Identification of Eye Movements from Non-frontal Face Images for Eye-controlled Systems

Min Lin^{1,2} Bin Li¹ Qiao-Hong Liu¹

¹School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China

²Shanghai Medical Instrumentation College, University of Shanghai Science and Technology, Shanghai 200093, China

Abstract: The novel eye-based human-computer interaction (HCI) system aims to provide people, especially, disabled persons, a new way of communication with surroundings. It adopts a series of continual eye movements as input to perform simple control activities. Identification of eye movements is the crucial technology in these eye-based HCI systems. At present, researches on eye movement identification mainly focus on frontal face images. In fact, acquisition of non-frontal face images is more reasonable in real applications. In this paper, we discuss the identification process of eye movements from non-frontal face images. Firstly, the original head-shoulder images of $0^\circ - \pm 60^\circ$ azimuths are sampled without any auxiliary light source. Secondly, the non-frontal face region is detected by using the Adaboost cascade classifiers. After that, we roughly extract eye windows by the integral projection function. Then, we propose a new method to calculate the $x - y$ coordinates of the pupil center point by searching the minimal intensity value in the eye windows. According to the trajectory of the pupil center points, different eye movements (eye moving left, right, up or down) are successfully identified. A set of experiments is presented.

Keywords: Eye movement identification, non-frontal face images, pupil center localization, human-computer interaction, eye-based system.

1 Introduction

Today, there are many new technologies studied for thousands of disabled persons to aid their communication and mobility in order to increase their life quality and allow them a more autonomous and independent life style and greater chances of social integration^[1, 2]. The intelligent human-computer interaction (HCI) system based on visual information is one of these innovation technologies which are becoming more and more popular.

The eye-based HCI system adopts a series of continual eye movements as input to perform simple control activities. The common eye movements include saccades, fixations and blinks. According to the methods of recording eye motions, the eye-based HCI system can mainly be classified into two categories: invasive and active versus non-invasive and passive^[3]. The active electrooculography (EOG) system needs to attach 3-5 electrodes on the skin around the eye region (as shown in Fig. 1) in order to measure the resting potential of the retina reflecting to the eye movements, which is intrusive and makes the user uncomfortable^[4-6]. Moreover, the contact resistance existing between the skin and the electrodes would decrease the accuracy of recording the EOG signal.

The passive video-oculography (VOG) system^[7-9] is based on digital images of eyes captured with a video camera and coupled with image processing and machine vision technologies, as shown in Fig. 2. The VOG system has no connected devices on the user's body, so the user may be unaware of the monitoring system. For its more natural interaction with the user, the VOG system attracts more

researchers' attention.

Fig. 1 EOG-based system using electrodes

Fig. 2 VOG-based system using video cameras

The VOG system encompasses a number of component technologies, including spatial eye position tracking, eye-gaze tracking, eye closure state tracking, eye movement tracking and pupil size monitoring^[3]. Among these technologies, the eye gaze tracking and the eye movement tracking are easily confused by researchers. In fact, they have different working principles and application fields.

Gaze tracking is the most widely known technology among the above four components. It means the process of finding the exact point on a monitor screen which the user is gazing at^[10–12]. Generally, it computes the gaze point on the screen based on the pupil center corneal reflection (PCCR) method^[13] which needs at least one infrared light emitting diode (LED) as the auxiliary light source. An infrared LED causes a reflection spot on the eyeball. As the eye is round, the reflection spot stays at the same position no matter in which direction the eye is looking. A video camera detects the reflection spot and the center of the pupil. The direction of the eye-gaze can be calculated from the distance of both points by simple linear mapping, as shown in Fig. 3. The eye gaze technology has been applied in the smart “eye-controlled” phones from the Samsung Company. It can automatically scroll the webpage or the window when the user gazes at the bottom of the screen of the phone. Normally, the screen is another essential component besides the video camera and the LEDs in the eye gaze tracking system.

Fig. 3 PCCR method used to calculate the point-of-gaze (POG). The parameters are: radius of the cornea r , distance from the center of the cornea to the center of the pupil r_d , and the index of refraction of the aqueous humor n . The center of the cornea is located at point c and the center of the pupil is located at point p_c . The optical axis L is the vector formed from c to p_c , and the POG p is the intersection of the OA with the monitor plane

Eye movement tracking means tracking and interpreting different eye activities. These eye activities include fixations, saccades and blinks. Sometimes, they could be captured by an eye tracker. The technology of eye movement tracking doesn't need the infrared LEDs as the auxiliary light sources. Also it doesn't need the screen either. It continually captures eye movement images through the video camera and then recognizes the different eye activities by adopting advanced image processing technologies. Fig. 4 shows the diagram of an eye movement tracking system. Furthermore, the identified eye activities can be encoded as input commands to the HCI interfaces, which can control some simple devices^[14, 15]. For example, when your eyes move to the left, it may guide the wheelchair to the left, your eyes moving right may guide the wheelchair to the right. So, it could be called an eye-controlled system.

Fig. 4 Diagram of eye movement tracking system

The eye-controlled system can not only help people interact with the computer or other devices with screen, but also help people to control some simple devices, such as the on-off switch and the TV at home just through rotating their eyes. Therefore, it is more valuable for the disabled person.

