

An Easy Iris Center Detection Method for Eye Gaze Tracking System

Mingxin Yu

Intelligent Human-Machine Systems Lab, Northeastern University, Boston, USA
School of Life, Beijing Institute of Technology, Beijing, China

Yingzi Lin

Intelligent Human-Machine Systems
Lab, Northeastern University, USA

Xiaoying Tang

School of Life, Beijing Institute of
Technology, China

Jing Xu

Intelligent Human-Machine Systems
Lab, Northeastern University, USA

David Schmidt

Intelligent Human-Machine Systems
Lab, Northeastern University, USA

Xiangzhou Wang

School of Automation, Beijing
Institute of Technology, China

Yikang Guo

School of Automation, Beijing
Institute of Technology, China

Iris center detection accuracy has great impact on eye gaze tracking system performance. This paper proposes an easy and efficient iris center detection method based on modeling the geometric relationship between the detected rough iris center and the two corners of the eye. The method fully considers four states of iris within the eye region, i.e. center, left, right, and upper. The proposed active edge detection algorithm is utilized to extract iris edge points for ellipse fitting. In addition, this paper also presents a predicted edge point algorithm to solve the decrease in ellipse fitting accuracy, when part of the iris becomes hidden from rolling into a nasal or temporal eye corner. The evaluated result of the method on our eye database shows the global average accuracy of 94.3%. Compared with existing methods, our method achieves the highest iris center detection accuracy. Additionally, in order to test the performance of the proposed method in gaze tracking, this paper presents the results of gaze estimation achieved by our eye gaze tracking system.

Keywords: iris center detection, eye gaze tracking, active edge detection algorithm, predicted edge points algorithm

Introduction

Eye gaze tracking plays an important role in communication between human and machine (Ferhat, Vilarino, & Sanchez, 2014). Eye gaze tracking systems developed in recent decades have been used in a lot of areas, such as studies of driver behavior (Flores, Armingol, & Escalera, 2011), virtual reality (Duchowski, Shivashamkaraish, Rawls, Gramopadhye, Melloy, & Kanke, 2000), assistive devices for motor-disabled persons (Barea, Boquete, Mazo, & Lopez, 2002), human-robot interaction (Yu, Lin, Schmidt, Wang, & Wang, 2014; Yu, Wang, Lin, & Bai, 2014), human-machine collaborations (Cai & Lin, 2012),

reading and scene perception (Liversedge, Meadmore, Corck-Adelman, Shih, & Pollatsek, 2011), neurology (Tseng, Cameron, Pari, Reynolds, Munoz, & Itti, 2012), and clinical research (Papageorgiou, Hardiess, Mallot, & Schiefer, 2012). Gaze tracking systems consist of two types: intrusive and non-intrusive. Intrusive systems require physical contact with the user. Contact manners mainly include contact lenses (Robinson, 1963), electrodes (Kaufman, Bandopadhyay, & Shaviv, 1993), and head-mounted devices (Li, Winfield, & Parkhurst, 2005; Świrski, Bulling, & Dodgson, 2012). However, these contact manners are not very conformable for users. Non-intrusive systems are also known as remote gaze tracking systems. These systems do not anything attached to the user, so they are widely applied and researched by users.

In gaze tracking systems, tracker calibration establishes the relationship between gaze and objects users look at. The gaze is defined as the center of fovea projection into object space through the eye center (Oyster, 1999). Hence, the correct eye center position plays an important role in gaze calibration. To do this, eye trackers have adopted a 2-D mapping calibration method (Blignaut, 2013; Yu, Wang, Lin, & Bai, 2014).

Existing eye center detection methods can be briefly classified into two categories: pupil center detection (Beymer & Flickner, 2003; Li, Winfield, & Parkhurst, 2005; Świrski, Bulling, & Dodgson, 2012) and iris center detection. In general, pupil center detection is used in intrusive systems, though some non-intrusive gaze tracking systems also implement pupil center detection. The technique often depends on near infrared light (IR). Using IR light, light with a wavelength outside the visible spectrum, makes detecting the pupil easy while avoiding user distraction. However, the use of IR imaging techniques in outdoor scenarios during daytime is very restricted due to ambient IR illumination (Sigut & Sidha, 2011). During operation the pupil changes in size and wobbles during saccades. This variability causes issues with data quality (Kimmel, Mammo, Newsome, 2012; Nyström, Hooge, Holmqvist, 2013; Hooge, Nyström, Cornelissen, & Holmqvist, 2015). The iris center detection is widely used in non-intrusive systems. The method often works under visible light, which is not as sensitive to the IR light in the environment. With the above background, the usefulness of iris detection becomes much more evident.

In general, the iris region gray intensity is lower than that of the surrounding anatomy. Furthermore, the contrast on the edge between the sclera and iris is higher, so iris center detection can take advantage of this cue to easily determine iris center. Here, we present some existing iris center detection methods which have successfully worked for gaze tracking. Sigut and Sidha (2011) developed an eye gaze tracking system which adopted the iris center detection method called the ICCR (Iris Center Cornea Reflection). The bright spot on the eye created by a 5-W Halogen lamp was first detected as a base point for iris contour extraction, and the Canny edge detector was applied to the eye gray image in order to obtain the iris edge image in binary mode. Then a distance filter was used to eliminate edge points too close or too far from the base point. Lastly, a RANSAC algorithm was used to

extract the iris edge points for iris contour fitting. Wang, Sung, and Venkateswarlu (2005) adopted a threshold value to automatically segment the iris from the sclera on a binary image. The edges of the image were obtained by the Canny operator. Lastly, an edge following technique was used to find the longest vertical edges on the image for iris contour fitting using an ellipse fitting algorithm.

Mohammadi and Raie (2012) proposed a novel algorithm for iris center location. The Canny operator was first used to produce the human eye edge image with a fixed threshold value. Then, the split points were removed using limited change of slope in an ellipse. Lastly, a SVM classifier was used to select some of the segments as iris parts and merged them together for iris edge ellipse fitting. Zhang, Zhang, and Chang (2001) also utilized the Canny operator to create an edge image. Then a horizontal template edge operator was run to detect the two longest vertical edges of the iris. Lastly, a CMP-RANSAC algorithm was adopted in order to remove the noise edges and left edges for ellipse fitting. The method was more effective than morphological operator methods. Torricelli, Conforto, Schmid, and Alesio (2008) proposed a method based on Sobel operator for iris edge detection. Relative to the Canny operator, although it resulted to be more robust to light changes, a very high number of edges were detected within the eye image, making the discrimination of the correct iris edge very difficult. With the Sobel operator a lower number of edges were detected.

Sirohey, Rosenfeld, and Duric (2002) used a semicircular annulus template with one-third of the eye length as iris radius to detect the iris edge contour on the image. The most number of edge pixels contained in the annulus was regarded as the iris edge points. In addition, Perez, Lazcano, and Estevez (2007) proposed a similar method, which created generic templates to detect the iris.

However, eyelids always cover parts of the iris, making iris edge extraction difficult. Additionally, because the eyeball is an active structure, iris edge detection methods need to consider different states of the iris within the eye region, i.e., center, left, right, and upper, as shown in Figure 1. Here, we did not consider the down state, because the eyeball is hard to complete when the iris is underneath the lower eyelid. In general, iris center detection depends on the extraction of iris edge points precisely. As for upper state, although upper edges of iris are occluded by upper eyelid, lower left and right edges can still be kept well. The iris center position is deter-

mined by the contour elliptical fitting. Through experiment, when left and right edge points of iris are obtained, the corresponding ellipse can easily be obtained. However, if the human eyes gaze at objects on the left and right periphery of the screen, iris edges closer to eye corners would be hidden. In this case, it is hard to obtain the correct ellipse fitting with obtained points from one side of the iris edge. At the same time, through the above literature review, existing algorithms seldom consider the two cases. Hence, this paper presents an easy and efficient iris center detection method to solve this problem.

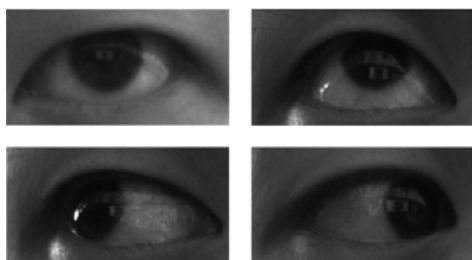


Figure 1: Four states of iris within the eye region.

Proposed Method

The procedure of proposed iris center detection method consists of two parts: features detection and iris edge detection, as shown in Figure 2.

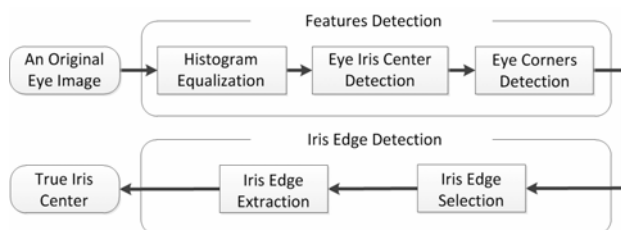


Figure 2: The procedure of iris center detection method.

The feature detection starts from an original eye image, and then three steps are performed for detecting rough iris center and two eye corners, respectively.

Firstly, histogram equalization is used for enhancing eye image contrast. Under visible light the gray intensity of eye images is darker, and necessary details can be hidden in the dark areas, as shown in Figure 3a. Hence, through pre-processing of histogram equalization, the dynamic range of image grey intensity should be large enough for iris edge detection, as shown in Figure 3b.

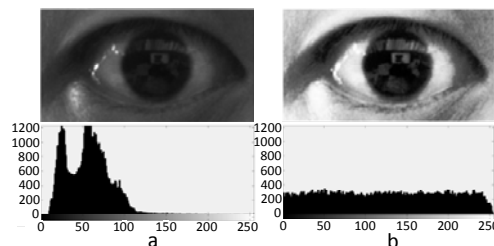


Figure 3: (a) shows the human eye image and gray histogram, (b) shows the human eye image with histogram equalization and gray histogram.

