Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8213946, 10 pages
https://doi.org/10.1155/2021/8213946

Research Article

Real-Time Precise Human-Computer Interaction System Based on Gaze Estimation and Tracking

Junhao Huang[1,2] **Zhicheng Zhang**,[3] **Guoping Xie**,[4] **and Hui He**[1]

[1]*Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai 519087, China*
[2]*Engineering Research Center of Intelligent Technology and Educational Application, Ministry of Education, Beijing 100875, China*
[3]*Department of Radiation Oncology, Stanford University, Stanford, CA 94305, USA*
[4]*State Information Center, Beijing, China*

Correspondence should be addressed to Hui He; 986293685@qq.com

Received 22 August 2021; Revised 5 October 2021; Accepted 21 October 2021; Published 8 November 2021

Academic Editor: Chi-Hua Chen

Noncontact human-computer interaction has an important value in wireless sensor networks. This work is aimed at achieving accurate interaction on a computer based on auto eye control, using a cheap webcam as the video source. A real-time accurate human-computer interaction system based on eye state recognition, rough gaze estimation, and tracking is proposed. Firstly, binary classification of the eye states (opening or closed) is carried on using the SVM classification algorithm with HOG features of the input eye image. Second, rough appearance-based gaze estimation is implemented based on a simple CNN model. And the head pose is estimated to judge whether the user is facing the screen or not. Based on these recognition results, noncontact mouse control and character input methods are designed and developed to replace the standard mouse and keyboard hardware. Accuracy and speed of the proposed interaction system are evaluated by four subjects. The experimental results show that users can use only a common monocular camera to achieve gaze estimation and tracking and to achieve most functions of real-time precise human-computer interaction on the basis of auto eye control.

1. Introduction

The wireless sensor network (WSN) consists of many sensors like visual sensors, thermal sensors, and various others. Sensor nodes are widely used for nonstop sensing, event detection, position sensing, and many other things, including helping the disabled with interfaces [1]. Capturing the eye tracking signals can help the disabled improve their quality of life by noncontact human-computer communicating with visual sensors; for example, this can be applied to the eye control wheelchair for the disabled [2]. Eye tracking refers to eye scanning, gaze, blinking, etc. And eye movement is closely related to brain activity. Therefore, the process of human learning and cognition can be studied through eye movement [3]. This is how the eye tracking technique was born. Eye tracking can obtain the focus of sight in real time and be applied to analyze the user's eye movement in reading [3] or in a critical state [4], so as to infer the user's content of interest in reality [5, 6]. Furthermore, it can also be used for the prediction and treatment of patients with brain neurological diseases [7, 8].

Eye-based human-computer interaction has a variety of other potential applications, especially on the WSN [9], such as smart home monitoring based on the Internet of Things [10]. Furthermore, sound users will accept eye control interaction as an additional means in the future [11], for instance, the combination of eye control interaction and virtual reality technology [12] and the small game platform with outdoor noncontact control by eye tracking technology [13].

Classified by devices, there are three mainstream interactive methods of eye tracking in recent years, including electrooculography [14], infrared fundus imaging [15], and appearance-based methods. Among them, electrooculography and infrared fundus imaging technology have been sufficiently mature. Appearance-based methods estimate the gaze by the shape and texture properties of the eye or the position of the pupil relative to canthus. These methods do not rely on good hardware

configuration, which makes them suitable for implementation on platforms without high-resolution cameras or additional light sources [16]. Methods for gaze estimation include methods based on feature points around the eyes [17], methods based on pupil position [18], and methods based on deep learning, such as CNN proposed in recent years [19].

In appearance-based researches, literature [20] trained the machine learning algorithm by combining the azimuth information with the geometric features of the eyes of which the angle error of the gaze tracking was about 4°. In literature [18], a pupil detection algorithm based on color intensity change was used to detect the center of the pupil in order to estimate the user's sight. Divided into 25 areas of the screen, the gaze estimation accuracy is 78%. In literature [21], 4 cameras with 5×5 pixel resolution were used to train the neural network for gaze tracking, and the minimum angle error was about 1.79°. Although the four cameras used are more complex to construct than a single webcam, the study reveals the great potential of the appearance-based approaches to gaze estimation using a DCNN. Literature [22] established a large data set of gaze tracking and achieved an average error of 10.5° in prediction across data sets by using a DCNN model. Literature [23] trained a CNN with images of both eyes and face posture, and the prediction errors of 1.71 cm and 2.53 cm can be achieved without calibration on the mobile phone and tablet terminals. Literature [24] used the CNN to predict the blinking behavior in nine directions and used it to control the nine-grid keyboard whose input speed could reach 20 letters per minute.

