879	11.6724	15.6970	**7.6990**	**8.8607**
10	15	20	N	11.1506	11.8007	28.9007	30.8683	28.3327	32.0601	18.9273	24.7320	**5.3474**	**6.6850**
			Y	12.0020	13.0616	25.5913	29.9485	27.7683	29.8455	11.6724	12.0877	**7.7310**	**8.2131**
15	15	15	N	14.2850	14.6816	27.6403	33.5127	31.8014	32.8885	19.8456	49.6649	**6.2071**	**6.2967**
			Y	16.9640	14.3165	26.3552	32.3338	31.9167	30.3488	50.2019	56.3110	**8.5704**	**7.8359**
15	15	20	N	11.3630	11.8007	26.0022	30.8683	30.4309	31.9676	9.1762	24.7320	**5.3204**	**6.6825**
			Y	13.7381	13.0616	23.6519	29.9485	30.6353	29.7479	11.8018	12.0877	**7.5179**	**8.3047**
15	15	30	N	8.6410	8.7742	22.6492	24.9805	17.2149	18.5672	9.3810	12.1122	**5.3474**	**6.6850**
			Y	10.0076	10.4854	19.4991	24.7456	14.5271	14.0039	8.7610	9.8646	**8.9236**	**10.6008**
20	20	10	N	32.7149	31.7175	26.1186	35.3349	34.3185	35.0186	101.6163	130.2502	**17.8590**	**16.4949**
			Y	32.2985	28.5622	26.2712	33.6191	35.3118	33.3244	98.3826	180.6288	**17.9054**	**17.0623**
20	20	20	N	16.1607	16.6253	23.9432	30.2136	33.0203	33.9195	15.6336	25.9735	**6.8903**	**6.9977**
			Y	17.6458	15.7249	23.2132	29.6538	33.9761	31.8229	26.9013	36.7482	**9.5454**	**9.0046**
25	25	25	N	18.0389	18.3878	22.9548	27.7899	31.5209	30.9199	16.3953	19.0228	**8.0583**	**8.0098**
			Y	19.4998	16.5216	22.8359	27.4097	30.0941	26.7831	22.4700	21.1542	**11.5500**	**11.1676**
30	30	30	N	19.0114	19.5342	22.5301	24.4805	33.9446	32.0435	17.6531	17.4595	**10.0130**	**9.4072**
			Y	20.1336	17.3752	22.5271	23.6620	31.7006	27.4174	24.3975	19.4603	**14.7273**	**14.5666**

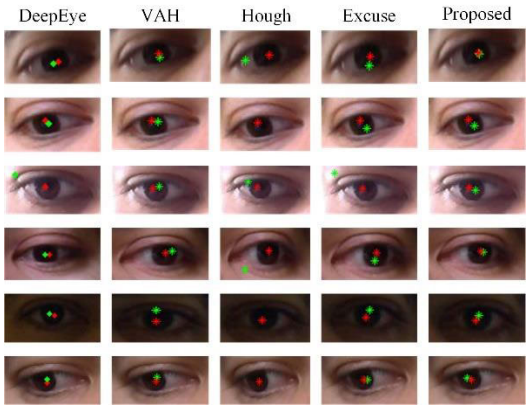


FIGURE 15. Comparation result with 5 algorithms red marks (⋆) are labeled in database and green marks (⋆) are calculated by algorithm.

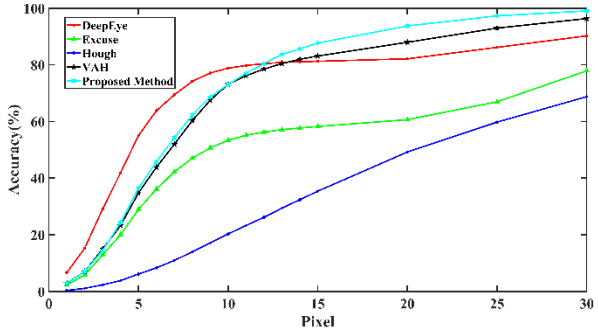


FIGURE 16. Detection error in function of the pixels distance in the cross validation compared with DeepEye, Excuse, Hough and VAH (LB = 15, RB = 15 and H = 30).

parts: with glasses and without glasses. MPIIGaze dataset collected from 15 laptop users over several months in their daily life and there are 5 users wear glasses (p01, p04, p07, p09, p10). Some example images with glasses or without

FIGURE 17. Some examples of images with or without glasses in MPIIGaze Dataset.

glasses can be seen in Figure 17 and the result can be seen in TABLE 6.