Secondly, we use a hybrid projection function (HPF) (Zhou & Geng, 2004) to estimate rough iris center of the eye. In general, the image projection function can be used to detect the boundary of different image regions. The most commonly used projection function is integral projection function (IPF). However, IPF cannot capture the variation of the image well. Then, a variance projection was proposed (Feng & Yuen, 2001), which is usually more sensitive to the variation in the image than IPF. In order to obtain higher accuracy in rough iris center, Zhou and Geng (2004) presented a new projection function, i.e. combining IPF and VPF, known as HPF. The performance of HPF in rough iris center detection indicated that combination IPF and VPF could be more powerful than sole IPF or VPF. Some examples of successful rough iris center detection are shown in Figure 4. Additionally, through the experiments, we found the offset was smaller between the true and the rough iris center. The results help to the selection of iris edges (See next section).

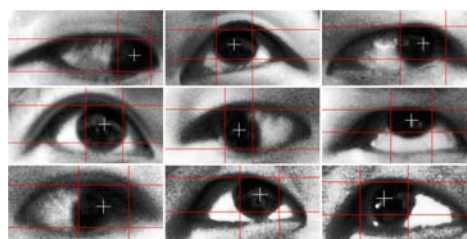


Figure 4: Results of successful rough iris center detection on eye database.

Thirdly, two search windows are created on eye images for detecting nasal and temporal eye corners using the method proposed by Torricelli, Conforto, Schmod, and Alesio (2008). For the nasal corner, the search window was created over the inner area of the eye. Within the window, the most lateral pixel of the binary image was considered as the estimated nasal corner. For the

temporal corner, the search window was created over the external area of the eye. Ten-level quantization was used to the image within the window. By eliminating the brighter levels, the external extremity of eye corner would be considered as the temporal eye corner. Some examples of successful eye corner detection are shown in Figure 5.

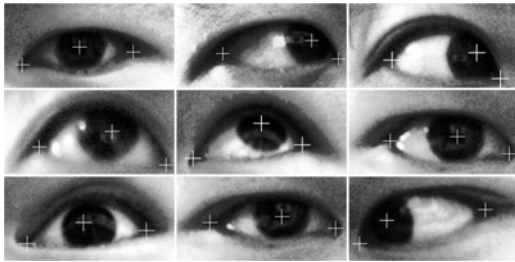


Figure 5: Results of successful eye corners detection on eye database.

For iris edge detection, the aim is to extract correct iris edge points for determining the true iris center. In the following sections, we will give the procedure of the proposed iris edge detection. All eye images used in our paper are extracted from our eye database (see *Eye Database Setup* section).

Selection of Iris Edges

According to features detection, after obtaining the rough iris center and two eye corners, we take advantage of the information to detect the iris edge.

As the *Introduce* section described, the eye is an active structure, so the iris can roll into the two eye corners when human eyes gaze at objects on the left and right periphery of the screen. In this case, the left or right edge will be hidden. This section first creates a model to determine which edges belonging to the iris should be detected.

The model is based on a distance ratio between the detected rough iris center and eye corners. In general, eye corners are stable features and often used as fixed points

relative to iris center for calculating eye gaze in some tracking systems (Zhu & Yang, 2002; Wang & Venkateswarlu, 2002; Wang, Sung, & Venkateswarlu, 2005). Hence, the distance between two eye corners are almost unchanged when eye corners are detected accurately.

Figure 6 shows three iris states within the eye region, i.e. at three different positions: left, center, and right. If the iris rolls into the nasal and temporal eye corners, the edges of the iris closer to eye corner will be hidden. In this case, the best detection method is to extract the apparent iris edge on the other side. Hence, we design an ideal model to estimate which iris edges need to be extracted. Here, the right eye is chosen as an example for model description. In Figure 6, the point P_r and point P_l represent nasal and temporal corners of the right eye, respectively. The length of Euclidean Distance D between P_r and P_l can be calculated by the following formulation:

$$D = \|P_l P_r\| = \sqrt{(x_l - x_r)^2 + (y_l - y_r)^2} \quad (1)$$

The point P_c represents the rough iris center. Here, we assume P_c is an ideal iris center, i.e. true iris center, in order to establish the modeling. The d_r is used to represent the length of distance between the cross point, which is P_c perpendicular to $P_r P_l$ and the point P_r . The d_l is used to represent the length of distance between the cross point, which is P_c perpendicular to $P_l P_r$ and the point P_l . $\angle P_c P_r P_l$ is represented as α and $\angle P_c P_l P_r$ is represented as β . According to the geometries shown in Figure 6, d_r and d_l can be achieved by the formulations:

$$d_r = \cos \alpha \cdot \|P_c P_r\| \quad (2)$$

$$d_l = \cos \beta \cdot \|P_c P_l\| \quad (3)$$

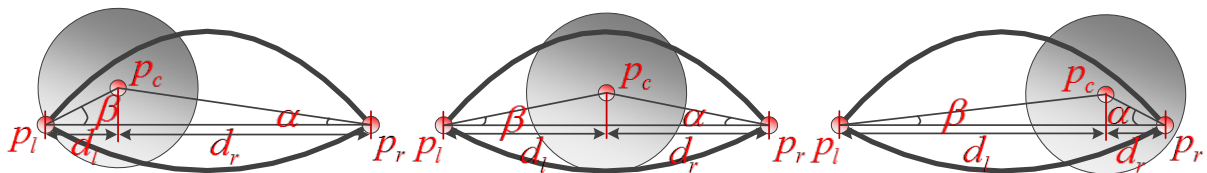


Figure 6: Three states of iris within eye region and geometrical relationship between the eye corners and the ideal iris centre.

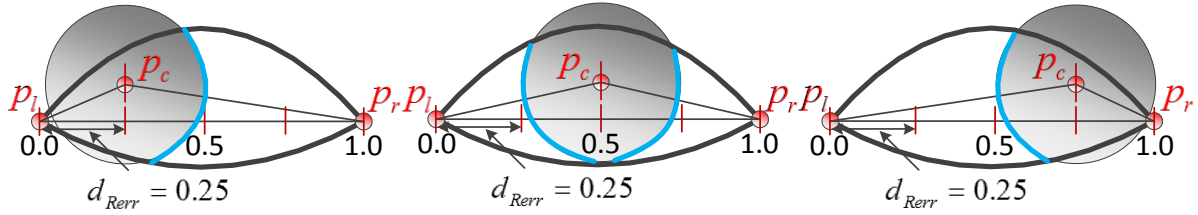


Figure 7: Iris edge marked with blue arc lines need to be detected according to the geometrical relationship of the position between the eye corners and the ideal iris centre.

According to distances d_r and d_l , the distance ratio R_t can be defined as follows:

$$R_t = \frac{d_l}{D} = \frac{D - d_r}{D} \quad (4)$$

In Figure 7, the distance between two eye corners can be equally divided into four segments. The length of the distance of each segment can be expressed as a relative constant, $d_{Rcons} = 0.25$. Because the offset between the rough and the true iris center is smaller, we can adopt the relative constant to set threshold values of right and left iris edges, i.e. T_{er} and T_{el} , in order to determine which edges should be extracted from the iris region, as follows:

$$T_{el} = d_{Rcons} \quad (5)$$

$$T_{er} = 3 \cdot d_{Rcons} \quad (6)$$

According to threshold values, we give the decision criterion for selection of iris edge. The S_{Edge} represents which edges belonged to iris that needs to be detected. It can be represented as follows:

$$S_{Edge} = \begin{cases} S_{RE} & \text{if } R_t < T_{el} \\ S_{RLE} & \text{if } T_{el} < R_t < T_{er} \\ S_{LE} & \text{if } T_{er} < R_t \end{cases} \quad (7)$$

where S_{RE} represents the right edge of the iris, S_{LE} represents the left edge of the iris, and S_{RLE} represents both edges (right and left) of the iris. The S_{Edge} is marked with blue lines in Figure 7. In the next subsection, we will complete the detection algorithm to obtain S_{Edge} .

Extraction of Iris Edges

The edge of the iris can be split into four parts by two lines between two eye corners and the rough iris center,

as shown in Figure 8b. Here, the left edge of the iris is taken as an example to describe the algorithm. We need to first determine search angles. The upper and lower limbus of the iris are usually occluded by eyelids. The intersection of the iris and eyelids creates the angles ϕ and φ with respect to horizontal, as shown in Figure 8a. Daugman (1993) proposed the angular arc of contour was restricted in search range by two opposing 90° cones, i.e. $\phi + \varphi = 90^\circ$. In (Sankowski, Grabowski, Napieralska, Zubert, & Napieralski, 2010), the search range was increased slightly into $\phi = 45^\circ$ and $\varphi = 60^\circ$. According to examining the frontal faces from the AR Face database, Torricelli, Conforto, Schmid, and Alesio (2008) found the average value for angles ϕ and φ were 50° and 70° , respectively. Through analysis of the eye structure in our eye database, we found the maximum angles were $\phi = 80^\circ$ and $\varphi = 85^\circ$, respectively. Hence, the search angles λ and γ can be defined as the following (The left arc of iris is used as an example):

$$\gamma = \varphi - \beta \quad (8)$$

$$\lambda = \phi + \varphi - \gamma \quad (9)$$

Here, the search arc marked with the blue arrow labeled 1 in Figure 8b is used as an example for the description of the iris edge detection algorithm. Other search arcs are similar to this. The proposed detection method is similar to the method in (Zhang, Zhang, & Chang, 2001), which based on 1D line search along normal vector at each point of the contour, but there are three different aspects. Firstly, the detection method in (Zhang, Zhang, & Chang, 2001) was run on the binary image while our method dealt with the gray image. Secondly, the initial search line radius was not predefined in our search. Thirdly, a simple template was not used for edge detection in our research.