In summary, eye tracking has been widely used for decades in vision research, language, and usability. However, most prior research has focused on large desktop displays using specialized eye trackers that are expensive [25]. However, simple devices limit the resolution of images so that the accuracy of gaze estimation is often unsatisfactory, which makes it difficult to perform high-precise interactive operations. Thus, most research adopts interaction models that divide functions according to the roughly effective area of the screen, where the eye is gazing and uses specially designed interfaces. On account of the limited interactive function, these models face problems such as poor user experience, low operation efficiency, and low degree of freedom of interaction.

Therefore, this work designed a simple eye movement control human-computer interaction system, which can completely replace the mouse and keyboard hardware to operate the computer. This system is characterized by a high degree of freedom of interaction, and it can be carried out directly on the graphical user interface, such as the computer screen with a normal webcam.

2. Materials and Methods

2.1. Basic Methods

2.1.1. Eye Image Acquisition. The method of obtaining the user's eye image in this work is shown in Figure 1. First, each frame is extracted from the video read from the webcam, and the frame size is 640×480. For a single frame image, graying and flipping are performed. When the face is not recognized, face detection is performed with all frames. Then, the template matching method is used to track the face in the subsequent frames, marked with a tracking box.

Based on the cropped face image corresponding to the tracking box, 68 facial landmarks including eye corners, eyebrow, mouth center, and face contour are obtained by detecting the facial feature points [26]. Using the location of the eye, the two eyes' images can be cropped separately. Compared with the method of using the pupil to locate the eye [27], the positions of the four corners of the eye are determined based on the whole facial contour. So, eye tracking will not affect the stability of the eyes' positioning. This process is also not easily affected by the environment such as the illumination change. As a result, the recognition accuracy is improved.

Then, to enhance the eye image, an edge-preserving filter is used for denoising, and power transform is used to enhance the overall contrast of the region and weaken the influence of shadow and illumination change. Finally, two eye images with the size of 36×36 pixels are obtained. Both images will be used as the data source for the next processing.

2.1.2. Eye Movement Recognition and Gaze Tracking

(1) Eye State (Opening or Closed) Recognition. Eye state (opening or closing) recognition is used to reduce the gaze tracking error and as a means of interaction. The SVM (Support Vector Machine) classification algorithm with the HOG (Histogram of Oriented Gradient) features of the input eye image is used to classify the eye states: opening and closing, without distinguishing between the left eye and the right eye.

The parameters of the HOG are as follows: detection window (36, 36), cell size (18, 18), block size (18, 18), block step size (18, 18), and bin number of the histogram 9, and as a result, the feature descriptor of 36 dimensions is obtained. The SVM uses the Gaussian kernel function, and other parameters are default.

The recognition of the closed state and the duration need to know the approximate frame rate. For example, if 30 frames per second and 15 consecutive frames are classified as closed state, it means the duration is 0.5 seconds.

(2) Gaze Estimation and Tracking Based on CNN. First, a brief architecture based on CNN is established by reference to the iTracker model [23] for the estimation task, shown in Figure 2. It consists of a convolutional layer with 20 convolution cores with a size of 5×5 pixels, followed by a max-pooling layer with a size of 2×2 pixels and a concatenate layer followed by a final full-connection layer. The features of the two input images are fused through the concatenate layer, and the two outputs are obtained by the full connected layer, that is, the normalized coordinate value (x, y) of the predicted fixation on the screen.