Accuracy and dispersion are metrics used to evaluate gaze tracking. Accuracy is the average error between the actual position of the calibration point and the gaze points. Dispersion is the degree of dispersion of an ordinary binocular camera as it continuously records the same fixation point. The standard deviation of the continuous sample is used to measure the degree of dispersion. Experiments were performed on a computer with a screen resolution of 1,920 × 1,080. The computer screen was divided into nine areas, and the central pixel of each area was marked as the label. Hence, the nine label points were observed by seven students and their images recorded using a monocular camera and binocular camera.

To avoid the large gaze error, the head movement is limited in a range. The x-translation, y-translation and z-translation is about ±16mm, ±16mm and ±27mm respectively. Each label point was observed for 10s. One student's observation result are shown in Figure 18(a). The average error and root mean square error of the points are reported. The accuracy and dispersion results are shown in Figure 18(b).

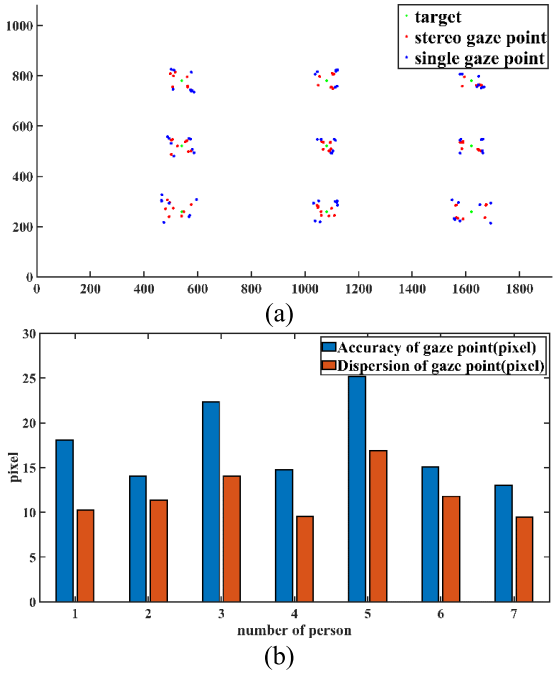


FIGURE 18. Results for accuracy and dispersion: (a) gaze position results for one-person (binocular camera and monocular camera) and (b) statistics of accuracy and dispersion for seven students.

Figure 18(b) shows that the mean dispersion between the gaze and label points ranges from 13 to 25 pixels. Moreover, the dispersion ranges from 9 to 16 pixels. Expression (13) is used to calculate degrees according to user to screen distance.

$$\theta = \tan^{-1} y/x \tag{13}$$

where y is the distance between calibration point and Gaze point and x is the distance between user and screen. With the ''user-screen'' distance between 60cm and 80cm, the accuracy can reach up to 1.5° to 2.2°.The eye does not view a single point, so the method proposed in this paper can meet the requirements for gaze tracking.

IV. CONCLUSION

To deal with pupil location when user sit in front of laptop, an algorithm PNOP is proposed in this paper, it can locate pupil accuracy under real-world environment and this algorithm is robust to illuminate change and wear glasses. Compare with deep learning method, it does not need several train images and the algorithm can not only achieve high accuracy but also short computing time. The proposed PNOP algorithm can precisely and accurately locate the pupil position by measuring the angle between the head and the front view (for accurate measurement, the angle between the head and the front view should be within 30°). The user-to-screen distance is measured using a binocular camera and a regression model was established to determine the distance and iris size. The regression model is also important to compensate for the iris measurement error caused by changes in the user-to-screen distance. Iris size is calculated according to the relationship between the peak value of the lines of each projection. Finally, the proposed method was used to estimate the gaze position based on images from a binocular camera and detect the regions of interest. The experimental results show that the proposed method can effectively estimate the gaze position. This will enable students' learning habits to be analyzed in a better way. There are three limitations of the approach. First, we segment eye region according to landmarks around eyes, start point, width and height of the eye region which may influence the accuracy of pupil location as the accurate segmentation of the eye area is a prerequisite for pupil location. Second, pupil size is very small and the pupil and iris are similar in color. Thus, it is difficult to locate pupil as the user is far from the screen. Third, when the light suddenly focus on a certain point of the eye, strong reflection may occur which may affect the accuracy of pupil location. In the future, these points should be focused.