In the eye-controlled system, successful identification of different eye movements is very important. Some intelligent eye-tracker products can identify the eye movements. But almost all the eye trackers need infrared LEDs as the auxiliary light source. These eye trackers must be worn on the head, which would make the user uncomfortable. Moreover, they are very expensive to the ordinary people. For these disadvantages, the eye trackers are not popular in the disabled population.

Today, the image processing technology has greatly developed. But so far, there are few references on using the image processing methods to realize recognition of eye movements. In this paper, we adopt the software (image processing methods) rather than the hardware (eye trackers) to recognize the eye movements for the eye-controlled system.

The paper is organized as follows. Section 1 explains the difference between the eye gaze tracking technology and the eye movement technology. Section 2 discusses the proper placement region of the video camera under the natural light by constructing a mathematical model on the Matlab platform. Section 3 emphasizes the methods for identification of the eye movements from non-frontal face images including image acquisition, face detection, eye window extraction and eye movement recognition. Experiments are presented in Section 4.

2 Modeling of single camera location under the natural light

More and more VOG systems based on the eye gaze tracking technologies use two or more video cameras and several infrared LEDs as the auxiliary light source to achieve more accurate point of gaze with free head motion. However, all the video cameras must be calibrated. This will increase the complexity of the system^[16, 17]. In addition, these systems are limited indoors because of the LEDs light source. The eye controlled system discussed in this paper is based on the eye movement tracking technology. It is composed of only one video camera and one computer, without any auxiliary light source to produce glints or any image projecting screen to compute the gaze point. The camera is used to capture the user's eye moving images in real time, while the computer is used to process the images or analyze data. So, it can simplify or avoid the calibration process and reduce the cost. Also, the system can be used both indoors and outdoors.

To achieve a satisfying accuracy, we should analyze the influence of the head motion on the eyes image acquisition

when only one video camera is used.

2.1 Model of eyes rotation with head motion

Assuming the user's eyes are in the same high plane as the optical axis of the video camera, we design a mathematical model (as shown in Fig. 5) on the Matlab platform. In Fig. 5, C is the video camera, line CO is the optical axis of the video camera, L is the lateral canthus of the left eye, R is the lateral canthus of the right eye, apex angle is the horizontal visual angle of the video camera. The opposite side LR with the length w is the line between the left and the right lateral canthi. O is the midpoint of line LR . The length of one eye is d . Line LR represents the original eyes position when the user faces straight to the video camera. We adjust the focus of the camera so that the straight line LR must be in the rectangle imaging plane of the video camera.

Fig. 5 Model of eyes rotation with head motion

If the user's head rotates by α degrees to the right side, then line LR connecting the two eyes lateral canthi should rotate to the $L'R'$ position. Note that the video camera keeps still during the head motion. By projecting the two corner points N and R' of the right eye onto the horizontal axis of the rectangle imaging plane of the video camera, points N' and M' are obtained, respectively. Because the length of the eye is constant, i.e., the length of line $R'N$ equals d , the relation of $R'N$ and $M'N'$ can be given as

$$R'N = d \quad (1)$$

$$M'N' = d \times \cos \alpha. \quad (2)$$

We discuss on the allowable maximum degree of head rotating: If the length of line $M'N'$ is too short, then the complete two-eye image could not be captured on the imaging plane of the video camera. It means that the length of line $M'N'$ should be longer than half length of the eye, i.e.

$$d \times \cos \alpha \geq \frac{d}{2} \quad (3)$$

$$0^\circ \leq \alpha \leq 60^\circ. \quad (4)$$

So, the allowable maximum degree of the head horizontally rotating to the right or left side is 60° .

2.2 Model of single camera location

We have known that the widest range of the head moving is within $\pm 60^\circ$ ("±" represents the left or right direction). In actual application, the video camera should be placed at one side of the user to avoid shading user's sightlines. Where is the appropriate location of the video camera? Few references are found to focus on this question. In our previous paper^[18], a mathematical model was made to discuss the reasonable location of the video camera as shown in Fig. 6. To simplify the model, we assume the video camera is as high as the user's eyes, i.e., the lens of the camera locate in the same plane as eyes. And the optical axis of the camera points at the eyes. The line between the user's two eyes is regarded as the x -axis in a rectangular coordinate system, the midpoint of the line is the origin. Therefore, what the optical axis pointing at is the y -axis direction when the video camera is placed straight ahead of the user.

In Fig. 6, apex C is the video camera, apex angle θ is the horizontal visual angle of the camera, the opposite side LR with the length w is the line between the two lateral canthi. Parameters θ and w are known. We rotate apex C (camera) around the midpoint O of the opposite side while keeping θ and w constant. When the camera moves from point C to C' (or C''), there is an angle between its present optical axis direction $C'O$ (or $C''O$) and the original optical axis direction, which is called the horizontal deflection angle β of the camera. In fact, the moving trace of apex C is the nearest positions of the camera to the user's eyes in every β direction.

