Yu, Pingping & Duan, Wenjie & Sun, Yi & Cao, Ning & Wang, Zhenzhou & Lu, Guojun. (2020). A Pupil-Positioning Method Based on the Starburst Model. Computers, Materials & Continua. 64. 1199-1217.

Available online at: https://doi.org/10.32604/cmc.2020.010384

Computers, Materials & Continua CMC, vol.64, no.2, pp.1199-1217, 2020

A Pupil-Positioning Method Based on the Starburst Model

Pingping Yu[1], Wenjie Duan[1], Yi Sun[2], Ning Cao[3, *], Zhenzhou Wang[1] and Guojun Lu[4]

Abstract: Human eye detection has become an area of interest in the field of computer vision with an extensive range of applications in human-computer interaction, disease diagnosis, and psychological and physiological studies. Gaze-tracking systems are an important research topic in the human-computer interaction field. As one of the core modules of the head-mounted gaze-tracking system, pupil positioning affects the accuracy and stability of the system. By tracking eye movements to better locate the center of the pupil, this paper proposes a method for pupil positioning based on the starburst model. The method uses vertical and horizontal coordinate integral projections in the rectangular region of the human eye for accurate positioning and applies a linear interpolation method that is based on a circular model to the reflections in the human eye. In this paper, we propose a method for detecting the feature points of the pupil edge based on the starburst model, which clusters feature points and uses the RANdom SAmple Consensus (RANSAC) algorithm to perform ellipse fitting of the pupil edge to accurately locate the pupil center. Our experimental results show that the algorithm has higher precision, higher efficiency and more robustness than other algorithms and excellent accuracy even when the image of the pupil is incomplete.

Keywords: Human eye localization, ellipse fitting, pupil contour, pupil center.

1 Introduction

With the rapid development of virtual reality (VR) technology, an increasing number of VR products exist on the market; however, optimal solutions for human-computer interaction and dizziness caused by the use of VR devices have not been obtained. One of the main reasons is that VR devices encounter difficulties when responding to of the human head in a timely manner. Eye movement is one of the fastest movements of the human body. Applying eye tracking technology to monitor eye movement in real time will help solve problems associated with VR devices. Eye feature extraction is an important component of eye tracking technology. For this component, precise positioning

[1] School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050000, China.

[2] Hebei Electric Power Research Institute, Shijiazhuang, 050022, China.

[3] School of Internet of Things and Software Technology, Wuxi Vocational College of Science and Technology, Wuxi, 214028, China.

[4] School of Science, Engineering and IT, Federation University Australia, Gippsland Campus, Churchill, Australia.

[*] Corresponding Author: Ning Cao. Email: ning.cao2008@hotmail.com.

Received: 01 March 2020; Accepted: 17 April 2020.

CMC. doi:10.32604/cmc.2020.010384 www.techscience.com/journal/cmc

of the pupil is important.

The most commonly employed human eye-detection methods are based on computer vision technology [Wu and Jin (2018); Zheng and Wang (2012)]. This paper focuses on human eye detection methods that are based on computer vision. Liu et al. [Liu, Dong, Liu et al. (2018)] proposed a relocation tracking algorithm based on the adaptive boosting (AdaBoost) algorithm, random forest (RF) model and spatiotemporal context (STC) method. This study aims to track the face by recognizing the face and using the RF algorithm to locate the eye. Tian et al. [Tian and Wang (2018)] proposed a human eye localization and open-or-close recognition algorithm based on the gray value changes in various parts around the eye. The method uses the AdaBoost algorithm to roughly position the human eye, and the features of the eye are positioned by using the horizontal integration method. Raudonis et al. [Raudonis, Simutis and Narvydas (2010)] proposed a method that uses principal component analysis (PCA) to determine the six major components in a human eye image and locate the pupil with artificial neural networks (ANNs). Tang et al. [Tang and Zhang (2009)] proposed a human eye-detection algorithm combined with a grayscale prediction model. The estimated human eye position can be employed as a reference coordinate for detecting the eye based on an entire image. Fu et al. [Fu and Yang (2011)] proposed a high-performance human eye detection algorithm for establishing eye detection templates on the left and right eye. The method establishes a two-dimensional orthogonal correlation function between the matching template and the face region after the face region was detected. The function applies edge detection and Hough circle detection for iris [Ding, Tian, Han, et al. (2019)] edge extraction and locates the pupil center. Mehrubeoglu et al. [Mehrubeoglu and Muddu (2011)] proposed a system for detecting eyes using a pattern recognition algorithm that can record an eye image of the tester in real time to quickly extract the necessary information and locate the eye. Wang et al. [Wang and Chang (2011)] have proposed a method of human eye location based on area projection. By calculating the two-dimensional features in the projection process, the face image is segmented in the vertical and horizontal directions, and the boundary tracking is used to achieve the positioning of the human eye pupil. For the problem that the traditional CamShift algorithm only calculates color features, Huang et al. [Huang, Sang, Hao et al. (2014)] proposed an improved CamShift algorithm that combines color features and edge features to track the pupils and improve the human eye detection rate. Zhang et al. [Zhang, Chi, Zhang et al. (2010)] utilized corneal reflection to locate the human eye and identified the pupil center by determining the angle between the corneal reflection and the pupil center with the fixation point. Yang et al. [Yang, Sun, Liu et al. (2010)] proposed a method for human eye detection using the grayscale differences among facial, pupil and corneal reflections. However, the test for a facial image with glasses did not yield the ideal detection result, which revealed its relatively poor robustness. Yan et al. [Yan and Wu (2018)] proposed a method for accurate positioning of the pupil center based on the extraction of face feature points. The geometrical features of the pupil and the direction information of the human eye are used to determine the position of the pupil through the ray method with direction information and the improved RANSAC method, but the algorithm is easy disturbed by external factors such as light and noise. Li et al. [Li, Gao and Qian (2016)] collectd facial images through a USB camera, and uses the Adaboost detection algorithm and gray projection human eye positioning algorithm to determine the position of the pupil of the human eye. Fuhl et al. [Fuhl, Tonsen, Bulling et al. (2016)] employed an additional method to propose a strategy for edge selection. This method is based on the application of human-computer interaction or attention analysis to the line of sight. A head-mounted camera is required for eye tracking, and by exploring the appearance and strictly dependent threshold of the figure, the pupil position can be obtained. Li et al. [Li, Li, Tong et al. (2018)] proposed a pupil detection method based on a near-infrared head-mounted camera. By establishing an eyeball model, the method utilizes a pupil changing model in the head-mounted camera and the edge gradient of the circular pattern to obtain the pupil contour and determine the pupil position.