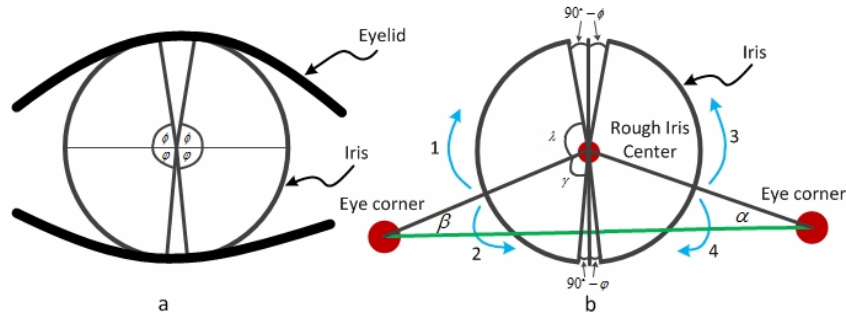


Figure 8: (a) shows angles ϕ and ϕ considered for the iris edge detection ranges free from eyelids occlusion. (b) shows directions of iris edge detection marked with four blue arrows labeled 1, 2, 3 and 4. λ and γ represent the search angles between initial search line (See Figure 10b) and maximum search boundaries.

The proposed 1D edge search method can be depicted in Figure 10b. The contour of the iris consists of a number of points which are placed at the same degree interval $\Delta\theta$. The line between eye corner and rough iris center is designated as the initial search line. Firstly, the initial point marked with the black circular point in Figure 10b is detected along the initial search line using the line integral ratio.

Before detecting the initial point, we need to use a smooth filter with size 7×7 along the search line to smooth the peripheral region corresponding to each pixel along the initial search line, as shown in Figure 10a. The reason for that is there are many highlights produced by natural (visible) lights on the iris. Usually, the size of some highlights is larger, so we need to eliminate those in order to avoid inference when detecting the initial point. The smoothed filter takes advantage of more pixels (within a smoothed window) to eliminate the fluctuations of pixels and complete the noise cancellation. Certainly, we also can directly operate on the whole eye image with smoothed filter. However, experimentation showed that this increased computational cost.

Then, we will detect the initial point P_{init} along smoothed initial search line. The line integral is calculated within L1 marked by a green rectangle and L2 marked by red rectangle ranges, as shown in Figure 10c. The ratio K between two line integrals in discrete form is shown as follows:

$$K = \frac{\sum_{(x,y) \in L1} I(x,y)}{\sum_{(x,y) \in L2} I(x,y)} \quad (10)$$

Figure 9 shows that the location of the initial point is located at the maximum ratio. Although a difference could also be used in initial point detection on the initial search line, we found that this may achieve a worse result. Firstly, sometimes within eye corner regions gray intensity is lower than that of the iris. Hence, we might obtain a bigger difference within the eye corner region than on the iris. Secondly, the highlights with larger size could not be eliminated well with a smooth filter. In this case, we could obtain two edge points with dark-to-light, one produced by the highlight, and another produced by the actual edge point. Thus, those would impact the result of initial point detection. As for line integrals, it computes pixel sum within a certain range on the initial search line, through experiment, this method avoided the above cases effectively.

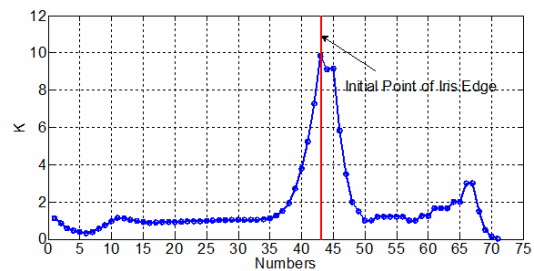


Figure 9: The ratio of line integral along the initial search line.

Next, a search range, which is marked with two blue search arcs in Figure 10b, is built according to the initial point. The length of the line between initial point P_{init} and rough iris center P_c is used as initial search radius r_{init} . Hence, the initial search range R_s is defined as $[r_{init} - \delta, r_{init} + \delta]$.

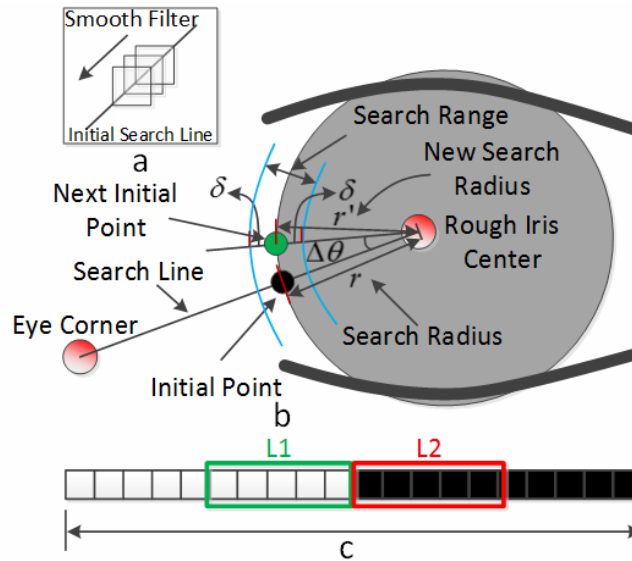


Figure 10: (a) shows a smooth filter moving along the initial search line for removing highlights on iris region. (b) shows the schematic diagram of the detection method of iris edge. (c) shows two sliding windows marked with red and green rectangles for the calculation of the line integral ratio between them to determine the initial point.

Within the search range, the next edge point will be detected using a gradient edge detection algorithm. It is used as a new initial point, which is marked with a green circular point in Figure 10b. Lastly, the new search radius r' can be achieved in terms of new edge point. In other words, the edge location of the previously detected point is utilized to determine the initial position of next point and the search range (also called active search range) is determined by previous detected point.

Through the iris edge extraction method depicted above, we could achieve iris edge points, as shown in Figure 11a. However, it failed in certain regions. Many error points, i.e. noise points marked with red elliptic regions, were detected, since the grey intensity of parts of eyelid and eyelash are similar to those of iris, and some bright reflection spots are created by visible lights on the iris.

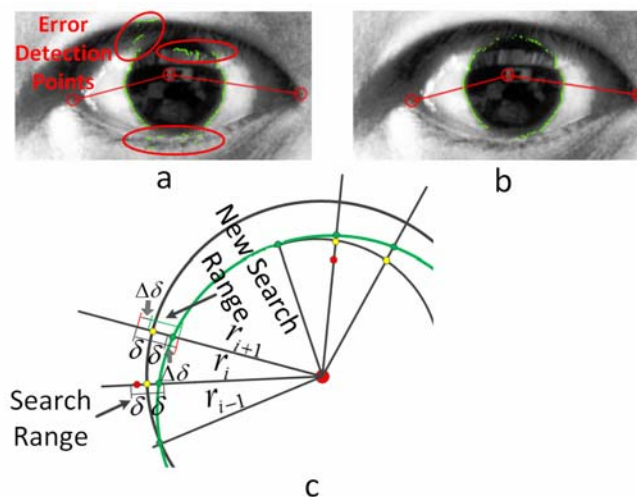


Figure 11: (a) shows detection result of edge points of iris. (b) shows detection result of edge points with the predicted edge points algorithm. (c) shows schematic diagram for Iris Edge Detection Algorithm1.

In order to solve this problem, we proposed an active edge detection algorithm. In Figure 11c, the iris edge points (Marked with green points) should be on the real iris edge corresponding to the green arc line. The red points are assumed as noise points. The yellow points are obtained according to the former radius in terms of the depicted method above.

The algorithm iterates through all the edge points. We first need to set a threshold value T_{err} as a cease condition for the algorithm, where T_{err} is the number of error edge points happening continually. Because noisy points are either inside or outside the iris edge, a *Threshold* is set to compare with the distance between current and former radius, i.e. $|r_{i+1} - r_i|$. Additionally, according to the positive and negative values for $r_{i+1} - r_i$, we can determine the moving direction of the search arc. If the length is larger than *Threshold* and $r_{i+1} - r_i$ is positive, the search range R_s would move little distance $\Delta\delta$ toward the rough iris. If $r_{i+1} - r_i$ is negative, the search range R_s would move little distance $\Delta\delta$ in the opposite direction with respect to rough iris center. If the length is smaller than *Threshold*, the search range R_s is not changed. Here, we take the left upper edge labeled with 1 in Figure 8b as an example to depict active edge detection algorithm. The Pseudocode of the algorithm is presented in *Appendix A*. The result of the iris edges extraction is shown in Figure 11b, the noise points are removed by Algorithm1 well.

True Iris Center Detection

Once the iris edge points were detected, the iris center can be found. A common approach is to use the Hough transform to fit a circle to the detected points (Dobes, Martinek, Skoupil, Dobesova, & Pospisil, 2006; Matsumoto & Zelinsky, 2000). However, the projection of the iris on the image will always be an ellipse, except when the eye is pointed directly at the camera. In our research, the extracted contour points are further refined using a direct least square ellipse fitting algorithm (Fitzgibbon, Pilu, & Fisher, 1999). Additionally, the number of iris edge points extracted by Algorithm1 is less than 6 pixel points, making the ellipse fitting fail. In this case, the subpixel edge detection method (Zhu & Yang, 2002) is essential for correctness of fitting. An example of ellipse fitting for iris edge is shown in Figure 12.

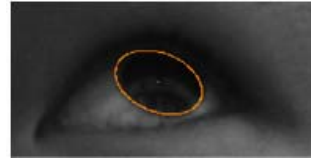


Figure 12: An Example of ellipse fitting for iris edge.