(a) Location 1

(b) Location 2

Fig. 6 Modeling of the camera's location at the right side

In Fig. 6 (a), line $C'O'$ is vertical to line LR , and $|C'O| = r'$. According to the triangle relations, a series of equations are got as

$$\begin{aligned}\theta &= \theta_1 + \theta_2 \\ \beta &= \theta_2 - \varphi\end{aligned}\quad (5)$$

$$\gamma = \frac{\pi}{2} - \beta - \theta_1 = \frac{\pi}{2} - \beta - (\theta - \theta_2) = \frac{\pi}{2} - \theta + \varphi \quad (6)$$

$$\frac{\frac{w}{2}}{\sin \theta_1} = \frac{r'}{\sin \gamma} = \frac{r'}{\cos(\theta - \varphi)} \quad (7)$$

$$\frac{\frac{w}{2}}{\sin \theta_2} = \frac{r'}{\sin \delta} = \frac{r'}{\cos \varphi}. \quad (8)$$

From these equations, the solution is

$$r'^2 = \frac{w^2}{4} \times [(4\cot^2\theta + 1) \times \cos^2\varphi + 4\cot\theta \times \sin\varphi \times \cos\varphi + \sin^2\varphi] \quad (9)$$

$$\tan\varphi = \frac{\frac{w}{2}}{r' \cos\beta} \Rightarrow \beta = \sin^{-1} \left[\frac{w}{2} \times \frac{\cos\varphi}{r'} \right] - \varphi. \quad (10)$$

Similarly, in Fig. 6 (b), line $C''O''$ is vertical to line LR , and $|C''O| = r''$. The solution for Fig. 6 (b) is

$$r''^2 = \frac{w^2}{4} \times [(4\cot^2\theta + 1) \times \cos^2\varphi - 4\cot\theta \times \sin\varphi \times \cos\varphi + \sin^2\varphi] \quad (11)$$

$$\beta = \sin^{-1} \left[\frac{w}{2} \times \frac{\cos\varphi}{r''} \right] + \varphi. \quad (12)$$

The comprehensive conclusion by integrating (9)–(12) is

$$r = \sqrt{\frac{w^2}{4} [(4\cot^2\theta + 1) \cdot \cos^2\varphi \pm 4\cot\theta \cdot \sin\varphi \cdot \cos\varphi + \sin^2\varphi]} \quad (13)$$

$$\beta = \sin^{-1} \left[\frac{w}{2} \cdot \frac{\cos\varphi}{r} \right] \mp \varphi \quad (14)$$

where θ is the horizontal vision angle of the video camera, and w is the length between two eyes lateral canthi.

Parameter r in (13) is the shortest distance between the video camera and the user's eyes. It means the horizontal visual angle of the video camera can cover the complete two eyes on the captured eye image when the camera locates at these nearest positions. And the coordinates $(r \sin \beta, r \cos \beta)$ represent the location of the camera. The moving of the apex point C (camera) is traced with the corresponding r and β .

A detailed deducing process can be found in [18], and the parameter is computed on Matlab platform. The Cognex In-Sight Micro1020 camera with the Computer M2514-MP lens is used to capture and store the eye images of the model head, as shown in Fig. 7. The horizontal visual angle is $\theta = 20^\circ$ and the length of two eyes' lateral canthi of the model head is $w = 94$ mm. At first, the camera is straight ahead of the user, so the parameter $\varphi = \frac{\theta}{2} = 10^\circ$.

We shift the video camera to the right side slowly, and then angle φ decreases gradually. When the video camera is just in front of the right eye lateral canthus, angle α decreases to 0° . When the camera locates beyond the lateral canthus of the right eye, angle φ starts to increase gradually. Given $\varphi = 0^\circ, 1^\circ, 2^\circ, \dots, 90^\circ$ in (13) and (14), the values of r and β are computed in pairs, as shown in Table 1.

Table 1 Computing results of the shortest distance between the camera and the user

φ	10°	9°	8°	7°	6°	5°	4°	3°	2°	1°	0°
β	0°	1°	2.1°	3.1°	4.1°	5.2°	6.2°	7.2°	8.3°	9.3°	10.3°
$r_{\min}$ (mm)	266.6	266.5	266.4	266.2	265.9	265.5	265.1	264.6	264.0	263.3	262.5
φ	0°	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°
β	10.3°	11.4°	12.4°	13.4°	14.4°	15.5°	16.5°	17.5°	18.6°	19.6°	20.7°
$r_{\min}$ (mm)	262.5	261.7	260.7	259.7	258.6	257.5	256.2	254.9	253.5	252.0	250.5
φ	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°
β	25.8°	31.0°	36.2°	41.5°	46.8°	52.1°	57.5°	63.1°	68.8°	74.9°	81.6°
$r_{\min}$ (mm)	241.6	230.9	218.4	204.3	188.6	171.5	153.0	133.5	112.9	91.5	69.5
φ	70°	75°	80°	85°	90°						
β	90°	104.6°	160°	94.6°	90°						
$r_{\min}$ (mm)	47.0	24.7	8.3	24.7	47.0

Fig. 7 The video camera and the model head

From the simulation and computation results of Table 1, we can find the conclusions as follows:

1) The shortest distance r between the video camera and the midpoint of the two eyes lateral canthi decreases when α drops from 10° to 0° . Meanwhile, the deflection angle β of the optical axis of the camera approximately increases linearly.