Based on the insufficiencies of the current public algorithm, a high-precision, high-efficiency and robust pupil location is proposed based on the combination of physiological characteristics of the human eye and advanced technology in the field of information technology. First, we locate the eye on a face image based on the skin color model and approximately locate the pupil region through the integral curve; second, the reflections in the eye are detected and filled, and then the feature points on the pupil edge are detected using the starburst model. The ellipse, where the pupil edge points are located, can be accurately fitted based on the feature point clustering of the high-density connected region; thus, the method can locate the pupil center. The implementation of this algorithm is to combine the physiological characteristics of the human body structure, eliminate the interference of various complex factors as much as possible, and effectively locate the pupil boundary and pupil center even if additional interference factors are not present.

This paper is organized as follows: Section 2 describes the position of the human eye based on integral projection. Section 3 describes the processes of detecting and filling the reflections in the human eye. Section 4 introduces the pupil-positioning algorithm based on the starburst model and high-density connected region clustering. Section 5 introduces the simulation experiments and results analysis. A conclusion can be found at the end of the paper.

2 Positioning of the human eye based on integral projection

2.1 Face segmentation based on the skin color model

Skin color is an important feature in face detection. Brightness is the greatest difference in the color difference between the skin color and the object background. No correlation exists between luminance information and chromaticity information, the clustering of skin color is satisfactory, and the calculation is simple. Therefore, segmentation of the human face based on the skin color model is feasible.

Statistics show that brightness is the main factor affecting the skin of people of different races, and the influence of chromaticity is relatively small. Thus, we use only the Cb-Cr component as the recognized color feature in this thesis. The Cb-Cr space is two-dimensionally distributed, with slight changes in brightness. When a pixel falls within $Cb \in [85,135]$, $Cr \in [-Cb + 260, -Cb + 280]$, it is considered a skin color pixel [Hu, Deng, Xiao et al. (2020)]. The chromaticity value is the value that best relates the skin colors of different people, and the difference in brightness is relatively large. Skin color is concentrated on a two-dimensional (2D) chromaticity plane, and the Gaussian distribution can effectively describe it [Chen, Zhang and Yang (2017)]. The center of the Gaussian distribution of skin color in chromaticity space can be obtained using a large number of statistical training methods. The skin color similarity is calculated by using the distance to all the pixels in the image from the center of the Gaussian distribution. The skin color similarity grayscale image [Li and Tang (2018)] can be obtained by dividing the similarity of each pixel by the maximum similarity.

The similarity calculation formula is shown in Eq. (1):

$$\boldsymbol{P(CbCr) = \exp[-0.5(x-m)^T C^{-1}(x-m)]} \tag{1}$$

where m is the mean, $m = E(x)$, and C is the covariance matrix,

$$x = (CbCr)^T, C = E[(x-m)(x-m)^T] \tag{2}$$

The 2D distribution Gaussian model of skin color is shown in Eq. (3);

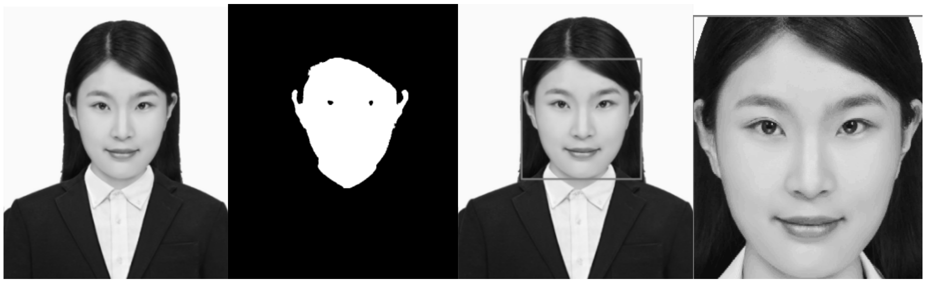
$$\overline{Cb} = \frac{1}{N}\sum_{i=1}^{N} C_{bi}, \overline{Cr} = \frac{1}{N}\sum_{i=1}^{N} C_{ri} \tag{3}$$

$$m = (\overline{Cb}, \overline{Cr}) \quad C = \begin{bmatrix} \delta_1^2 & \rho\delta_1\delta_2 \\ \rho\delta_1\delta_2 & \delta_2^2 \end{bmatrix} \tag{4}$$

where $\overline{Cb}$ and $\overline{Cr}$ are the means of Cb and Cr, respectively, and C is the covariance matrix. After training is performed based on the skin color points, the parameters of the skin color model are selected:

$$m = \begin{pmatrix} 117.4361 \\ 148.5599 \end{pmatrix} c = \begin{pmatrix} 260.1301 & 12.1430 \\ 12.1430 & 150.4574 \end{pmatrix}$$

If the final value is within the range, then it is regarded as a skin color point. The divided image is shown in Fig. 1(b). The white area represents regions that are skin color. The face area is positioned according to the aspect ratio of the face, as shown in Fig. 1(c), and the divided image is shown in Fig. 1(d).



(a) Original image, (b) skin color model segmentation, (c) face positioning, and (d) face

Figure 1: Human face segmentation

2.2 Positioning of the human eye

The extracted face is binarized, as shown in Fig. 2. Since the grayscale values from the hair, eyebrows, and eyes are low, we can use these features to separate the eyes from other features.