However, we found that the ellipse fitting algorithm could not always achieve a good result of finding the iris center on the condition of the iris rolled into nasal and temporal eye corners through the experiment. We assume that the edge of ellipse shape consists of two sides split by the minor axis. If more edge points on the two sides are detected and their distribution is uniform, the shape of ellipse fitting is tending towards correct, such as the states of iris on the center and upper regions of the eye. For another two cases, edge points on one side of ellipse are only used for fitting, although edge points are perfectly detected by the Algorithm1 as shown in Figure 13a. Figure 13b shows the performance of ellipse fitting is not ideal. This case can be justified by the fact that no edge points tend to the upper and lower vertex of true ellipse and distribution of edge points detected are not uniform. In order to achieve high accuracy of ellipse fitting, we proposed an algorithm of predicted edge points.

Here, we take the iris' right upper edge as an example to describe this algorithm, as shown in Figure 14. Firstly, the last edge point P_{lst} is taken from detected edge points array. Then, the Euclidean Distance r_{lst} between P_{lst} and rough iris center position P_c , i.e. $r_{lst} = \|P_{lst} - P_c\|$, is computed. Thirdly, compared initial radius distance r_{init} to r_{lst} , i.e. $r_{lst} - r_{init}$. If the result is positive, the predicted edge points tend towards the direction of the rough iris center. Or else, the direction is toward to the opposite direction with respect to rough iris center. Lastly, predicted edge points P_e will be achieved. The Pseudocode of the algorithm is presented in *Appendix B*.

Figure 13c shows the detection result with the Algorithm2, the red points (Marked within red elliptic regions) represent predicted iris edge points. The detected and predicted points are used for ellipse fitting. The last result is shown in Figure 13d. We can clearly observe that the ellipse fitting performance achieved by Algorithm1 and

Algorithm2 is better than the performance of ellipse fitting only with detected edge points.

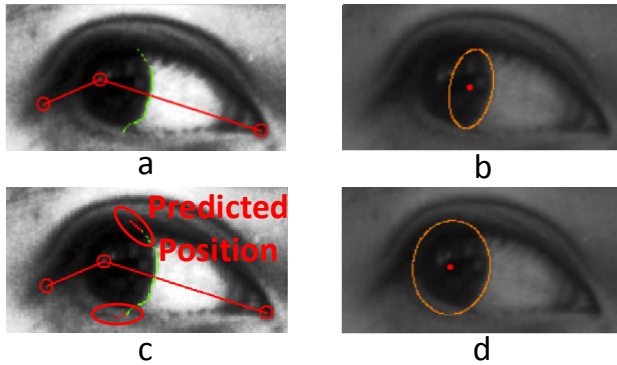


Figure 13: (a) and (b) show the edge points with Algorithm1 and are used for ellipse fitting. (c) shows the predicted edge points with Algorithm2. (d) shows the final result of ellipse fitting with detected and predicted edge points.

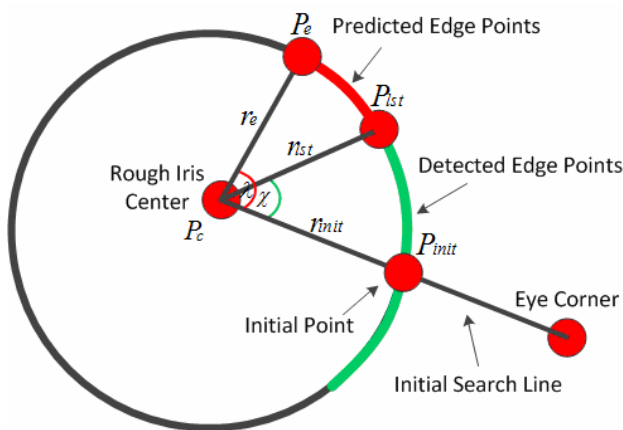


Figure 14: The schematic diagram for Predicted Edge Points Detection Algorithm2.

Experimental Results

The evaluation of our method is carried out on our eye database. The evaluation criterion of iris center detection is given first. Then, the iris center detection results are achieved using the proposed algorithm in this paper. Lastly, comparison of iris center detection results with the existing methods is presented.

Eye Database Setup

Twenty subjects from different regions of China, such as Beijing, Jiangsu, Henan, Shanxi, Inner Mongolia, etc.,

ten female and ten male, aged from 23-31 years, took part in the experiment. All had normal (without glasses) vision. Each subject was asked to sit in front of the computer screen, the distance between subjects and screen is 60cm.

Then, we captured the subject's face image using our developed software. During face image capture, we asked subjects to move their head slightly in order to make different facial poses. At the same time, we controlled the visible lights open or close, making different illumination environment. It is noting that we did not crop subjects' face images from recording video, but cropped images from real-time video stream. In other words, subjects were asked to complete several eye motions, such as look at center, left, right, and up, in the processing we pressed button to record a face image frame from the real-time stream. Each subject contributed to more than 400 face images. After image acquisition, we used the rectangle region with size 240×120 pixels, which could cover the eye image well, to crop 4800 eye images on the faces by manual. The iris state images within eye region include 1200 center state, 1200 left state, 1200 right state, and 1200 upper state, respectively. The image acquisition system can be seen in *Gaze Tracking Test* Section.

Measurement

In order to evaluate true iris center detection accuracy, we proposed an evaluation criterion by modifying a relative error measure proposed by Jesorsky, Kirchberg, and Frischolz (2001).

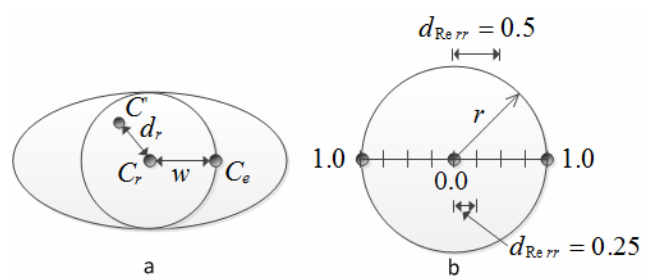


Figure 15: The model of one iris on the eye.

Firstly, the iris center position is extracted manually as the expected iris center, denoted as C_r . Secondly, the iris center of eye is estimated by proposed algorithm is denoted as C' . Thirdly, an iris edge point extracted manually is denoted as C_e . Those positions are depicted

in Figure 15a. Lastly, the relative error $d_{Re rr}$ is defined as:

$$d_{Re rr} = \frac{d_r}{\|w\|} \quad (11)$$

where d_r is the distance between the expected iris center and the corresponding estimated iris center. The Euclidean distance $\|C_e - C_r\|$ is defined as $\|w\|$.

The threshold value T is defined for determining detection correctness. In Figure 15b, the line between true iris center and an iris edge point is divided into four segments, each segment is 0.25. If the $d_{Re rr}$ is less than T ($d_{Re rr} < T$), the iris center detection is considered to be correct. When $d_{Re rr} = 1$, d_r might reach the distance of half the width of one iris from the expected eye center position to an edge point of iris, namely the circle with a radius of $r = 1$, as shown in Figure 15b. Here, we could not easily point out which relative thresholds T were defined as a correct threshold, but we considered the closer between estimated iris center and truth iris center,

the higher correct detection rate. In this paper, the true iris center of eye is considered as the region with a radius of $r = 0.15$, namely the threshold value T is less than 0.15.

Evaluation of Iris Center Detection

This section shows quantitatively the accuracy of our proposed method for different T corresponding to the four states of the iris, i.e. center, left, right and upper, within eye region.

Firstly, the iris often stays within the center region of the eye. In this case, the left and right edges of the iris are clearly visible. Thus, we observed that the proposed method achieved highest accuracy in four states, as shown in Figure 16a. The accuracy of the proposed method is up to 99% when $T = 0.15$.

Secondly, the right edge of iris would be hidden (nasal or temporal corner corresponding to left or right eye), when the iris rolls into the right corner of the eye. Thanks to the algorithm of iris predicted edge points, our proposed method with $T = 0.05$ can be up to 74.21%, as shown in Figure 16c.

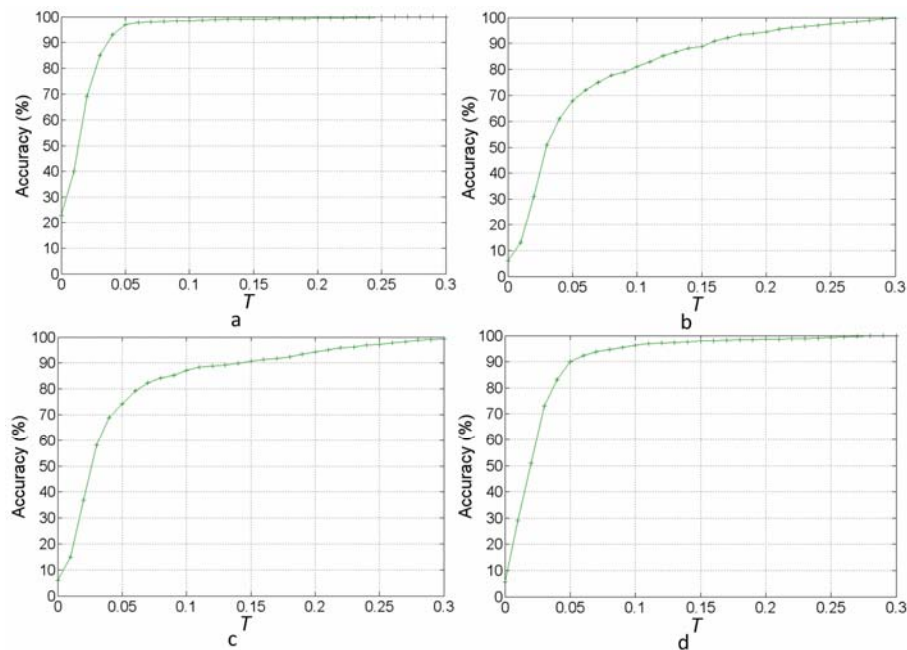


Figure 16: The distribution functions of relative error against accuracy of our proposed method with respect to true iris center. (a) Center, (b) Left, (c) Right, (d) Upper.