2) When the camera is just in front of the user's eyes, i.e., $\beta = 0^\circ$, parameter r reaches the maximum value, $r = 266.6$ mm. With angle β growing, the distance r drops gradually. When $\beta = 10.3^\circ$, the camera just locates in front of the right eye lateral canthus.

3) If $\varphi \geq 60^\circ$, then $r < 100$ mm. It means the camera is very close to the eyes. It isn't reasonable, so these positions should be given up. Parameter φ should be smaller than 60° .

Based on these simulation results, we capture the images of the model head in each β direction (Fig. 7) by shifting the video camera. We find that when the deflection angle β is less than 40° , complete two-eye images can be captured. When β is more than 40° , it can acquire only one-eye images rather than two-eye images, as shown in Fig. 8.

Test results show that the single video camera should locate at one side of the user within 40° so as to capture the complete two-eye images. And the actual distance between the video camera and the user's head should be farther than the theoretically calculated shortest value r in each β direction.

Fig. 8 Eye images in different β direction

3 Identification of eye movements from non-frontal face images

Identification of eye movements is very important to the HCI system based on visual information. Many researchers have been devoting themselves to identify eye movements through all kinds of EOG signal processing methods. In this paper, we try to identify the eye movements through image processing methods. Fig. 9 shows the processing diagram of recognizing eye movements from the sampled images. The image processing procedure includes 4 steps: 1) face detection from the non-frontal head-shoulder images, 2) rough extraction of the eye window from the detected face image, 3) accurate localization of the pupil center in the eye window, 4) identification of different eye movements according to the changes of the pupil center trajectory.

Fig. 9 The working process of the eye-controlled system

3.1 Image acquisition

We use the In-Sight Micro1020 video camera from the Cognex Company to sample the head-shoulder images under the natural light in our lab. During the acquisition, the video camera is placed at a fixed position and the user is asked to turn to the left side or the right side. The application software "Compass" installed in the iPhone 4 smart cellphone of Apple is adopted to accurately adjust the position of the user. The user is about 1.5 meters away from

the video camera. The detailed image acquisition process is illustrated as follow.

First, the user is asked to sit directly in front of the video camera. And the cellphone is put on the chair. We start the Compass software and adjust the azimuth to 0° as the original position. Then, we start the In-Sight Explorer Easy Builder View application software from the Cognex Company to sample the head-shoulder images with a 2 Hz sampling rate. During the acquisition period of 15 seconds, the user is asked to continually move his eyes to the left, right, upwards and downwards in a sequential order without head motion.

Second, the user is asked to turn left or turn right to the 10° azimuth indicated by the Compass software without moving the video camera. We capture 60 head-shoulder images with the same frequency while the user moves his eyes in sequence.

Third, we repeat this acquisition process at each azimuth of $\pm 20^\circ$, $\pm 30^\circ$, $\dots$, $\pm 60^\circ$ (“+” corresponding to the right side, and “-” corresponding to the left side). Thus, we capture total 390 images in different azimuths. Each image has the same size of 640×480 pixels.

3.2 Nonfrontal face detection

Among the face detection methods, the Adaboost-based cascade classifier algorithm^[19, 20] has the shortest response time, which is classical and popular. However, the algorithm can only detect the frontal faces. An improved Adaboost algorithm based on the extended rectangle features proposed by Lienhart^[21] can detect the non-frontal faces. According to the principle and work steps of this improved Adaboost algorithm, the non-frontal face detection program is developed based on the OpenCV library^[22]. Table 2 lists the non-frontal face detection results in 0° – 60° azimuths. There are two evaluation indices which are important for face detection. One is hit rate, and the other is false detec-

tion. The hit rate is defined as the ratio of the number of successfully detected faces and the number of total images. The false detection indicates the number of falsely detected faces. An idealized algorithm for face detection should have a 100% hit rate and no false detection. Results from Table 2 show that if the video camera locates on the side of the user within $\pm 30^\circ$, then the non-frontal face hit rate is 100% and the false detection is 0. But the hit rate would drop off or the number of false detection would increase if the azimuth is beyond $\pm 30^\circ$. Fig. 10 shows parts of successful nonfrontal face detection examples. Each row represents one azimuth which is adjusted by the Compass software in the iPhone. For example, “ 10° L” means the user sits on the left side of the video camera with 10° azimuth. Each column represents a different eye gesture. For example, “straight” means the user looks straight ahead, “left” means the user’s eyes are moving to the left. The rectangle labels the detected face region in each head-shoulder image. We extract the face region from each head-shoulder image with the union size of 150×150 pixels for further processing.

3.3 Eye windows extraction

In general, all image projection functions can be used to detect the boundary of different image regions. In a face image, there are eyebrows, nose and lips besides eyes. There are two distinct characteristics in the eye area. The eye area is darker than its neighboring areas. Also, the intensity of the edge of the eyeball rapidly changes. So some researchers use integral projection function (IPF) method to locate two eyes in the frontal face images. Experiments made by us show the accuracy of locating eye center point in the non-frontal face is not satisfying. But we can roughly extract eye windows from the detected face images based on the results from IPF to prepare for the pupil center localization. Eye windows mean the image regions including eyes^[23].