2.1.3. Head Pose Estimation. Head pose estimation can be used to judge whether the user is facing the screen or not. It can also be used to control the page scrolling or operate the mouse direction. Using 68 facial feature points, the pitch angle, yaw angle, and roll angle can be estimated accurately [28]. However,

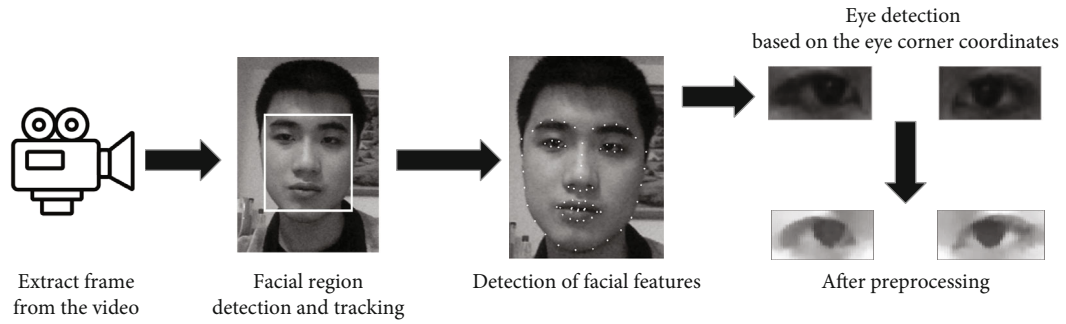

Figure 1: Eye image acquisition processing.

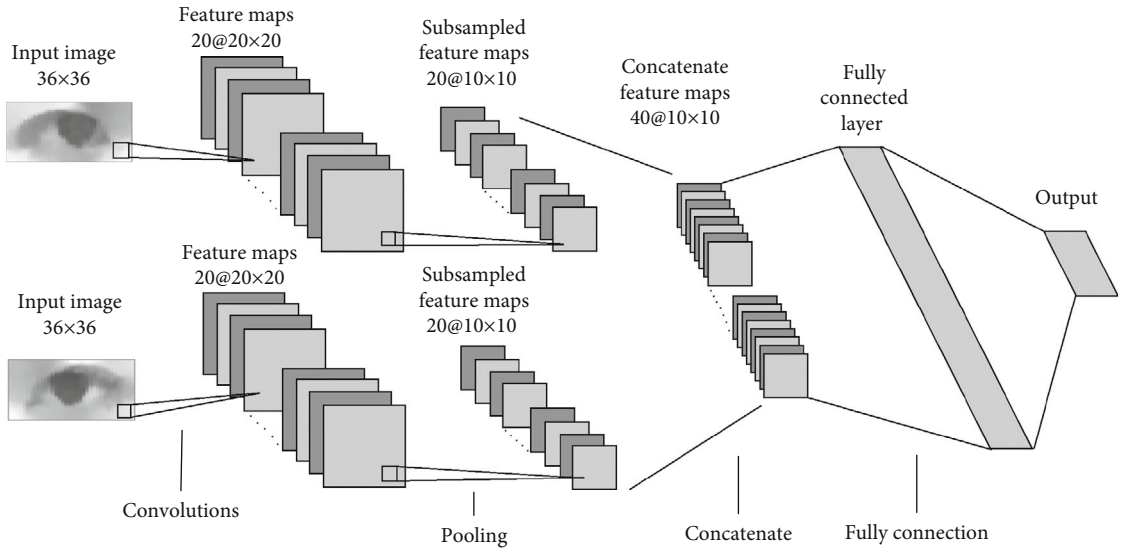

Figure 2: Architecture of the proposed CNN-based model for gaze estimation and tracking.

it only needs to roughly judge whether the user's head turns beyond a certain angle in this work. As shown in Figure 3(a), when the feature point P in the center of the nose is located in the area of the left quarter face, the head is considered to the left. Similarly, when the feature point P is located in the right quarter of the facial region, the head is considered to the right. As shown in Figure 3(b), the center of gravity of P1, P3, and P4 and the relative positions with P2 are calculated, and two thresholds are set to determine whether the head has an up or down deflection.

2.2. Design of the Interactive System

2.2.1. The System Operation Process. A problem of human-computer interaction through appearance-based gaze estimation is Midas contact [29]. The eyes not only act as an important sensory channel but also can provide motion response to control the computer [30]. Therefore, designing an appropriate identification process can greatly reduce the errors of false touch. The proposed system interaction process is shown in Figure 4.