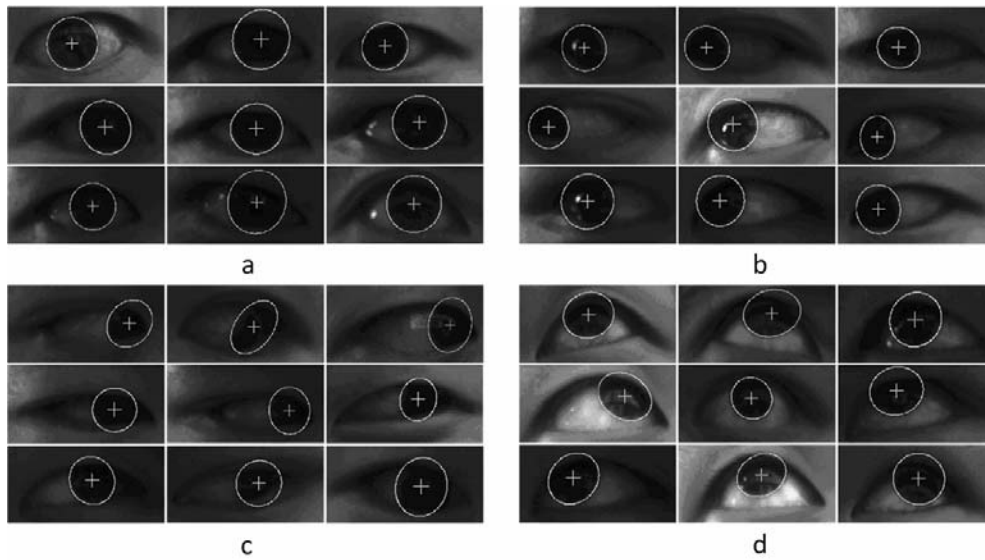


Figure 17: Successful examples of iris center detection corresponding to four states: (a) Center, (b) Left, (c) Right, (d) Upper.

Thirdly, when the eye looked in the upward direction, the upper edge of iris would be hidden under the upper eyelid of eye. In general, this case is easier to handle than the second case, because enough points on the lower edge of both sides of the iris can still be obtained for ellipse

fitting. Figure 16d shows the accuracy result is better than the second case. Especially, the accuracy is just 6.9% lower than the first case when $T=0.05$

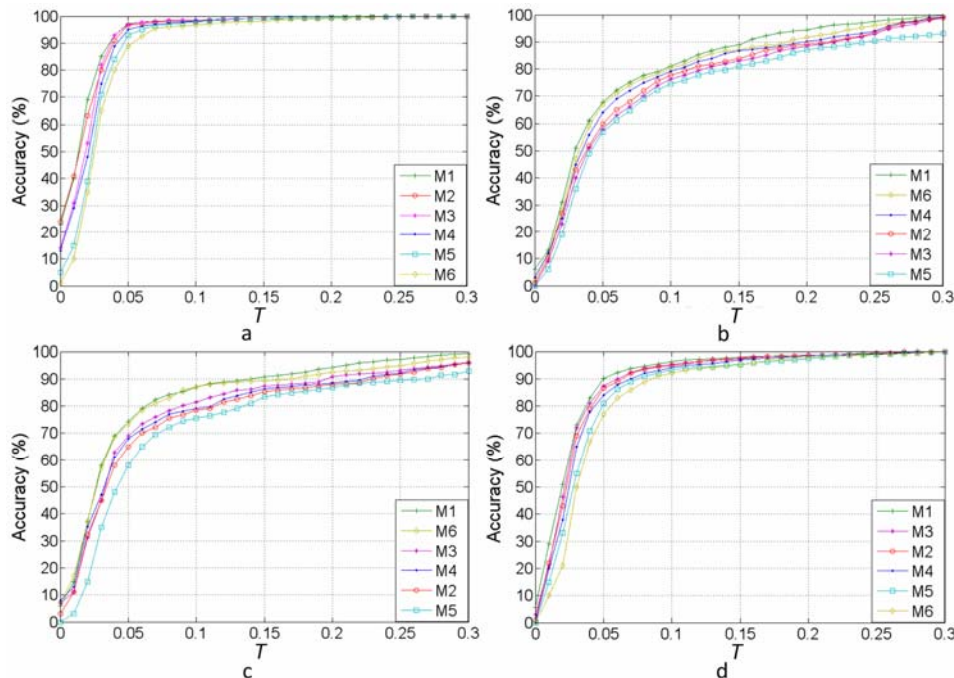


Figure 18: The distribution functions of relative error against accuracy of six methods (M1, M2, M3, M4, M5, and M6) with respect to true iris center. (a) Center, (b) Left, (c) Right, (d) Upper.

Lastly, when the iris rolled into the left corner of the eye, the left edge of the iris would be hidden. This case is similar to the second one. Its accuracy is little lower than in the second case, as shown in Figure 16b. According to our analysis, individual differences on eye images, such as different illuminations and different eye sizes existing in eye pictures, made this result.

Figure 17 shows some successful examples of iris center detection corresponding to the four iris states within eye regions.

Comparison With Other Methods

The method has been compared with other existing methods that were discussed in the *Introduction* section. Those picking methods, i.e. (Zhang, Zhang, & Chang, 2001) (M2), (Wang, Sung, & Venkateswarlu, 2005) (M3), (Torricelli, Conforto, Schmid, & Alesio, 2008) (M4), (Sigut & Sidha, 2011) (M5), and (Perez, Lazcano, & Estevez, 2007) (M6), were successfully used for iris center detection in the eye tracking system. All methods run on the images from our eye database and adopt the proposed measurement to estimate accuracy of iris center detection. In Figure 18a to Figure 18d, it is clear that the performance of our proposed method (M1) achieved the highest accuracy compared with other methods.

As for methods M2, M3, and M5, the Canny operator was used to detect the edge of the iris. M5 used the reflection point (Glint) as a reference point to create a distance filter to eliminate unwanted pixels on edge image of eye. However, our method did not adopt an auxiliary light source. Hence, in our research, the nasal corner was taken as a reference point to replace the reflection point. Methods M4 and M5 took advantage of horizontal template operators and edge following technique as the manners of iris edge detection. However, when the iris rolled into two eye corners, the performance of M4 and M5 would weaken significantly. As shown in Figure 18b and Figure 18c, the accuracies are lower than others with $T = 0.05$.

For the center state, because iris edges on both sides are apparent, high accuracies are obtained by all methods. It also proved that enough detected edge points and distribution uniformly on two sides of the iris could enhance the accuracy of fitting. But we found that method M6 has low accuracy with $T = 0.05$, as shown in Figure 18a and Figure 18d, the reason is possibly

inappropriate parameter selection of face size through the experiment analysis.

The average accuracies of the four states are given in Table 1, where our proposed method achieves highest accuracies of 84.12%, 91.1%, and 94.3% versus other methods when $T \leq 0.05$, $T \leq 0.1$ and $T \leq 0.15$. For comparative methods selected by our research, the accuracy of method M5 is 6.27% lower than our method for $T \leq 0.05$. M6 achieves highest accuracies of 89% and 92.48% corresponding to $T \leq 0.1$ and $T \leq 0.15$ except our method, respectively. It's worth nothing that the detection accuracy of Canny operator is lower than method M6. The reason is that Sobel operator made less noise edges than Canny operator when processing eye images. Hence, the accuracy of ellipse fitting achieved by Sobel operator is higher than by Canny operator. The same conclusion was also presented by Perez, Lazcano, & Estevez (2007).

Table 1

Comparison accuracy versus relative error.

Method	Accuracy ($T \leq 0.05$)	Accuracy ($T \leq 0.1$)	Accuracy ($T \leq 0.15$)
M1	84.12%	91.10%	94.30%
M2	77.05%	87.35%	91.34%
M3	77.85%	87.93%	91.98%
M4	77.75%	87.63%	92.30%
M5	72.25%	85.35%	89.73%
M6	76.63%	89.00%	92.48%

Figure 19 shows the distribution of relative errors for all methods, i.e. the histogram of relative error value d_{err} , as they were defined in (11). The range of each value has been quantized to 1200 bins. Table 2 gives the summary statistics of the mean and standard deviation of the relative error corresponding to the four states of iris within eye region. The average value of our proposed method for four states is 0.043 ± 0.004 , that is, the mean maximum error d_r for the iris center is only 4.3% of the actual iris center and an edge point. Compared with other existing methods, the proposed method achieves minimum relative error. Especially, as for left and right states of the iris, the mean values are 0.062 and 0.055, i.e. the mean relative errors are 6.2% and 5.5% of the actual iris center and an edge point, respectively. The results show the proposed algorithms can deal with the left and right states of the iris well.

Table 2

The summary statistics of mean and standard deviation of relative error for iris center localization.