Table 2 Non-frontal face detection results in 0° – 60°

Azimuth	Total images	Missing detection	Hit rate (%)	False detection
0°	30	0	100	0
10° Left	30	0	100	0
10° Right	30	0	100	0
20° Left	30	0	100	0
20° Right	30	0	100	0
30° Left	30	0	100	0
30° Right	30	0	100	0
40° Left	30	2	93.33	0
40° Right	30	0	100	1
50° Left	30	5	83.33	4
50° Right	30	0	100	0
60° Left	30	15	50	15
60° Right	30	11	63.33	1

Suppose $I(x, y)$ is the intensity of a pixel at location (x, y) , the horizontal integral projection $IPF_h(x)$ and the vertical integral projection $IPF_v(y)$ of $I(x, y)$ in intervals $[x_1, x_2]$ and $[y_1, y_2]$ can be defined as^[24]

$$IPF_h(y) = \int_{x_1}^{x_2} I(x, y) dx \quad (15)$$

$$IPF_v(x) = \int_{y_1}^{y_2} I(x, y) dy. \quad (16)$$

The detailed procedure of using IPF to roughly extract eye windows from the non-frontal face images with the union size of 150×150 pixels is as follows.

Step 1. Pre-processing the detected face image. This step includes 3 sub-steps. 1) Converting the color image to the gray scale image. 2) Equalizing the histogram of the gray scale image to compensate the illumination. 3) Automatically segmenting the gray scale image by using the Otsu method^[25] to get a binary image.

Step 2. Making the horizontal integral projection. The left eye and the right eye may not be in a horizontal line when the user doesn't face straight to the video camera. So we use (15) to perform the horizontal IPF on the left half face and right half face, and get the two corresponding y -coordinates (y_L, y_R) for the two eyes.

Fig. 10 Successful non-frontal face detection results in 0° – 30°

Step 3. Making the vertical integral projection. According to (16), we can calculate the x -coordinates (x_L, x_R) of the two eyes.

Then, the roughly estimated left eye center point is $P_L(x_L, y_L)$ and the roughly estimated right eye center is

$P_R(x_R, y_R)$. In Fig. 11, symbol “*” denotes the rough eye center points calculated by the IPF method.

Step 4. Eye windows extraction based on the roughly estimated eye center points. Suppose the distance between P_L and P_R is d . Then the eye windows are rectangles of the size of $0.6d \times 0.3d$. The coordinates of the top left corner of the right eye window could be set as $(x_{\text{border}}, y_R - 0.15 \times d)$, and the coordinates of the top left corner of the left eye window could be set as $(x_{\text{border}} + d, y_L - 0.15 \times d)$. The value of parameter x_{border} could be set in 0–10 pixels. For the right image in Fig. 11, e.g., $x_{\text{border}} = 5$ (pixels), $d = 120 - 36 = 84$ (pixels), then the top left corner of the right eye window is at (5, 43), and the top left corner of the left eye window is at (89, 38). These two eye windows have the same size of 50×25 pixels.

Sometimes the eye center points estimated by IPF method are outside the true eye sockets, and the calculated eye windows based on the estimated eye center points may not include the true eyes. To avoid missing the correct eye windows, we can choose the first frame of continually sampled face images to manually pre-estimate the range of eye sockets before starting to extract the eye windows. If the eye center points calculated by IPF are beyond the pre-estimated eye range, then the eye windows are the pre-estimated eye sockets rather than the calculated rectangle based on the IPF results.

Fig. 11 Eye windows extraction

In actual application, images are sampled continually. The head pose has few changes between any two neighboring frame images. Thus, we track the eye sockets successfully from totally 390 detected face images with different azimuths.

3.4 Pupil center locating

The eye is a slightly asymmetrical globe. The iris is the pigmented part in the eye. The pupil is the black circular opening in the iris that lets light in.

When eyes move, the position of pupils will change with eye movements. According to the change trajectory of the pupil center, different eye movements (left, right, up and down) could be recognized. Rigidly, the pupil would not appear to be a circle when the eye moves to the corner, shown as the left image in Fig. 11, i.e., the pupil center point may disappear at the time. So, the pupil center mentioned in this paper means the central point of the black region in the eye.

The popular method for achieving the coordinates of the pupil center is to calculate the centroid of the connected area on the basis of edge detection on the eyes^[26]. In this method, the centroid is regarded as the pupil center.

Among the edge detection algorithms, the Canny operator based method^[27] is commonly adopted. Because the two eyes always move towards the same direction at any time, we just need to detect the edge of one eye by using the Canny operator. But sometimes we can't get the desirable centroids. The failure may be caused by the property of the Canny edge detection algorithm itself. For the Canny method, it needs two threshold values to detect and connect the eye edge. When the user doesn't face straight at the camera, i.e., the user faces at the left or the right side of the camera with a deflection angle, the threshold values used in the Canny method need to be changed adaptively with different deflecting directions in real time. If the fixed threshold values are used, it is possible to detect the eye edge incompletely. Fig. 12 shows two failure cases of the eye edge detection. There is no desirably connected area or more than one connected area after Canny edge detection on the right eye so that we can't get the correct centroid which corresponds to the pupil center.