REFERENCES

[1] M. Saqr, U. Fors, and M. Tedre, ''How learning analytics can early predict under-achieving students in a blended medical education course,'' *Med. Teacher*, vol. 39, no. 7, pp. 757–767, Jul. 2017.
[2] A. Susac, A. Bubic, M. Planinic, M. Movre, and M. Palmovic, ''Role of diagrams in problem solving: An evaluation of eye-tracking parameters as a measure of visual attention,'' *Phys. Rev. Phys. Edu. Res.*, vol. 15, no. 1, p. 13101, Jan. 2019.
[3] K. Kathiresh Vijayan, O. Jon Mork, and I. Emily Hansen, ''Eye tracker as a tool for engineering education,'' *Universal J. Educ. Res.*, vol. 6, no. 11, pp. 2647–2655, Nov. 2018.
[4] S. Zander, M. Reichelt, and S. Wetzel, ''Does personalisation promote learners' attention? An eye-tracking study,'' *Frontline Learn. Res.*, vol. 3, no. 4, pp. 1–13, 2015.
[5] M. Bienkowski, M. Feng, and B. Means, ''Enhancing teaching and learning through educational data mining and learning analytics: An issue brief,'' U.S. Dept. Edu., Office Educ. Technol., Washington, DC, USA, Tech. Rep., 2012, pp. 1–64.
[6] F. Jan, ''Pupil localization in image data acquired with near-infrared or visible wavelength illumination,'' *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 1041–1067, Jan. 2018.
[7] F. Lu, Y. Gao, and X. Chen, ''Estimating 3D gaze directions using unlabeled eye images via synthetic iris appearance fitting,'' *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1772–1782, Sep. 2016.
[8] J. Liang, M. Zhang, D. Liu, X. Zeng, O. Ojowu, K. Zhao, Z. Li, and H. Liu, ''Robust ellipse fitting based on sparse combination of data points,'' *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2207–2218, Jun. 2013.
[9] A. Laddi and N. R. Prakash, ''An augmented image gradients based supervised regression technique for iris center localization,'' *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 7129–7139, Mar. 2017.
[10] C.-N. Meng, ''Fast and precise iris localization for low-resolution facial images,'' *Opt. Eng.*, vol. 51, no. 7, Jul. 2012, Art. no. 077008.
[11] W. Fuhl, T. Kubler, K. Sippel, W. Rosenstiel, and E. Kasneci, ''Excuse: Robust pupil detection in real-world scenarios,'' in *Proc. Int. Conf. Comput. Anal. Images Patterns*. New York, NY, USA: Springer, 2015, pp. 39–51.
[12] F. J. Vera-Olmos, E. Pardo, H. Melero, and N. Malpica, ''DeepEye: Deep convolutional network for pupil detection in real environments,'' *Integr. Comput.-Aided Eng.*, vol. 26, no. 1, pp. 85–95, Dec. 2018, doi: 10.3233/ICA-180584.
[13] N. M. Arar, H. Gao, and J.-P. Thiran, ''A regression-based user calibration framework for real-time gaze estimation,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2623–2638, Dec. 2017.
[14] Y.-M. Cheung and Q. Peng, ''Eye gaze tracking with a Web camera in a desktop environment,'' *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 4, pp. 419–430, Aug. 2015.
[15] D. Torricelli, S. Conforto, M. Schmid, and T. D'Alessio, ''A neural-based remote eye gaze tracker under natural head motion,'' *Comput. Methods Programs Biomed.*, vol. 92, no. 1, pp. 66–78, Oct. 2008.