In particular, in Step I, the interaction function should be suspended in the following situations: the user's face has left the screen, the face is too far or too close to the screen, and the face moves too fast. In Step II, when the head obviously deviates from the screen, the interaction function is suspended. In Step III, the SVM+HOG method is used to identify the eye state in each frame. If any one eye is classified as closed at least, the gaze estimation in Step IV will be not performed in this round.

2.2.2. Techniques for Noncontact Mouse Control. Assume that the user's actual point of gaze on the screen is called APOG and the user's predicted point of gaze on the screen is called PPOG. The position of the mouse on the screen is called POM.

Generally speaking, the user's gaze can be accurately calculated when using a high-precision camera; the distance between APOG and PPOG is very small. In this case, we only need to set the POM directly on the PPOG and a certain dwell time to trigger click, so as to complete the function of the hardware mouse. In recent years, there are also methods to detect EEG signals and visual dwell time to accurately distinguish the user's operation intention [31, 32]. In contrast, the result of appearance-based gaze estimation is rough. Although the user's APOG is stable, there may be large differences between the PPOG of adjacent frames, resulting in the user's inability to stably select

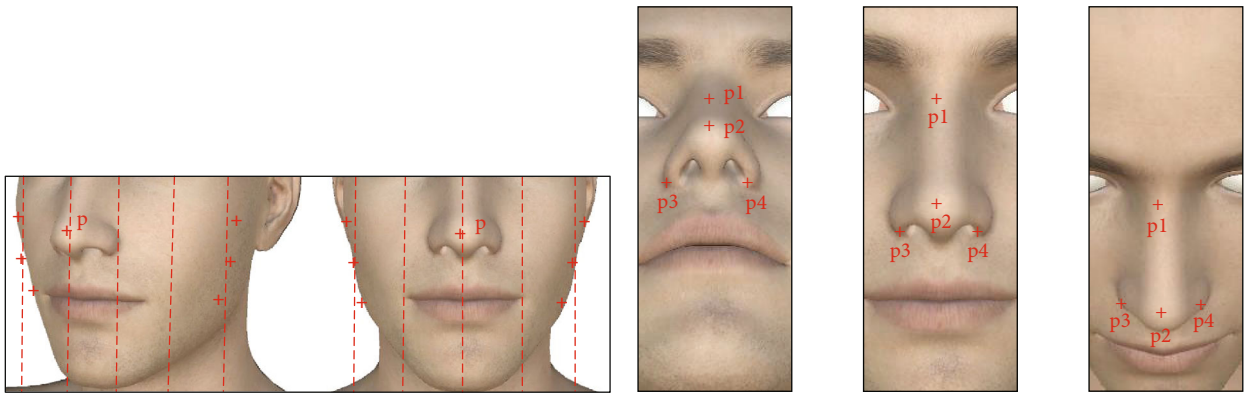

Figure 3: Diagram of head posture estimation: (a) lateral rotation direction estimation; (b) longitudinal rotation direction estimation.

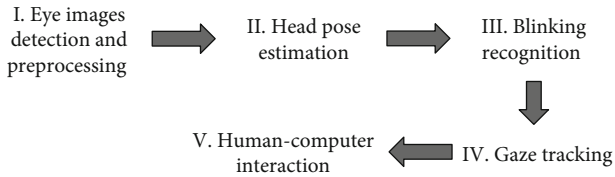

Figure 4: The system operation process.

the target on the screen. This is the biggest challenge of using appearance-based gaze estimation for human-computer interaction.

This work is aimed at designing a rule of accurate mouse movements, which allows the user to control PPOG by adjusting APOG, and then indirectly controlling the POM through this rule to achieve accurate and stable control. The general idea is, first of all, to use the mouse with high sensitivity to let the user use the gaze to control the mouse to move near the target location. Then, the user can move the mouse slowly to the top of the target with the gaze. Finally, one eye blinking is used to trigger the click to complete a round of targets from selection to click operation. There are three ways to control the mouse movement based on gaze estimation; the scheme is described in detail as follows.

The distance of mouse movement is determined by D_i $(i = 1, 2, 3)$, and the calculation method is shown in

$$D_i = L_i \times W_i, \tag{1}$$

where L_i refers to the reference distance of mouse movement, and different mouse movement modes have different meanings. W_i is the sensitivity of mouse movement.