Method	Center	Left	Right	Upper	Mean
M1	0.021 ± 0.001	0.062 ± 0.006	0.055 ± 0.006	0.032 ± 0.001	0.043 ± 0.004
M2	0.023 ± 0.001	0.085 ± 0.019	0.111 ± 0.026	0.037 ± 0.002	0.064 ± 0.012
M3	0.026 ± 0.001	0.101 ± 0.001	0.080 ± 0.014	0.035 ± 0.002	0.061 ± 0.005
M4	0.028 ± 0.001	0.077 ± 0.011	0.095 ± 0.020	0.040 ± 0.002	0.060 ± 0.009
M5	0.033 ± 0.001	0.105 ± 0.021	0.133 ± 0.033	0.044 ± 0.003	0.079 ± 0.015
M6	0.038 ± 0.002	0.067 ± 0.007	0.064 ± 0.010	0.049 ± 0.003	0.055 ± 0.006

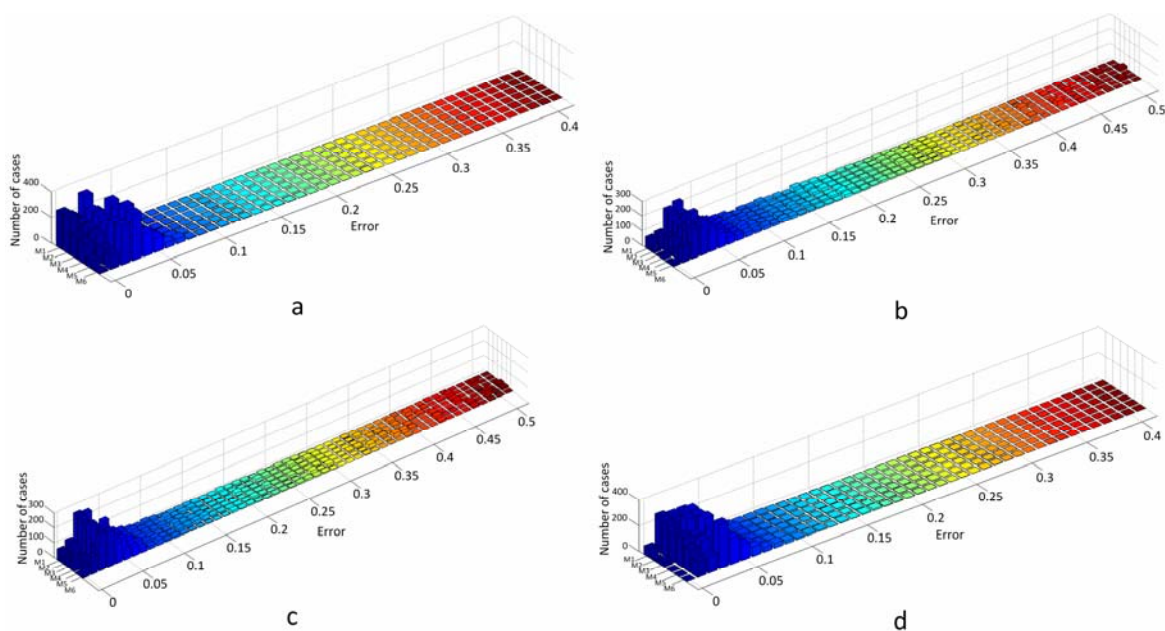


Figure 19: The distribution of error for iris center localization with six methods (M1, M2, M3, M4, M5, and M6). (a) Center, (b) Left, (c) Right, (d) Upper.

Gaze Tracking Test

In order to test the performance of the proposed iris center detection method, we use it in our eye gaze tracking system. Firstly, the setup of gaze tracking system and experimental procedure are described. Then, we compare the gaze estimation results obtained by our method with methods M2~M6 presented in last section.

System Description

The gaze system adopts a Gigabit Ethernet camera produced by the German Basler corporation. The type of the camera is scA1390-17gc with a resolution of $1390 \times$

1038 pixels, and it can capture 17 images per second. The lens is a product of the Japan Computer company with C-mount interface installed and a $2/3''$ interlaced CCD imaging sensor. The focal length is 16mm.

The system software consists of two parts: image processing and gaze estimation. The image processing contains eye and iris center detection. The template matching method is used for locating eye regions (Yu, Wang, Lin, & Bai, 2014). The gaze estimation is used to build the mapping relationship between eye feature information and gaze regard points. The system software was written using the NI Labview 2011 and Labview vision development toolkit 2011.

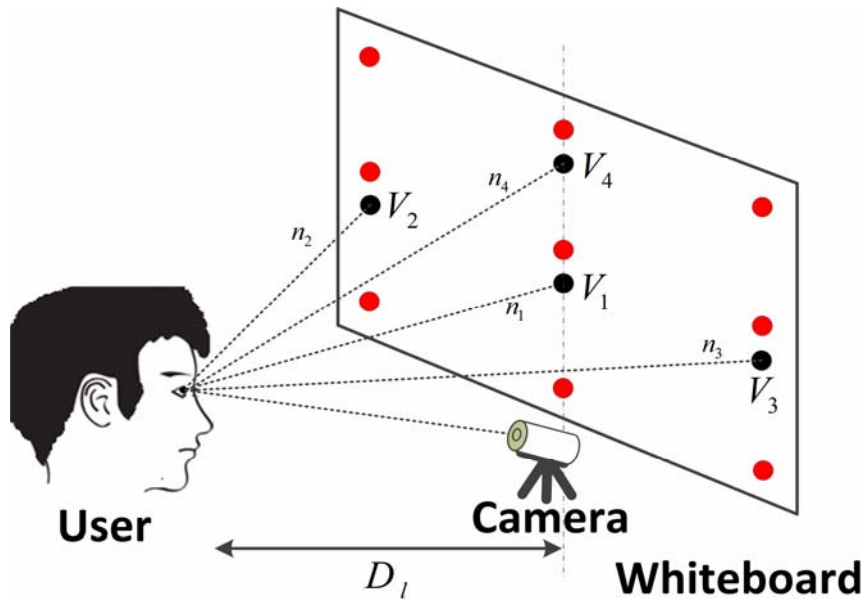


Figure 20: Experimental setup.

Experimental Procedure

The experimental setup is shown in Figure 20. The size of whiteboard is 100cm (horizontal) \times 60cm (vertical). Nine red points are represented as gaze calibration points. Four black points located at center, left, right and upper positions (V_1 , V_2 , V_3 , and V_4) on the whiteboard are defined as test points (target points). The space coordinates of all points with respect to the whiteboard are known. The camera is placed between the user and the whiteboard. The distance D_1 between the subject and the whiteboard is 60cm. The four lines of sight, n_1 , n_2 , n_3 , and n_4 , correspond to the target points, V_1 , V_2 , V_3 , and V_4 .

In our research, because no auxiliary light source was to produce a glint (reference point) on the iris, the nasal corner was taken as a reference point to replace it. Ten subjects from different regions of China (Different subjects in *Eye Database Setup* subsection), four female and six male, aged from 21-32 years, took part in the experiment. All had normal (without glasses) vision. The experiment was completed in our laboratory.

Before the start of each trail, a calibration procedure was required as follows. Subjects were asked to fixate on each calibration point and corresponding iris center and nasal eye corner coordinates were recorded, allowing the

calibration algorithm to calculate the points of gaze on the screen. Here, we used a second-order polynomial function for gaze estimation, as follows:

$$\begin{aligned} s_x &= a_0 + a_1v_x + a_2v_y + a_3v_xv_y + a_4v_x^2 + a_5v_y^2 \\ s_y &= b_0 + b_1v_x + b_2v_y + b_3v_xv_y + b_4v_x^2 + b_5v_y^2 \end{aligned} \quad (12)$$

where (s_x, s_y) is screen coordinates, (v_x, v_y) is the vector between the nasal eye corner and the iris center. The $a_0 \sim a_5$ and $b_0 \sim b_5$ are the unknowns.

However, the calibration method is very sensitive to head motion. Thus, the subjects were asked to keep his or her head still (no head motion) relative to the camera in order to achieve good performance when subjects gazed at each point on the whiteboard. At the same time, the position of nasal eye corner can also remain stable nearly.

Gaze Estimation

In the following, the accuracy has been calculated in terms of mean and standard deviation of the gaze error e_g between the true observed and the estimated positions. It is commonly expressed in angular degrees A_{lg} according to the following equation:

$$A_{lg} = \arctan\left(\frac{e_g}{D_1}\right) \quad (13)$$



Figure 21: Estimated eye gaze for the four target points: V_1 , V_2 , V_3 , and V_4 . (a) Subject #3, (b) Subject #7.

The results of the gaze estimation of the subjects #3 and #7 are shown in Figure 21. Table 3 gives the average accuracy of gaze estimation of 10 subjects corresponding to four target points, respectively. The global mean accuracy is approximately 0.99° in horizontal direction and 1.33° in vertical direction with a standard deviation 0.23° and 0.33° , respectively. We found that the accuracy in horizontal direction is higher than in vertical direction. The fact that part of the limbus is occluded by the eyelids results in a decrease in accuracy in vertical direction. It is also fair to remark that the gaze accuracies on right and left target points, i.e. V_2 and V_3 , show a significant decrease. As for the two cases, the iris happen left and right states within the eye region. The average accuracies of 1.49° and 1.70° are achieved in horizontal and vertical directions, respectively.

Table 3

Average accuracy in horizontal and vertical directions.

Target	Horizontal Accuracy (Degrees)	Vertical Accuracy (Degrees)
V1	0.43 ± 0.14	0.79 ± 0.15
V2	1.32 ± 0.36	1.70 ± 0.40
V3	1.29 ± 0.31	1.49 ± 0.47
V4	0.93 ± 0.12	1.34 ± 0.29
Mean	0.99 ± 0.23	1.33 ± 0.33

Additionally, we compared the performance of our method with methods M2~M6 presented in last section.

Those iris detection methods were used in our eye gaze tracking system. Figure 22 shows our proposed method achieves better performance compared with other methods. Especially, the accuracies of gaze estimation on left and right target points achieved by method M1 are significantly higher than others, it also evident that the algorithm of predicted edge points of iris has a certain effect on enhancing the accuracy of gaze tracking. Table 4 shows the global accuracies of all methods in horizontal and vertical directions. As for method M6, it also achieves higher accuracy of gaze tracking. On the one hand, left and right states of iris within eye regions achieve higher accuracies of gaze tracking, as shown in Figure 22. On the other hand, it is likely to achieve better accuracies of true iris center. (See Experimental Results section).