Eyes have a feature, i.e., the pupil is always darker than other parts in an eye. So, we propose a new method to achieve the accurate coordinates of the pupil center. The detailed steps of locating the pupil center are described as follows.

Fig. 12 Failure cases of the eye edge detection

Step 1. Filtering the right eye window by the Gaussian filter with the template $g = [0 \ 1 \ 0, \ 1 \ 1 \ 1, \ 0 \ 1 \ 0]$.

Step 2. Segmenting the right eye window in the horizontal and vertical directions respectively into 3 equal parts. In other word, segmenting the right eye window into 3×3 sub-areas, as shown in Fig. 13. The size of each sub-area is equal.

Step 3. Counting the number N of black pixels in each sub-area. Normally, the pupil area has the maximal value of the number N .

Step 4. Searching the point with the minimal intensity value in the pupil area, which corresponds to the pupil center point (the red “*” symbol shown in Fig.13).

ter point (the red “*” symbol shown in Fig.13).

Fig. 13 Segment the eye window into 3×3 sub-areas

According to these four steps, we calculate the pupil center coordinates of right eyes in different azimuths, as listed in Table 3. The first column, “Azimuth”, means a deflection angle between the user and the camera. The calculated pupil center point is represented with x - y coordinates in pixels on the detected face image.

We refer to the criterion provided by Jesorsky et al.^[28] to test the quality of pupil center locating. The accuracy of pupil center locating from the 28 face images corresponding to Table 3 is 96.43%. It shows that the proposed method can successfully locate the eye pupil center in non-frontal faces within $\pm 30^\circ$ side directions.

3.5 Eye movements identification

The pupil center changes with eye movements. According to the continually changing coordinates of pupil center, we can recognize different eye movements (left, right, up and down). In Fig. 14, it shows the relationship between the eye movements and the pupil center of any two neighboring images. When eyes move to the left, the x -coordinate value of the pupil center will increase. On the contrary, the x -coordinate value will decrease with eyes moving right. Similarly, the y -coordinate value decreases with eyes moving up. And the y -coordinate value increases with eyes moving down. Thus, we can identify the eye movements including eye moving left, moving right, and moving up and down. The prerequisite is that the images representing eye movements must be continually sampled.

4 Experimental example

Experiment is done to validate the proposed methods. In the experiment, the distance between the user and the camera is about 1 m. At first, the user faces straight to the video camera, i.e., the deflection angle between the user and the video camera is 0° . Then, the user turns to the left side

Table 3 x - y coordinates of the right pupil center in 0° – 30° azimuths (unit: pixel)

Azimuth	x - y coordinates			
	Glance straight	Glance left	Glance right	Glance upwards
0°	(37, 57)	(45, 58)	(26, 60)	(40, 53)
10° Left	(29, 52)	(39, 48)	(26, 54)	(28, 46)
10° Right	(33, 52)	(41, 51)	(22, 50)	(31, 43)
20° Left	(28, 49)	(37, 44)	(19, 55)	(23, 43)
20° Right	(30, 51)	(39, 52)	(23, 50)	(29, 44)
30° Left	(27, 42)	(32, 39)	(26, 52)	(33, 45)
30° Right	(30, 47)	(37, 53)	(22, 48)	(7, 40)

or right side of the video camera to make the deflection angle be $\pm 10^\circ$, $\pm 20^\circ$, $\pm 30^\circ$ (“+” corresponding to the right side, and “-” corresponding to the left side). For each azimuth, 30 images are captured at a rate of 2 fps in the horizontal or vertical direction, respectively.

Fig. 14 Relationship between eye movements and pupil center

During sampling the images of eye movements, the user is asked to move eyes according to a certain sequence like in Fig. 15. Through the procedures of face detection, eye windows extraction and pupil center locating, a series of x - y coordinates of pupil center points of the right eyes are obtained.

Fig. 15 Sequence of eye movements in the experiment

We plot these coordinates in a chronological sequence on the Matlab platform. In Fig. 16, the coordinate values of the right pupil center are plotted, with Fig. 16 (a)–(g) corresponding to 0° – $\pm 30^\circ$ azimuths, respectively. The x axis represents the time sequence while the y axis represents coordinate values. The blue “●” line and the red “■” line show the changing x -coordinate and the y -coordinate of the pupil center in the right eye when the user looks left or right, respectively. The green “▲” line and the purple “*”

line show the changing x -coordinate and the y -coordinate of the pupil center of the right eye when the user looks upwards or downwards, respectively. Note that these x -coordinate and y -coordinate take the eye socket as the reference.

By observing Fig. 16 carefully, we can know the horizontal eye movements and the vertical eye movements are recognized separately.

1) From the blue “●” lines, we can tell the eye movements in the horizontal direction. When the blue lines go up, i.e., the value of x -coordinate increases, it means the eyes move left. On the contrary, the blue lines going down means the eyes move right. For example, in Fig. 16 (a), one period of horizontal eye movement includes four segments: The first segment (points 1–5, C–L) represents the process of eyeball moving from the center of the eye socket to the left corner. The second segment (points 5–9, L–C) represents the process of eyeball returning from the left corner of the eye socket to the center. The third segment (points 11–14, C–R) represents the process of eyeball moving from the

(a) 0° (straight) direction

(b) 10° left-shift direction

(c) 10° right-shift direction

Fig. 16 Plots of coordinate values of pupil center in different directions

center to the right corner of the eye socket. The fourth segment (points 16–19, R–C) represents the process of eyeball returning from the right corner of the eye socket to the center. The x -coordinate values of points 9–11 are the same and it means the eyeball is stopping in the center of the eye socket. From point 20, it begins another circle of eye horizontal movement.