[16] B. Li, H. Fu, D. Wen, and W. Lo, "Etracker: A mobile gaze-tracking system with near-eye display based on a combined gaze-tracking algorithm," *Sensors*, vol. 18, no. 5, pp. 1–18, 2018.
[17] S.-W. Shih and J. Liu, "A novel approach to 3-D gaze tracking using stereo cameras," *IEEE Trans. Syst., Man Cybern. B, Cybern.*, vol. 34, no. 1, pp. 234–245, Feb. 2004.
[18] K. A. Funes Mora and J.-M. Odobez, "Gaze estimation from multimodal kinect data," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 25–30.
[19] J. Choi, B. Ahn, J. Parl, and I. S. Kweon, "Appearance-based gaze estimation using Kinect," in *Proc. 10th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Oct. 2013, pp. 260–261.
[20] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," in *Proc. Eur. Conf. Comput. Vis.*, vol. 4, 2008, pp. 504–513.
[21] B. Ahn, Y. Han, and I. S. Kweon, "Real-time facial landmarks tracking using active shape model and LK optical flow," in *Proc. 9th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Nov. 2012, pp. 1–3.
[22] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[23] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image Vis. Comput.*, vol. 47, pp. 3–18, Mar. 2016.
[24] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 896–903.
[25] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 397–403.
[26] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9905, 2016, pp. 349–364.
[27] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.
[28] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.
[30] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1867–1874.
[31] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," UMass Amherst, Amherst, MA, USA, Tech. Rep. 6, Jan. 2010.
[32] O. Jesorsky, K. Kirchberg, and R. Frischholz, "Robust face detection using the Hausdorff distance," in *Audio-and Video-Based Biometric Person Authentication*. Berlin, Germany: Springer, 2001, pp. 90–95.
[33] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proc. IEEE Comput. Social Conf. Comput. Vis. Pattern Recognit.*, vols. 07–12, Jun. 2015, pp. 4511–4520.
[34] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "MPIIGaze: Real-world dataset and deep appearance-based gaze estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 162–175, Jan. 2019.
[35] P. Getreuer, "Automatic color enhancement (ACE) and its fast implementation," *Image Process. Line*, vol. 2, pp. 266–277, Nov. 2012.
[36] H. Bao, W. Fang, B. Guo, and J. Wang, "Real-time wide-view eye tracking based on resolving the spatial depth," *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 14633–14655, Jun. 2019.

JUNJIE ZHANG was born in Beijing, China, in 1993. He received the B.Sc. degree from the Beijing University of Technology, Beijing, in 2015. He received an opportunity of combined master's and Ph.D. degrees from the Beijing University of Technology, where he is currently pursuing the Ph.D. degree.

GUANGMIN SUN was born in Shanxi, China, in 1960. He received the B.Sc. degree in electronic engineering from the Beijing Institute of Technology, Beijing, China, in 1982, the M.Sc. degree in communication and information systems from the Nanjing University of Science and Technology, Nanjing, China, in 1991, and the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 1997.

He is currently a Professor with the Beijing University of Technology. His current research interests include neural networks and applications, image processing, and pattern recognition.

KUN ZHENG was born in Hebei, China, in 1977. He received the B.Sc. degree in electronic engineering from Hebei University, Hebei, in 2001, and the M.Sc. degree in software engineering and the Ph.D. degree in electronic engineering from the Beijing University of Technology, Beijing, China, in 2006 and 2018, respectively.

He is currently an Associate Professor with the Beijing University of Technology. His current research interests include neural networks and applications, image processing, and intelligent education.

SARAH MAZHAR received the M.S. degree in computer science from the National University of Modern Languages, Islamabad, Pakistan. She is currently pursuing the Ph.D. degree with the Beijing University of Technology, Beijing, China.

She is a Lecturer of computer science with the Faculty of Engineering and Computer Science, National University of Modern Languages, H-9, Islamabad, Pakistan. Her research interests include neural networks, image processing, remote sensing, emotion-based requirement engineering, green computing, and fog computing.