Table 4

Compared globe average accuracy in horizontal and vertical directions obtained by our method with M2~M6.

Method	Horizontal Accuracy (Degrees)	Vertical Accuracy (Degrees)
M1	0.99 ± 0.23	1.33 ± 0.33
M2	1.28 ± 0.24	1.50 ± 0.31
M3	1.29 ± 0.30	1.52 ± 0.36
M4	1.30 ± 0.34	1.55 ± 0.39
M5	1.44 ± 0.43	1.73 ± 0.54
M6	1.10 ± 0.23	1.45 ± 0.40

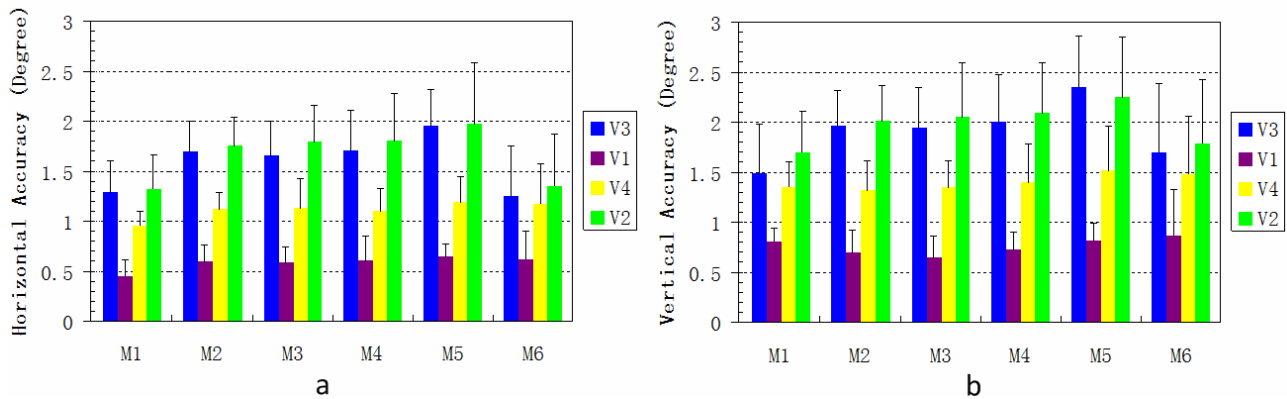


Figure 22: Histograms of mean and standard deviation of the error in degrees for methods M1-M6 corresponding to the four target points: V_1 , V_2 , V_3 , and V_4 . (a) Horizontal Accuracy, (b) Vertical Accuracy.

Discussion

Existing eye center detection methods used in eye trackers have two categories: pupil center detection and iris center detection. The pupil center detection method generally depends on near infrared light (IR). Because the pupil is much more apparent and easily tracked under IR light, and IR is not visible, the light does not distract the user when shone upon. However, the use of IR imaging techniques in outdoor scenarios during daytime is very restricted due to ambient IR illumination. Hence, it gives some limitations in certain applied fields.

Furthermore, in order to improve the performance of the pupil extraction task, a technique which is called the bright- and dark-pupil effect is used in the eye tracker. The effect produces a high-contrast image of the pupil. The bright pupil is created by the on-axis light sources and the dark pupil is created by the off-axis light sources. The on- and off-axis are relative to the camera axis. Then the bright- and dark-pupil images are produced by a light controller which controls the light on or off, and the alternate frequency is the same as the image frame frequency of the video camera. The image differencing technique is used for the pupil extraction. The technique is that a difference image is calculated from the alternating bright and dark pupil images, and the high-contrast pupil image is left by removing the most same background (Morimoto, Koons, Amir, & Flickner, 2000).

However, the technique has two disadvantages. One is the artifact image. The artifact image is mainly produced

by two reasons. Firstly, the image differencing technique with on- and off-axis lights source produces artifact images, which remove a portion of the pupil and corrupt the identified contour between the iris and pupil. Secondly, interframe motion of gaze tracking images also produces artifact images, which is created by misaligning the bright and dark pupil images. It distorts the extracted pupil contour. The detailed knowledge about the artifacts images can be found in (Hennessey, Nouredin, & Lawrence, 2008). Another is the additional hardware device for the bright- and dark-pupil effect. It makes the eye gaze tracking system more complicated during the setup and the building cost higher. Also, it is hard to build an eye tracker for some researchers with less knowledge about the hardware.

Additionally, the pupil changes in size and wobbles during the saccades. This variability can cause issues with data quality (Drewes, Masson, & Montagnini, 2012; Drewes, Montagnini, & Masson, 2011). However, the iris center detection method often works under visible light. Visible light is not as sensitive to the IR light in the outdoor environment. The eye tracker with iris center detection method need less hardware devices and the cost of this type is cheaper than eye tracker with bright- and dark-pupil technique. In general, that eye tracker includes a camera, a computer and a visible light. At the same time, the iris size is stable compared with pupil size. With the above background, the usefulness of iris detection becomes much more evident.

However, because the eyeball can move freely within eye region, the iris edge detection method needs to con-

sider different states of the iris, i.e. center, left, right, and upper, as shown in Figure 1. Especially, if the human eyes gaze at objects on the left and right periphery of the screen, iris edges closer to eye corners would be hidden, i.e. the eyeball rolls into the nasal or temporal eye corner. In this case, we just obtain iris edge points on the uncovered iris edge, i.e. one side of the iris. Thus, it is hard to obtain the correct ellipse fitting with the detected points.

Although many eye gaze tracking systems based on iris center detection method, such as (Zhang, Zhang, & Chang, 2001), (Wang, Sung, & Venkateswarlu, 2005), (Perez, Lazcano, & Estevez, 2007), (Torricelli, Conforto, Schmid, & Alesio, 2008), and (Sigut & Sidha, 2011), were proposed by researchers, they seldom consider the states of eyeball within the eye region. Thus, as for gaze tracking in wide range, those systems are not ideal.

According to the discussion above, this paper presents an easy and efficient iris center detection method which considers the effect on accuracy of iris center detection for different states of iris within eye region. The proposed iris center detection method shows high positioning accuracy on eye images from our eye database and gaze estimation in our gaze tracking system. However, our proposed methods still have some uncertainties for iris edge detection. That has been proven to come from two reasons.

The first reason refers to the eye feature detection, namely the error positioning of rough iris center and eye corners. The successful run of our proposed method is based on a low false detection rate of each step. In other words, if the first step is not accurate, the following detection may be failed. Fortunately, the accuracy of the rough iris center and eye corners is higher through experimental results. However, the problem still existed in our eye gaze tracking system.

The second source of inaccuracy is that, in some extreme cases, if the gaze is directed towards the very lowest part of the camera, the eye can become semi-closed or closed. The proposed method does not achieve a high accuracy in iris center detection and eye gaze tracking due to occlusions from the eyelids and significant changes in iris shape.

Conclusion

An easy and efficient iris center detection method for eye gaze tracking system is presented in our paper. The method is based on modeling the geometric relationship between detected rough iris center and two eye corners and proposed active edge detection algorithm. The proposed method can automatically judge which iris edges need to be detected and extract iris edge points without any edge operators. Because the eyeball is an active structure, the iris often rolls into nasal and temporal eye corners. In this case, the part of the iris edge is hidden, making edge extraction of iris difficult. Hence, this paper presents a predicted edge points algorithm to enhance the accuracy of ellipse fitting. The evaluated results show the global average accuracy of 94.30% for four states of the iris within eye region when $T \leq 0.05$ and mean maximum error for the iris center is only 4.3% of the actual iris center and an edge point. Also, compared with other existing methods, our method achieves the highest iris center detection accuracy.

The proposed iris center detection method has been used in our gaze tracking system. The achieved average accuracies of gaze estimation for the four states of the iris are 0.99° in horizontal direction and 1.33° in vertical direction, respectively. Compared with other iris center detection methods, the proposed method enhanced the globe average accuracy of gaze tracking. Future efforts will be devoted to development and optimization of our method used in eye gaze tracking system. Especially, as for the two problems in *Discussion* section, we need to find better solutions. In addition, the gaze tracking system will be used in human-robot interaction and gaze gesture research fields. As for human-robot interaction research, the gaze tracking system can work outdoors to control agents, such as Drone and robotic vehicles, using eye gaze.

Acknowledgements

This work has been financially supported by Program of the National "985" Project -- Phase III of Beijing Institute of Technology, China Scholarship Council (No. 201306030055), U.S. National Science Foundation (NSF) through the grant No. 0954579 and No. 1333524. The suggestions from the anonymous reviewers are greatly

acknowledged. Special thanks also go to the participants who have participated in this work.

Appendix A

Algorithm1 Pseudocode of iris edge points detection as shown in the Pseudocode below

Initialize parameters

- Initial point P_{init} .
- Initial radius $r_{init} = \|P_{init} - P_c\|$.
- Interval angle $\Delta\theta = \gamma/n$.
- Initial the search Range $R_s = [r_{init} - \delta, r_{init} + \delta]$.
- Initial the array A .
- $Count_{err} = 0$.

Iterate through all edge points

- $r_1 = r_{init}$.
- $P_1 = P_c$.
- Take P_1 into A .
- For** $i = 1$ **to** n
 - Search the edge point P_{i+1} within R_s .
 - $r_{i+1} = \|P_{i+1} - P_c\|$.
 - if** $Count_{err} > T_{err}$ **then**
 - Quit search algorithm.
 - end if**
 - if** $|r_{i+1} - r_i| > Threshold$ and $r_{i+1} - r_i > 0$ **then**
 - $r_{i+1} = r_i$.
 - $R_s = [r_{i+1} - \delta - \Delta\delta, r_{i+1} + \delta - \Delta\delta]$.
 - $Count_{err} = Count_{err} + 1$.
 - else if** $|r_{i+1} - r_i| > Threshold$ and $r_{i+1} - r_i < 0$ **then**
 - $r_{i+1} = r_i$.
 - $R_s = [r_{i+1} - \delta + \Delta\delta, r_{i+1} + \delta + \Delta\delta]$.
 - $Count_{err} = Count_{err} + 1$.
 - else**
 - $|r_{i+1} - r_i| < Threshold$
 - $R_s = [r_{i+1} - \delta, r_{i+1} + \delta]$.
 - Take P_{i+1} into A .
 - $Count_{err} = 0$.
 - end if**
- end for**

where n is the number of angle separations, A represents the array of edge points.