2) From the red “■” lines, we find that the changing values of y -coordinate are not noticeable when eyes move in the horizontal direction.

3) Similarly, we can tell the eye movements in the vertical direction from the purple “*” line. If purple lines go down, i.e., the value of y -coordinate decreases, then it means the eyes move up. On the contrary, the purple lines going up means the eyes move down. For example, in Fig. 16 (a), one period of vertical eye movement includes four segments: The first segment (points 1–7, C–U) represents the process of eyeball moving from the center of the eye socket to the up edge. The second segment (points 7–11, U–C) represents the process of eyeball returning from the up edge of the eye socket to the center. The third segment (points 11–15, C–D) represents the process of eyeball moving from the center to the down edge. The fourth segment (points 15–18, D–C) represents the process of eyeball returning from the down edge of the eye socket to the center. From point 19, it begins another period of vertical eye movement.

4) From the green “▲” lines, we find that the changing values of x -coordinate are not noticeable when the eyes move in the vertical direction.

5) According to the trajectory of the pupil center, the eye movements are identified. Comparing the recognized results with the actual sampled eye movement images, we can get the rates of eye movement identification at each azimuth (as shown in Table 4) which equals the ratio of the number of correct identifications and the total frames (30 frames at each azimuth). In Fig. 16 (a), when the deflection angle between the video camera and the user is zero, the relationship between the eye movements and the trajectory of the pupil center is clear. With the deflection angle increasing, it becomes more difficult to identify eye movements and the accuracy of identifying eye movements would decrease, as shown in Table 4. Also, we know that the identifying accuracy of eye horizontal movements is higher than that of eye vertical movements.

5 Conclusions

The experiment proves our method can identify eye horizontal and vertical movements from 420 non-frontal face images in $\pm 30^\circ$ with an average identification rate of 86.67%. Furthermore, we sample face images from different users who wear glasses. Results show that we can still detect these faces and achieve the pupil center location in $\pm 30^\circ$ side directions of the camera and successfully recognize the eye movements.

The success of simulation experiment on Matlab platform

Table 4 Eye movement identification rate in 0° – $\pm 30^\circ$ azimuths

Azimuth	Total images	Horizontal eye movement identification		Vertical eye movement identification		Average identification rate (%)
		False numbers	Identification rate (%)	False numbers	Identification rate (%)	
0°	30	1	96.7	1	96.7	96.7
10° Left	30	3	90.0	4	86.7	88.35
10° Right	30	3	90.0	4	86.7	88.35
20° Left	30	2	93.3	5	83.3	88.3
20° Right	30	2	93.3	4	86.7	90.0
30° Left	30	7	76.7	8	73.3	75.0
30° Right	30	5	83.3	7	76.7	80.0
Horizontal eye movement identification						86.67

gives us much confidence. It encourages us to realize the physical system with the electrical devices. So, it is our next work.