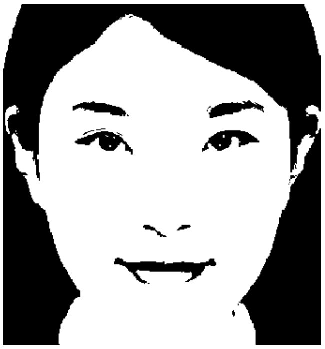


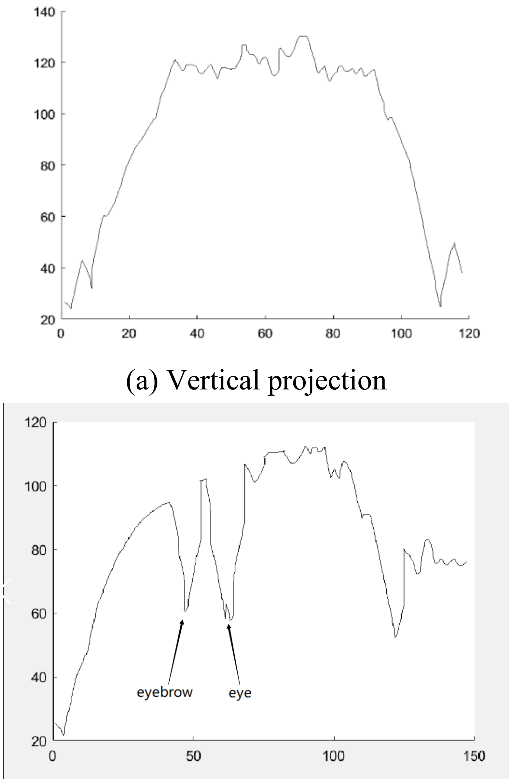
Figure 2: Otsu binarization

An integral projection is made of the face image. Let the gray value at the image (x, y) be $G\ (x, y)$, $H(x)$ be the horizontal integral projection of the image region $[y_1, y_2]$, and $V\ (y)$ be the vertical integral projection of the region $[x_1, x_2]$.

$$H(x) = [\frac{1}{x_2 - x_1}]\sum_{x_1}^{x_2} G(x, y) \tag{5}$$

$$V(y) = [\frac{1}{y_2 - y_1}]\sum_{y_1}^{y_2} G(x, y) \tag{6}$$

The obtained horizontal integral projection curve and vertical integral projection curve are shown in Figs. 3(a) and 3(b), respectively. Because the grayscale of the facial region is different from that of the surroundings, the vertical integral projection curve has a convex peak with a certain width, and the left and right boundaries of the peak are approximately the left and right boundaries of the human face, respectively. According to the two boundaries, the horizontal coordinate value of the face region can be obtained and set to (x_1, x_2). According to prior knowledge of the face image, the black areas near the hair in the face image are the eye and the eyebrow. The eyebrow is at the first minimum point, and the eye is at the second minimum point, as shown in Fig. 3(b). The vertical coordinate of the obtained eyebrow is y_1, and the vertical coordinate of the eye is y_2. According to the geometric features of the face, y_2 is the center of the rectangular area δ, with a width of $|y_2 - y_1| / 2$ in the longitudinal direction of the face image, and the area of the human eye is depicted as shown in Fig. 4. We use $1 / 15$ filtering of the image height.



(a) Vertical projection

(b) Horizontal projection

Figure 3: Integral projection of the face region

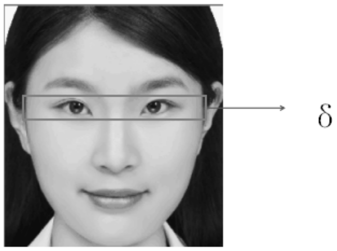
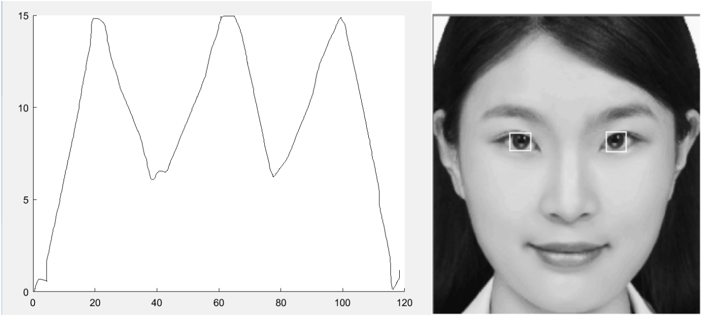


Figure 4: Eye region

To find the positions of the eyes in the rectangular region δ , we need to locate the eyes through the integral curve and perform vertical integration on the region δ . The curve has very stable symmetry, and we can easily determine its extreme values. The gray value of the eyeball is at the lowest point; thus, we can determine the left and right minimum symmetrical points of the curve to determine the positions of the human eyes. The projected curve is shown in Fig. 5(a), and the eye positions are shown in Fig. 5(b).



(a) (b)

Figure 5: Vertical integral projection of the rectangular area of the human eye

3 Reflection detection and filling

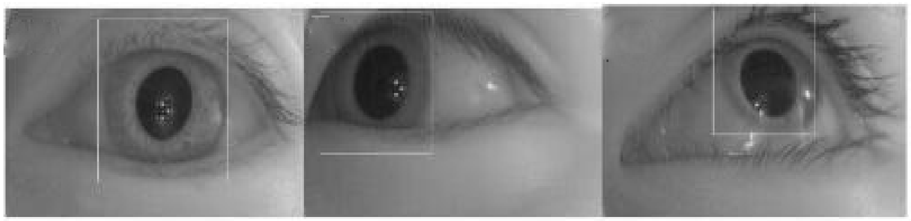


Figure 6: Amplified eye area

An amplified view of the eye area is shown in Fig. 6. When taking the picture, some of the light will inevitably be reflected by the surface of the cornea. The small and bright circular areas on the eyeball image are the corneal reflections. The presence of reflections destroys the physiological structure of the pupil and affects the positioning of the pupil center. To ensure the performance of the algorithm, it is necessary to accurately locate the reflections and eliminate them. This thesis uses the threshold binarization method to detect the reflections, which are then filled via linear interpolation.

3.1 Detection of reflection areas

The gray value of the reflection point is very different from that of the eyeball. Based on this characteristic, the gray threshold histogram method is used to segment the reflection point, and the reflection point is segmented by selecting the appropriate threshold. The region of the reflection point is filled by the linear interpolation method based on the circular model.

Step one: Binarization

Select an appropriate threshold for the eye region in Fig. 6 for binarization. The selection of the threshold is not universal. If the selected threshold is low, then the area around the eye will be mistaken as a reflection because of its higher brightness. In contrast, if the threshold is too high, then the reflection may be hidden, which causes some reflections to be left unfilled. Values should be based on specific research data. In this thesis, an appropriate threshold is selected by using the gray histogram of the eye image. The eye region histogram is shown in Fig. 7. By analyzing the Fig. 7, $T_{ref} = 200$ is selected as the threshold of binarization.