(1) Moving Based on Gaze Estimation Named Gaze Movement. A moment before moving the mouse in a frame of the video stream is shown in Figure 5. The circle in the figure represents the PPOG, and the distance between the mouse and the PPOG is expressed by L_1. Finally, the mouse will move with distance D_1 in the direction of PPOG.

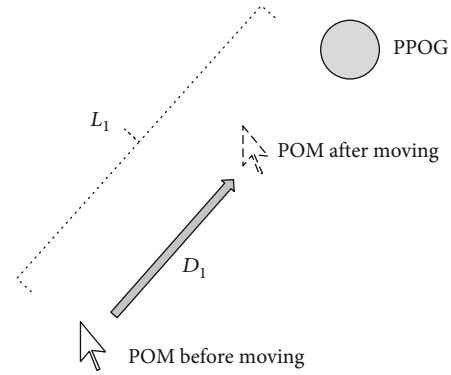

Figure 5: Moving sketching based on gaze estimation.

Gaze movement is a kind of movement mode in which the mouse sensitivity W_1 will gradually decrease. It is used to manipulate the mouse to reach or near the target to click. Ideally, at the beginning of the mouse position reset, the mouse will follow the PPOG. Then, the sensitivity W_1 decreases gradually, the mouse will still move to the PPOG, but the moving speed is slower and slower. When W_1 drops to 0, POM is fixed and cannot move anymore. At this time, if POM is already upon the target, you can click the target by blinking. If there is still some distance away from the target, and it will be difficult to move the POM through gaze movement, the following two techniques can be used to fine-tune the POM. The user can choose one of the following two methods according to their habits to continue to adjust the POM.

(2) Moving Relative to the Screen Center. This is a method to adjust the POM slightly, and sensitivity W_2 is fixed to a relatively small value. As shown in Figure 6, it calculates the distance L_2 between the screen center coordinate and the PPOM, and the mouse moves with the distance D_2 in the direction parallel to the PPOM.

(3) Moving with Remote Control. This is also a way to adjust the POM slightly. Sensitivity W_3 is also fixed to a relatively small value. The mouse can only move in four directions

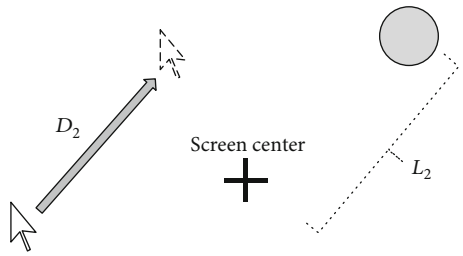

Figure 6: The diagram of moving relative to the screen center.

in this way, as shown in Figure 7; the screen is divided into four areas: up, down, left, and right. The mouse moves to the corresponding gaze area. In the figure, the PPOG appears in the area above the screen, and the distance between the PPOG and the Y axis of the screen center is expressed by L_3, so the mouse moves up along D_3.

2.2.3. Character Input

(1) Input by a Virtual Keyboard. A translucent virtual keyboard will be displayed on the screen while typing in a text box, as shown in Figure 8. This function is identical to that of the hardware keyboard. The user can click the keys on the virtual keyboard by the proposed noncontact mouse controls (see Section 2.2.2). Each key is placed in a square of about 1.5 cm $*$ 1.5 cm, and the entering speed is related to the user's proficiency in clicking a 1.5 cm $*$ 1.5 cm target. In addition, the "FN" key in the lower left corner of the keyboard is designed as a combination key input mode. Users need to click once at the beginning and end of the input combination key. For example, when inputting the capital letter "a" with "shift+a," the order of clicking is "FN," "shift," "a," "FN."

(2) Input by the Gaze Tracking Coding. This method implements character input through trajectories of different eye movements and the dwell time [33, 34]. The specific proposed strategy is described as follows.

The screen is divided into four regions as shown in Figure 9; each region corresponds to a number, namely, "1," "2," "3," and "4." When the user's gaze stays in one of the regions for a period of time, the corresponding number is entered. Every combination of some or all of these four numbers represents one character on the keyboard. For example, the combination "123" expresses the letter "a." The user's gaze only needs to stay in Regions (1), (2), and (3) for one second, respectively, that is, at least 3 seconds to finish the letter "a" input. Of course, the user can shorten the required gazing dwell time for each number (symbol) by multiple training so as to speed up the typing.