References

- [1] K. C. D. Fu, Y. Nakamura, T. Yamamoto, H. Ishiguro. Analysis of motor synergies utilization for optimal movement generation for a human-like robotic arm. *International Journal of Automation and Computing*, vol. 10, no. 6, pp. 515–524, 2013.
- [2] F. Chao, X. Zhang, H. X. Lin, C. L. Zhou, M. Jiang. Learning robotic hand-eye coordination through a developmental constraint driven approach. *International Journal of Automation and Computing*, vol. 10, no. 5, pp. 414–424, 2013.
- [3] R. I. Hammoud, J. B. Mulligan. Introduction to eye monitoring. *Passive Eye Monitoring: Algorithms, Applications and Experiments (Signals and Communications Technology)*, Berlin, Germany: Springer, pp. 1–19, 2008.
- [4] S. Venkataramanan, P. Prabhat, S. R. Choudhury, H. B. Nemade, J. S. Sahambi. Biomedical instrumentation based on electrooculogram (EOG) signal processing and application to a hospital alarm system. In *Proceedings of International Conference on Intelligent Sensing and Information Processing*, IEEE, Chennai, India, pp. 535–540, 2005.
- [5] L. Y. Deng, C. L. Hsu, T. C. Lin, J. S. Tuan, S. M. Chang. EOG-based human-computer interface system development. *Expert Systems with Applications*, vol. 37, no. 4, pp. 3337–3343, 2010.
- [6] L. Y. Deng, C. L. Hsu, T. C. Lin, J. S. Tuan, Y. H. Chen. EOG-based signal detection and verification for HCI. In *Proceedings of the 8th International Conference on Machine Learning and Cybernetics*, IEEE, Baoding, China, pp. 3342–3348, 2009.
- [7] D. Beymer, M. Flickner. Eye gaze tracking using an active stereo head. In *Proceedings of Computer Vision and Pattern Recognition*, IEEE, Madison, USA, vol. 2, pp. 451–458, 2003.
- [8] T. E. Hutchinson, K. P. White Jr., W. N. Martin, K. C. Reichert, L. A. Frey. Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1527–1534, 1989.
- [9] M. Betke, J. Gips, P. Fleming. The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 1, pp. 1–10, 2002.
- [10] C. Zhang, J. N. Chi, Z. H. Zhang, X. L. Gao. Gaze estimation in a gaze tracking system. *Science China Information Sciences*, vol. 54, no. 11, pp. 2295–2306, 2011.
- [11] H. Lee, N. Iqbal, W. Chang, S. Y. Lee. A calibration method for eye-gaze estimation systems based on 3D geometrical optics. *IEEE Sensors Journal*, vol. 13, no. 9, pp. 3219–3225, 2013.
- [12] R. J. K. Jacob, K. S. Karn. Eye tracking in human computer interaction and usability research: Ready to deliver the promises. *Computer Vision and Image Understanding*, vol. 24, pp. 682–701, 2003.
- [13] E. D. Guestrin, M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1124–1133, 2006.
- [14] A. Bulling, H. Gellersen. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 8–12, 2010.
- [15] R. Alonso, M. Causse, F. Vachon, R. Parise, F. Dehais, P. Terrier. Evaluation of head-free eye tracking as an input device for air traffic control. *Ergonomics*, vol. 56, no. 2, pp. 246–255, 2013.
- [16] A. Villanueva, R. Cabeza. Models for gaze tracking systems. *EURASIP Journal on Image and Video Processing*, vol. 2007, Article 23570, 2007.
- [17] H. Nakayama, N. Yabuki, H. Inoue, Y. Sumi, T. Tsukutani. A control system for electrical appliances using eye-gaze input. *IEEE International Symposium on Intelligent Signal Processing and Communications Systems*, IEEE, New Taipei, China, pp. 410–413, 2012.
- [18] M. Lin, B. Li. Modeling of single camera location under the natural light in the video-oculography system. *Advanced Materials Research*, vol. 663, pp. 638–644, 2013.

- [19] J. X. Wu, S. C. Brubaker, M. D. Mullin, J. M. Rehg. Fast asymmetric learning for cascade face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 369–382, 2008.
- [20] M. Castrillón, O. Déniz, D. Hernández, J. Lorenzo. A comparison of face and facial feature detectors based on the Viola-Jones general object detection framework. *Machine Vision and Applications*, vol. 22, no. 3, pp. 481–494, 2011.
- [21] R. Lienhart, J. Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings of International Conference on Image Processing*, IEEE, Rochester, USA, vol. 1, pp. 900–903, 2002.
- [22] A. Kaehler, G. R. Bradski. *Learning OpenCV*, USA: O'Reilly Media, Inc., 2008.
- [23] J. G. Wang, E. Sung, R. Venkateswarlu. Estimating the eye gaze from one eye. *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 83–103, 2005.
- [24] H. Z. Zhi, X. Geng. Projection functions for eye detection. *Pattern Recognition*, vol. 37, no. 5, pp. 1049–1056, 2004.
- [25] H. Z. Wang, Y. Dong. An improved image segmentation algorithm based on Otsu method. In *Proceedings of International Symposium on Photoelectronic Detection and Imaging: Related Technologies and Applications*, SPIE, Beijing, China, vol. 6625, 2007.
- [26] S. Megha, D. Sipi. Optimization of the edges detected by canny operator through segmentation. *International Journal of Computer Science Engineering and Technology*, vol. 2, no. 5, pp. 1176–1180, 2012.
- [27] H. I. Works. *Canny Edge Detector*, [Online], Available: <http://homepages.inf.ed.ac.uk>, March 10, 2013.
- [28] O. Jesorsky, K. Kirchberg, R. Frischholz. Robust face detection using the Hausdorff distance. In *Proceedings of the*

International Conference on Audio- and Video-Based Biometric Person Authentication, Lecture Notes in Computer Science, Springer, Halmstad, Sweden, vol. 2091, pp. 90–95, 2001.

Min Lin received the M.Sc. degree in testing and measurement technology and apparatus from Shanghai University, China in 2003. Now she is a Ph.D. candidate in control theory and control engineering in Shanghai University, China. Currently, she is a lecturer in Shanghai Medical Instrumentation College, University of Shanghai Science and Technology, China.

Her research interests include bioelectrical signal processing and control, human-machine interaction, and machine vision.

E-mail: Luc77sna@163.com (Corresponding author)

Bin Li is currently a doctoral supervisor at School of Mechatronic Engineering and Automation, Shanghai University, China.

His research interests include intelligent sensing and signal detection technology, instrumentation and automation device.

E-mail: sulibin@shu.edu.cn

Qiao-Hong Liu received the B.Sc. degree from Wuhan University of Technology, China in 2001, and the M.Sc. degree from Huazhong University of Science and Technology, China in 2004. She is currently a Ph.D. candidate at School of Mechatronic Engineering and Automation, Shanghai University, China.

Her research interests include computer vision, digital image processing, and image

deblurring techniques.

E-mail: hqlqh@163.com