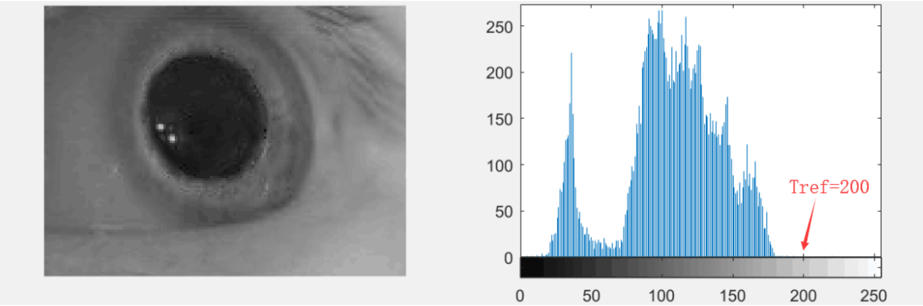


Figure 7: Eye area and eye grayscale histogram; the binarization threshold is 200

The pixel value of each point in the human eye region is compared with the threshold value. If the pixel value is less than the threshold, then the gray value is 0 (all black). If the pixel value is greater than the threshold, then the gray value is 255 (all white). This comparison enables us to obtain a two-value image, which is also referred to as the black-and-white image. The pixel with the highest gray value in the binarized image is then treated as a reflection pixel.

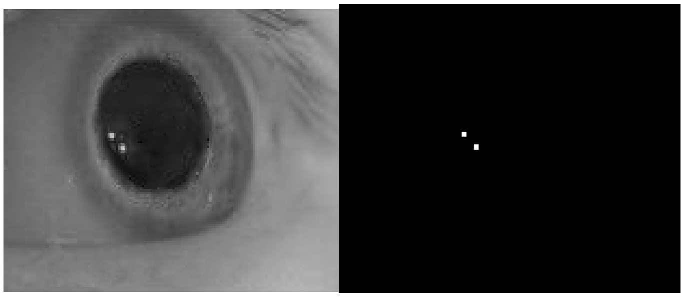

(a) (b)

Figure 8: (a) eye region; (b) reflections in an eye region that are binarized by $T_{ref} = 200$

Step two: Expansion

The binarized pixels are expanded using a 3×3 structural element. Each pixel in the eye region is scanned after binarization, and the structural element is used to perform an AND operation with the binarized image it encompasses. If the result of the operation is 0, then the pixel value in the binarized eye image is set to 0. Conversely, if the result of the AND operation is 1, then the pixel value in the structure image is set to 1. The result of the expansion operation is to expand the image of the target area to several adjacent pixels.

Step three: Extract the outline of the reflection area

The contour area of the reflections is limited to ensure that the reflections can be effectively extracted. Assume that the area of each contour that we detect is

$ContourArea(R_i)$. When the radius R_i satisfies certain conditions, the area is considered a reflection area. The Eq. (7) is expressed as

$$is\_reflection(R_i) = \begin{cases} true, 3 \le R_i \le 10 \\ false, other \end{cases} (i = 1, 2, ..n) \tag{7}$$

where $R_i = \sqrt{\frac{CountourArea(R_i)}{\pi}}$. When the contour radius is in the range $[3,10]$, it is regarded as the reflection area; otherwise, it is not.

3.2 Filling the reflection areas

The closer the relative distance between two pixels is, the smaller the difference between them is. When the reflections are filled, most of them are based on the pixel value information in the adjacent area. Here, we consider Fig. 9 as an example to describe the process of filling the reflections with linear interpolation based on the circular model.

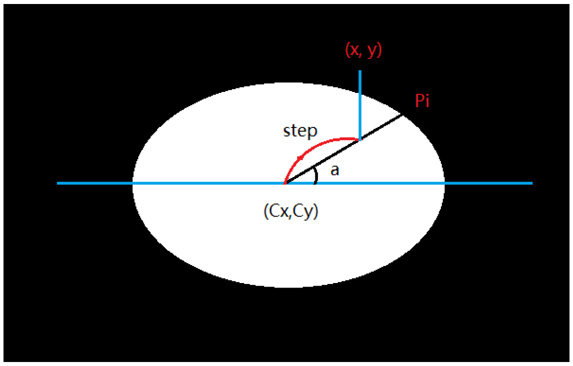


Figure 9: Filling the reflections

For the circular reflection filling area, the center point is set to (C_x, C_y) and it is divided into 360 parts (one for each degree). Here, we consider radius r_i and form the angle α with the horizontal axis. The intersection of this radius and the circle is $P_i(e_{xi}, e_{yi})$. Obtaining the pixel value v_i of the point and sequentially performing the same operation, we can obtain the two-dimensional vector set $angels$ $(i = 0...360)$, which is composed of (α_i, v_i). Add v_i and $average$ for the average inserted pixel agv. Linear interpolation is performed on the set $angels$ along each radius. Considering the radius at the angle α as an example and assuming that any point (x, y) to be inserted along the radius is at the distance $step$ from the center of the circle, the pixel calculation formula at point (x, y) is expressed as shown in Eq. (8):

$$d_{(x,y)} = \frac{step}{r} \times v_i + (1 - \frac{step}{r}) \times agv \tag{8}$$

Using this formula, we can perform the same pixel filling process for each point on each of the 360 portions of the circular area, and the entire reflection area can be interpolated. The filled eye area is shown in Fig. 10(a). After filling, the eye area is transformed into a Gaussian blur to reduce noise and smooth the pupil boundary. The Gaussian blurred image is shown in Fig. 10(b).