Appendix B

Algorithm2 Pseudocode of predicted edge points of iris as shown in the Pseudocode below

Initialize parameters

- Initial the array A_{PE} .
- $r_{lst} = \|P_{lst} - P_c\|$
- Initial the angle $\angle P_{lst} P_c P_{init} = \chi$.
- Initial numbers of search $n = (\chi - \lambda) / \Delta\theta$.

Iterate through all predicted edge points

For $i = 1$ **to** n

if $r_{lst} - r_{init} > 0$ **then**

- Compute coordinates of predicted edge points:

$$X_{P_e} = X_{P_c} + (r_{init} + V) \cos(\chi + i\Delta\theta)$$

$$Y_{P_e} = Y_{P_c} + (r_{init} + V) \sin(\chi + i\Delta\theta)$$

- Take P_e into A_{PE}

else

- Compute coordinates of predicted edge points:

$$X_{P_e} = X_{P_c} + (r_{init} - V) \cos(\chi + i\Delta\theta)$$

$$Y_{P_e} = Y_{P_c} + (r_{init} - V) \sin(\chi + i\Delta\theta)$$

- Take P_e into A_{PE}

end if

end for

Where P_e represents predicted edge point, (X_{P_e}, Y_{P_e}) is denoted as the coordinates of predicted edge points, $(\chi - \lambda)$ represents predicted search angle range, V is the distance constant.

References

- Barea, R., Boquete, L., Mazo, M., & Lopez, E. (2002). System for Assisted Mobility Using Eye Movement Based on Electrooculography. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(4), 209-218.
- Beymer, D., & Flickner, M. (2003). Shifts in reported gaze position due to changes in pupil size: ground truth and compensation. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (pp. 451-458). New York, NY, USA.

- Blignaut, P. J. (2013). Mapping the pupil-glint vector to gaze coordinates in a simple video-based eye tracker. *Journal of Eye Movement Research*, 7(1), 1-11.
- Cai, H., & Lin, Y. (2012). Coordinating Cognitive Assistance With Cognitive Engagement Control Approaches in Human-Machine Collaboration. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, Systems and Humans*, 42(2), 286-294.
- Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11), 1148-1161.
- Dobes, M., Martinek, D., Skoupil, Z., Dobesova, J., & Pospisil, J. (2006). Human eye localization using the modified Hough transform. *Optik*, 117(10), 468-473.
- Duchowski, A. T., Shivashamkaraish, V., Rawls, T., Gramopadhye, A. K., Melloy, B. J., & Kanke, B. (2000). Binocular eye tracking in virtual reality for inspection training. In: *Proceedings of the symposium on Eye tracking research & applications, ETRA'00* (pp. 89-96). New York, USA.
- Ferhat, O., Vilarino, F., & Sanchez, F. J. (2014). A cheap portable eye-tracker solution for common steps. *Journal of Eye Movement Research*, 7(3), 1-10.
- Feng, G. C., & Yuen, P. C. (2001). Multi-cues eye detection on grey intensity image. *Pattern Recognition*, 34(5), 1033-1046.
- Flores, M. J., Armingol, M. J., & Escalera, A. D. (2011). Driver Drowsiness Detection System under Infrared Illumination for the Intelligent vehicle. *IET Intelligent Transport Systems*, 5(4), 241-251.
- Fitzgibbon, A., Pilu, M., & Fisher, R. (1999). Direct least square fitting of ellipse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 476-480.
- Hennessey, C., Nouredin, B., & Lawrence, P. (2008). Fixation Precision in High-Speed Noncontact Eye-Gaze Tracking. *IEEE Transactions on Systems, Man, and Cybernetics Part B, Cybernetics*, 38(2), 289-298.
- Hooge, I., Nyström, M., Cornelissen, T., & Holmqvist, K. (2015). The art of braking: Post saccadic oscillations in the eye tracker signal decrease with increasing saccade size. *Vision Research*, 112, 55-67.
- Jesorsky, J., Kirchberg, K. J., & Frischolz, W. (2001). Robust Face Detection Using the Hausdorff Distance. In: *Third International Conference on Audio- and Video-based Biometric Person Authentication* (pp. 90-95). Halmstad, Sweden.
- Kimmel, D., Mammo, D., & Newsome, W. (2012). Tracking the eye non-invasively: simultaneous comparison of the scleral search coil and optical tracking techniques in the macaque monkey. *Frontiers in Behavioral Neuroscience*, 6(49), 312-331.
- Li, D., Winfield, D., & Parkhurst, D. J. (2005). Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *IEEE Computer Society Conference on CVPR Workshops* (pp. 79). San Diego, CA, USA.
- Li, D., Winfield, D., & Parkhurst, D. J. (2005). Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. *IEEE Computer Society Conference on CVPR Workshops* (pp. 79). San Diego, CA, USA.
- Liversedge, S. P., Meadmore, K., Corck-Adelman, D., Shih, S., & Pollatsek, A. (2011). Eye movements and memory for objects and their locations. *Studies of Psychology and Behavior*, 9(1), 7-14.
- Matsumoto, A., & Zelinsky, A. (2000). An algorithm for Real-time stereo vision implementation of head pose and gaze direction measurement. In: *Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 499-504). Grenoble, France.
- Mohammadi, M. R., & Raie, A. (2012). Robust Pose-Invariant Eye Gaze Estimation Using Geometrical Features of Iris and Pupil Images. In: *20th Iranian Conference on Electrical Engineering* (pp. 593-598). Tehran, Iran.
- Morimoto, H. C., Koons, D., Amir, A., & Flickner, M. (2000). Pupil detection and tracking using multiple light sources. *Image Vision and Computing*, 18(4), 331-335.
- Nyström, M., Hooge, I., & Holmqvist, K. (2013). Post-saccadic oscillations in eye movement data recorded with pupil-based eye trackers reflect motion of the pupil inside the iris. *Vision Research*, 11, 59-66.
- Oyster, C. W. (1999). *The Human Eye: Structure and Function*. Sunderland, MA: Sinauer Associates.

- Papageorgiou, E., Hardiess, G., Mallot, H. A. & Schiefer, U. (2012). Gaze patterns predicting successful collision avoidance in patients with homonymous visual field defects. *Vision Research*, 65(15), 25-37.
- Perez, C. A., Lazcano, V. A., & Estevez, P. A. (2007). Real-Time Iris Detection on Coronal-Axis-Rotated Faces. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Application and Reviews*, 37(5), 971-978.
- Robinson, D. A. (1963). A method of measuring eye movements using a scleral search coil in magnetic field. *IEEE Transactions on Biomedical Engineering*, 10(4), 137-145.
- Sankowski, W., Grabowski, K., Napieralska, M., Zubert, A., & Napieralski, A. (2010). Reliable algorithm for iris segmentation in eye image. *Image and Vision Computing*, 28(2), 231-237.
- Sigut, J., & Sidha, S. A. (2011). Iris Center Corneal Reflection Method for Gaze Tracking Using Visible Light. *IEEE Transactions on Biomedical Engineering*, 58(2), 411-419.
- Sirohey, S. A., Rosenfeld, A., & Duric, Z. A. (2002). A method of detecting and tracking irises and eyelids in video. *Pattern Recognition*, 35(6), 1389-1401.
- Świrski, L., Bulling, A., & Dodgson, N. (2012). Robust real-time pupil tracking in highly off-axis images. In: *Proceedings of the Symposium on Eye Tracking Research and Applications* (pp. 173-176). New York, NY, USA.
- Toricelli, D., Conforto, S., Schmid, M., & Alesio, T. A. (2008). A neural-based remote eye gaze tracker under natural head motion. *Computer Methods and Programs in Biomedicine*, 92(1), 66-78.
- Tseng, P., Cameron, I. G. M., Pari, G., Reynolds, J. N., Munoz, D. P., & Itti, L. (2012). High-throughput classification of clinical populations from natural viewing eye movements. *Journal of Neurology*, 260(1), 275-284.
- Wang, J., & Venkateswarlu, R. (2002). Study on Eye Gaze Estimation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, Cybernetics*, 32(3), 332-350.
- Wang, J., Sung, E., Venkateswarlu, R. (2005). Estimating the eye gaze from one eye. *Computer Vision and Image Understanding*, 98(1), 83-103.
- Yu, M., Lin, Y., Schmidt, D., Wang, X., & Wang, Y. (2014). Human-robot interaction based on gaze gestures for the drone teleoperation. *Journal of Eye Movement Research*, 7(4), 1-14.
- Yu, M., Wang, X., Lin, Y., & Bai, X. (2014). Gaze Tracking System for Teleoperation. *26th Chinese Control and Decision Conference* (pp. 4617-4622). Changsha, China.
- Zhang, W., Zhang, T., & Chang, S. (2001). Eye gaze estimation from the elliptical features of one iris. *Optical Engineering*, 50(4), 047003-047003-9.
- Zhou, Z., & Geng, X. (2004). Projection functions for eye detection. *Pattern Recognition*, 37(5), 1049-1056.
- Zhu, J., & Yang, J. (2002). Subpixel Eye Gaze Tracking. In: *Fifth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 124-129). Washington, DC, USA.