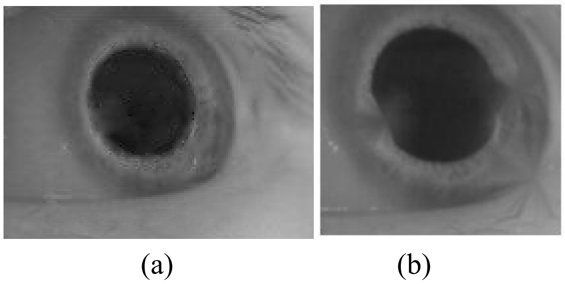

(a) (b)

Figure 10: (a) Eye area after filling the reflections and (b) effect of the Gaussian blur

4 Ellipse fitting for the pupil edge and pupil positioning

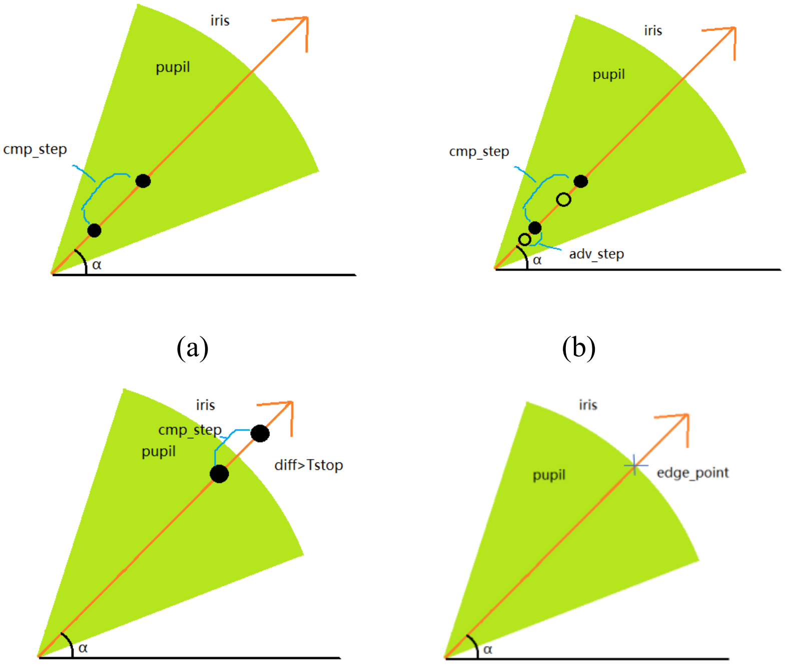
The algorithm in the previous section detects and interpolates the pupil reflex point in the image, ensures the smoothness of the pupil edge, and creates suitable conditions for the ellipse fitting of the pupil edge and the location of the pupil center. In this section, we use the starlight model to detect the feature points at the pupil edge, cluster the feature points in the high-density connection area to exclude the interference points that are not at the pupil boundary, and replace them with the surrounding adjacent points. The RANSAC algorithm is used to fit the pupil boundary. The ellipse model with the most points on the edge is selected as the pupil boundary, and the center of the circle where the ellipse is most suitable is the pupil center.

4.1 Using the starburst model to detect pupil edge feature point figures

In this paper, we use the starburst model to detect the feature points of the pupil edge. During detection, the thread division angle seg_n is set, and the age detection area is divided into $360/seg\_n$ threads that are executed concurrently. The use of multithreading detection increases the processing speed.

While processing each thread, a ray is emitted every degree from $i \times seg\_n$ to $(i+1) \times seg\_n$ along the geometric center (c_x, c_y) of the eye region. During the extension process of the ray, the comparison step length $cmp\_step = 9$, the advance step length $adv\_step = 2$, and the initial comparison pixel value $start\_value$ is the pixel value v of the geometric center point of the eye region. When the pixel difference value between the two extended end points is greater than the set value T_{stop}, the extension is stopped. The stop threshold value T_{stop} is set to the same value set in the reflection detection process, and the grayscale gradient feature extraction of the eye image is employed.

Fig. 11 shows the extension process of each ray by considering a ray at a horizontal angle of α (indicated by an orange line in the figure). A similar extension of each ray is performed to obtain all candidate feature points. Fig. 12 displays a flow chart that shows the entire process of edge point extraction, and the extracted result is shown in Fig. 13.



(a) (b)

(c) (d)

Figure 11: Extension of each ray when the starburst model detects the feature points at the edge of the pupil. (a) Assuming that a black point on the orange ray is the reference point, the distance between the point and the next point is cmp_step. (b) The reference point extends forward by adv_step pixels for comparison. (c) Extension is performed in sequence until the distance difference between the two endpoints is greater than the stop threshold. (d) The final boundary point of the extension is marked with a blue plus sign

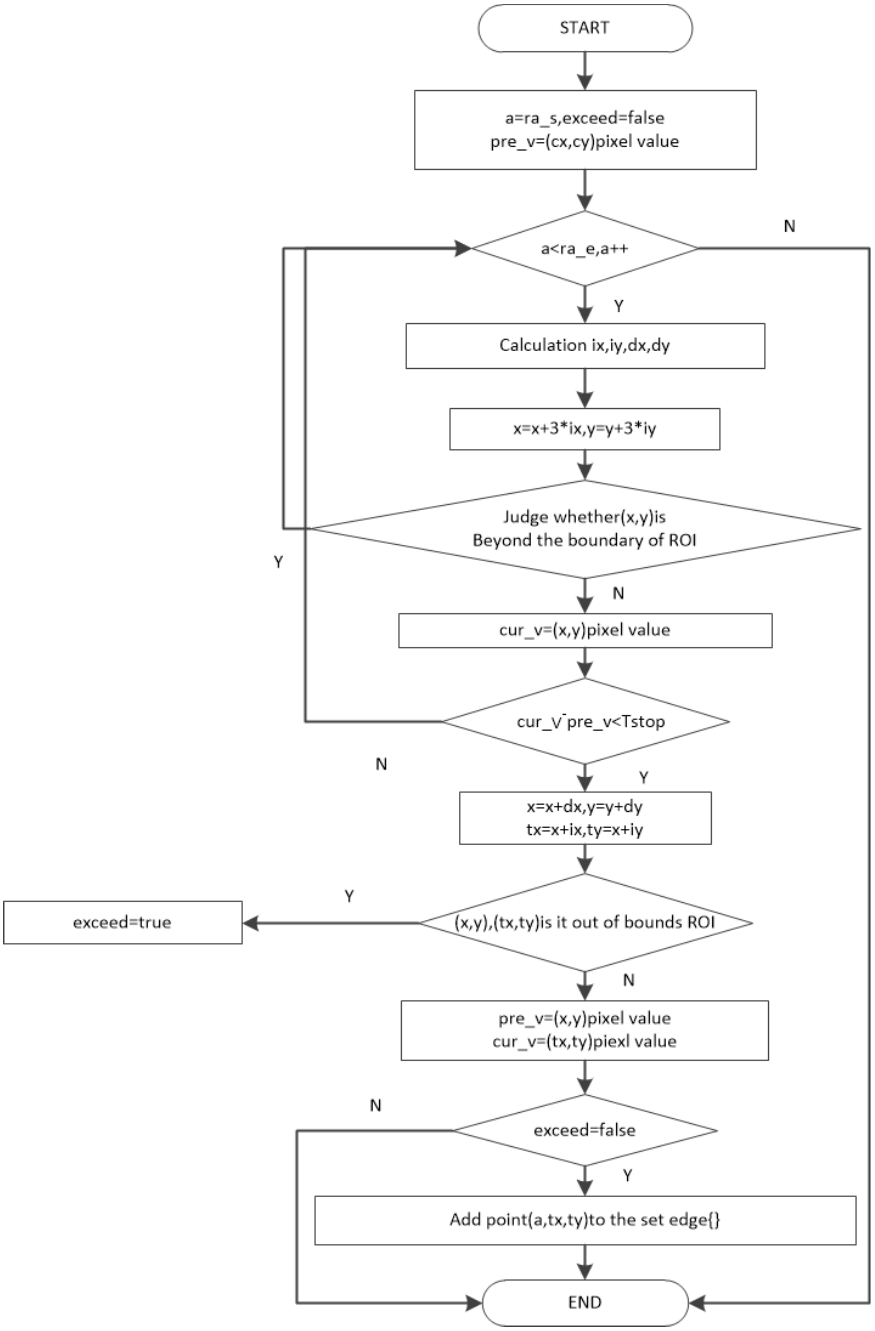


Figure 12: Edge detection flow chart

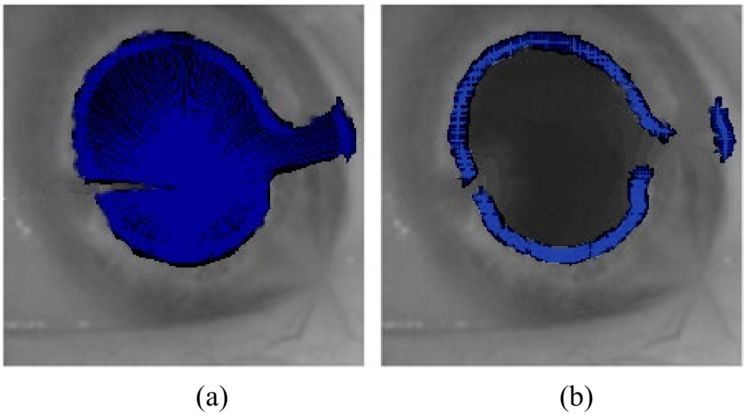


Figure 13: (a) Starburst model is used to detect pupil edge feature points, and (b) only the pupil edge feature points are retained

After edge detection, the set of edge feature point candidates is expressed as shown in Eq. (9):

$$edge = \{(\alpha_1, x_1, y_1), (a_2, x_2, y_2 ......(\alpha_n, x_n, y_n))\} \tag{9}$$

where α_i is the rotation angle; (x_i, y_i) and $(i = 1,2,3, \ldots, n)$ are the abscissa and ordinate, respectively, of the candidate edge feature points; and n is the number of candidate feature points.

4.2 Feature point clustering based on high-density connected regions

As shown in Fig. 13, the distribution of edge feature points detected by the starburst model is not continuous, and some nonconnected regions exist in Fig. 13. Some points are located far from the pupil boundary in the skin region near the eye. We need to cluster the feature points of the high-density connected region, eliminate the interference points that are not along the pupil boundary and replace them with the neighboring points.

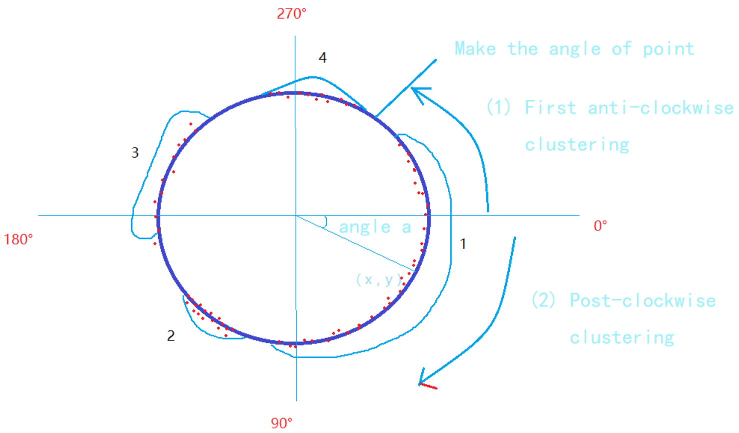


Figure 14: Feature point clustering

Now, the clustering process is explained in detail according to Fig. 14:

Step 1: Sort the candidate feature point sets by angle:

$$edge = \{(0^{o}, x_1, y_1), (1^{o}, x_2, y_2 ...... (359^{o}, x_n, y_n))\}$$

where n is the number of points in the set of candidate edge points.

Step 2: Perform counterclockwise clustering

(1): Initialize

$\min\_dist, \max\_dist, pre = 0, ne = n - 1, n\_ne = n - 2,$

$sub\_c = \{\}, category = \{\}$

(2): $p\_po\mathrm{int} = edge[0], next = edge[ne], n\_next = edge[n\_ne]$;

(3): Calculate the distance between $p\_po\mathrm{int}$ and $next$ as d1 and its distance to n_next as d2.

(4): If $d1 < \max\_dist, d2 > \min\_dist$, add the midpoint between $p\_po\mathrm{int}$ and n_next to the set of candidate subclasses sub_c; if $sub\_c.size > 20$, then add the points in sub_c to set $cateogry$.

(5): If $pre = ne, ne = n\_ne, n\_ne = ne - 1$, then repeat (2).

Step 3: Record the angle of the first counterclockwise clustering α and invert the points in the set sub_c. If $n\_ne = -1$ and the number in sub_c is greater than 20, then add the points in sub_c to the set $cateogry$ and return; otherwise, execute Step 4.

Step 4: Go to the point at the angle α, and from this point, start the clockwise clustering process of the remaining points in $edge$. The clustering step is the same as that in Step 2.

Step 5: Return to the set $cateogry$ and stop the clustering process.

The clustering results are expressed as follows:

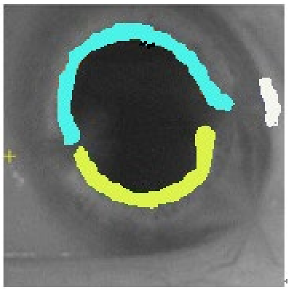


Figure 15: Results of feature point clustering using the high-density connected area clustering algorithm

According to these steps, after clustering, the class with the most elements is selected from the clustering results (as shown in Fig. 15; orange subclasses). If the central angle between the first element and final element is greater than 160 degrees, as shown in Fig.

16, then the subclass is considered a trusted subclass that is employed as a candidate subclass foredge fitting.

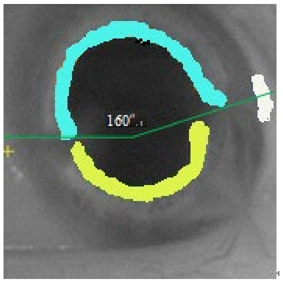
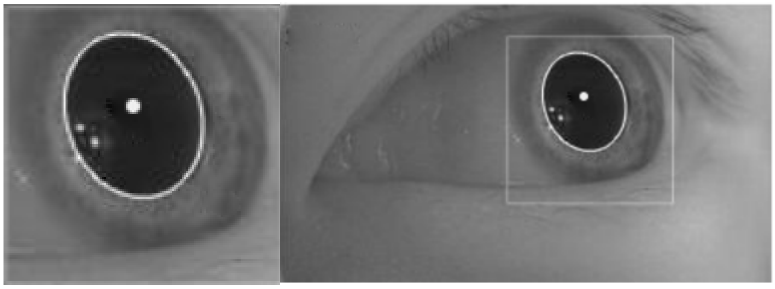


Figure 16: Center angle of the largest subclass

4.3 Pupil positioning

We employ the RANdom SAmple Consensus (RANSAC) algorithm to perform ellipse fitting along the pupil edge. The RANSAC algorithm is a random sampling consistency method that chooses a very small number of random subsets each time for fitting and applies each of them for modeling. For the same geometric model, the adjacent points are selected for fitting, and discrete points that are very far away are discarded. The geometric model with the best consistency is selected as the final model for fitting.



(a) (b)

Figure 17: Edge fitting results

5 Results and analysis

To objectively evaluate the effect of pupil contour fitting and pupil center location, this paper selects the correct detection number, error detection number, missed detection number, eye image number without pupil and detection image time to evaluate the difference among the Fuhl et al. [Fuhl, Tonsen, Bulling et al. (2016)] algorithm, Li et al. [Li, Li, Tong et al. (2018)] algorithm and this algorithm. We utilized the eye movement video collected by the Computer Application Technology Database of Hebei University of Science and Technology as the standard for testing the algorithm performance. In the experiment, the frames of each video are read and processed as eye image frames. The size of each frame is 768×432 pixels. We tested 800 images for the various positions of the pupil: centered pupil, pupil looking to the left or right and pupil looking to the top or bottom. Some of the results are shown in Fig. 18. According to the results, in all cases, the method can effectively determine the center of the pupil.

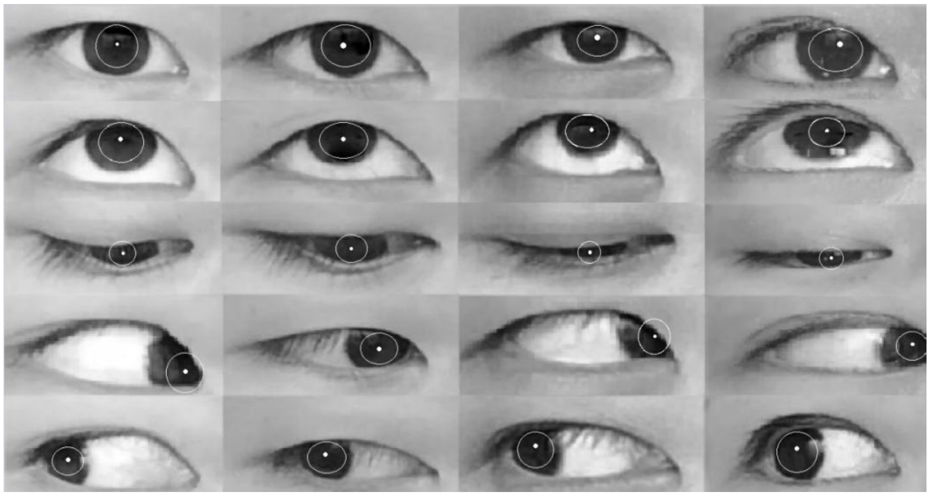


Figure 18: Pupil positioning for different pupil positions

For the results, the number of correct detections from all frame images for each video and those of failed detections and omitted detections are counted. While capturing the video, we occasionally detect some closed eyes. We count these images as eye images without pupils; they are not included in the accuracy calculation. We also calculated the correct detections using the Fuhl et al. [Fuhl, Tonsen, Bulling et al. (2016)] algorithm and Li et al. [Li, Li, Tong, et al. (2018)] algorithm, which provide the correct detection rate and the average detection time per frame, respectively. The comparison results are listed as follows:

Table 1: Statistical comparison of results

Algorithms	Total Frame (frames)	Correct (frames)	Failed (frames)	Omitted (frames)	Nonpupil (frames)	Average Detection Time per frame (ms)
This paper	358	329	15	9	5	178.6
	291	281	5	5	0	216.9
	280	266	7	3	4	203.8
	293	275	10	6	2	190.2
[Fuhl, Tonsen, Bulling et al. (2016)]	358	244	20	79	15	203.6
	291	255	25	11	0	240.3
	280	247	13	18	2	220.5
	293	204	43	40	6	215.2
[Li, Li, Tong et al. (2018)]	358	230	24	84	20	267.8
	291	245	26	10	10	241.3
	280	257	12	9	2	243.7
	293	195	42	40	16	256.9

Based on the experiments in Tab. 1, the average correct detection rate and average detection time per frame of the three algorithms are calculated. The calculation results are shown in Tab. 2. The results show that the proposed algorithm has higher accuracy and a higher detection speed than the other two algorithms. Our algorithm has excellent detection performance; Fig. 18 shows an example of the detection results based on this algorithm.

Table 2: Statistical comparison between average accuracy and detection time

Algorithms	Average Correct Rate (%)	Average Time per Frame (ms)
This paper	95.18	197.38
[Fuhl, Tonsen, Bulling et al. (2016)]	79.65	219.90
[Li, Li, Tong et al. (2018)]	79.43	252.43

6 Conclusions

A pupil-positioning algorithm, which is based on the starburst model and high-density connected region clustering, is proposed in this paper. The algorithm employed Cb-Cr color spaces to separate the skin color and background areas in images of the human face and accurately detects the positions of human eyes via horizontal and vertical integral projections. We detect reflections on the eyeball and fill them using linear interpolation based on circular models. We proposed a pupil-positioning algorithm that is based on the starburst model and high-density connected region clustering. The starburst model was used to detect the feature points along the pupil edge, and high-density connected region clustering was used to cluster the detected feature points. We performed ellipse fitting along the pupil edge with the RANSAC algorithm to identify the position of the pupil center. The experimental results show that, compared with other methods, the correct detection rate of this paper is increased by 18.8% and the average detection time per frame is reduced by 19.3%. The findings reveal that the algorithm not only improves the detection accuracy but also shortens the detection time. However, the algorithm has some shortcomings. When the edge of the pupil is blurred or the pupil is blocked by the upper and lower eyelids, the detection accuracy is moderately affected. This issue will be the focus of our next study.

Funding Statement: This research was funded by the Science and Technology Support Plan Project of Hebei Province (grant numbers 17210803D and 19273703D), the Science and Technology Spark Project of the Hebei Seismological Bureau (grant number DZ20180402056), the Education Department of Hebei Province (grant number QN2018095), and the Polytechnic College of Hebei University of Science and Technology.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chen, W. D.; Zhang, Y.; Yang, X. L. (2017): Face detection method based on skin color characteristics and depth model. *Industrial Control Computer*, vol. 30, no. 3, pp. 26-28.

Ding, Y.; Tian, L.; Han, B.; Wang, H. Y.; Wang, Y. J. et al. (2019): Achieving privacy-preserving iris identification via el Gamal. *Computers, Materials & Continua*, vol. 61, no. 2, pp. 727-738.

Fu, B.; Yang, R. (2011): Display control based on eye gaze estimation. *International Congress on Image and Signal Processing*, pp. 399-403.

Fuhl, W.; Tonsen, M.; Bulling, A.; Kasneci, E. (2016): Pupil detection for head mounted eye tracking in the wild. *An Evaluation of the State of the Art*, vol. 27, no. 8, pp. 1275-1288.

Hu, L.; Deng, J.; Xiao, Y. X.; Lu, Y. Z.; Zeng, Z. (2020): Face detection based on YCbCr and AdaBoost algorithm. *Artificial Intelligence and Application*, pp. 121-123.

Huang, Y. G.; Sang, N.; Hao, Z. B.; Jiang, W. (2014): Eye tracking method for improving CamShift algorithm. *Application Research of Computers*, vol. 31, no. 4, pp. 1220-1224.

Li, J. F.; Li, S. G.; Tong, C.; Liu, Y. G. (2018): A geometry appearance base pupil detection method for near-infared head-mounted cameras. *IEEE Access*, vol. 6, pp. 23242-23252.

Li, X. H.; Gao, Y.; Qian, G. B. (2016): Monocular distance measuring system based on pupil positioning. *Intelligent Computer and Applications*, vol. 6, no. 2, pp. 70-73.

Li, X. H.; Tang, J. J. (2018): Research on human eye location based on combination of skin color matching and integral projection. *Automation Application*, no. 12, pp. 59-61.

Liu, L. T.; Dong, X. Y.; Liu, J.; Wang, X. R.; Huang, Z. Q. (2018): Research on eye tracking algorithm based on spatiotemporal context and random forest. *Chinese Journal of Liquid Crystals and Displays*, vol. 33, no. 5, pp. 443-449.

Mehrubeoglu, M.; Muddu, R. (2011): Real-time eye tracking using a smart camera. *Applied Imagery Pattern Recognition Workshop*.

Raudonis, V.; Simutis, R.; Narvydas, G. (2010): Discrete eye tracking for medical applications. *International Symposium on Applied Sciences in Biomedical and Communication Technologies*.

Tang, J. X.; Zhang, J. X. (2009): Eye tracking based on grey prediction. *International Workshop on Education Technology and Computer Science*, pp. 861-864.

Tian, Q.; Wang, Y. Z. (2018): Human eye location and state recognition based on image gray feature. *Computer Knowledge and Technology*, vol. 14, no. 3, pp. 170-172.

Wang, W. C.; Chang, L. F. (2011): Human eye precise positioning method based on area projection. *Journal of Optoelectronic Laser*, vol. 22, no. 4, pp. 618-622.

Wu, H.; Jin, L. Z. (2018): Human eye detector based on hybrid cascade classifier. *Industrial Control Computer*, vol. 31, no. 6, pp. 44-47.

Yan, B.; Wu, M. Y. (2018): Method for accurately positioning pupil center in low-resolution images. *Electronic Measurement Technology*, vol. 41, no. 16, pp. 74-78.

Yang, C. X.; Sun, J. D.; Liu, J.; Yang, X. H.; Wang, D. S. et al. (2010): A gray difference-based pre-processing for gaze tracking. *International Conference on Signal Processing*, pp. 1293-1296.

Zhang, C.; Chi, J. N.; Zhang, Z. H.; Wang, Z. L. (2010): A new sight tracking method based on pupil-corneal reflection technology. *Chinese Journal of Computers*, vol. 33, no. 7, pp. 1272-1285.

Zheng, W.; Wang, Y. Q. (2012): Design and implementation of human eye detection system based on DM642. *Modern Electronics Technique*, vol. 35, no. 4, pp. 105-108.