To cover all keys on the keyboard in this way, it is necessary to design a coding algorithm with the numbers "1," "2," "3," and "4." We use an idea similar to the Quad Huffman tree to encode for each key on the normal keyboard. The code of any key will not be the prefix of other keys' codes. Adjust the length of each code according to the utilization frequency of the key; the complete coding result is shown in Figure 10.

In practical application, the input process will be displayed in real time on the screen. Take the letter "H" input as an example, the process and the contents of the screen prompt are shown in Figure 11.

It should be noted that the logic of the character input designed in this paper is the same as the hardware keyboard; users can use "shift" and "caps lock" to enter different characters in the same key. Then, two ways are designed for the combination key input: (1) For general two-key combination, add several special codes of commonly used keys, such as shift, alt, and ctrl. Specifically, input a common key code after a special key code input. That is, the double-key combination command is executed. (2) For key combination of not fewer than two keys, first, input a special code to represent multikey combination, and then, put the ordinary key in a queue. You should enter the special code again to end the input; meanwhile, all keys in the queue are executed as key combinations.

2.2.4. Triggering Methods of Other Interactive Commands

(1) Application of Eye State Recognition.

(1) Left single closed for 0.5 s: click with the left mouse button

(2) Left single closed for 1 s: double-click with the left mouse button

(3) Right single closed for 0.5 s: right click

(4) Closed for 1 s: reset the mouse position when the mouse is moving

(5) Blinking eyes three times in 5 seconds: turn on the keyboard or turn off the keyboard

(2) Application of Head Pose Recognition.

(1) Head up: roll up

(2) Head down: roll down

(3) Head left: delete characters on the character input mode

(4) Head right: turn the eye control on or off

3. Results and Discussion

3.1. Calibration. As shown in Figure 12, a laptop with a front camera or a desktop with an external USB camera is used as the experimental equipment; the camera resolution should be 640 $\times$ 480 and above. The user is facing the screen, about half a meter away from the camera.

Before the interaction, we need to record the face feature information when the user is facing the computer screen, the gaze tracking training model, and the eye state (opening or closed) training model. The process of collecting this information and training is called calibration. In order to improve the

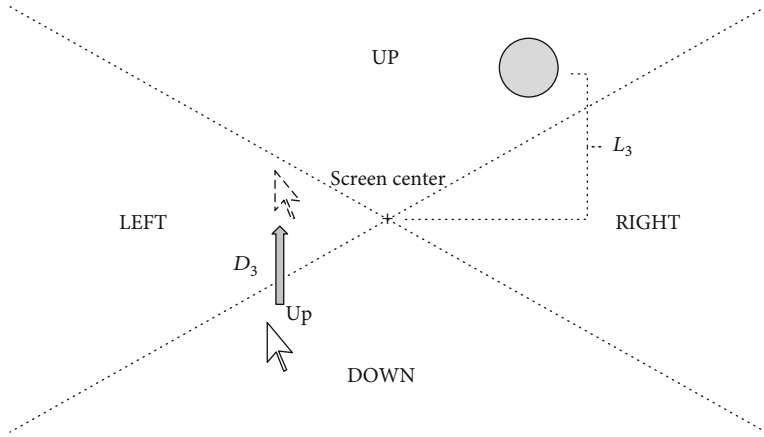

Figure 7: The diagram of moving with remote control.

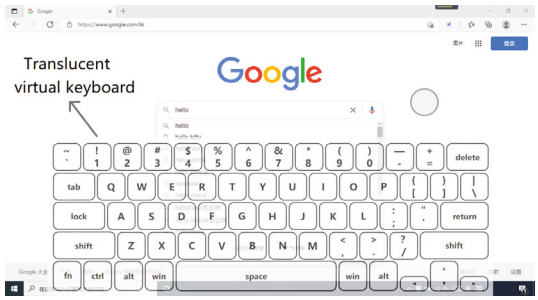

Figure 8: The proposed virtual keyboard.

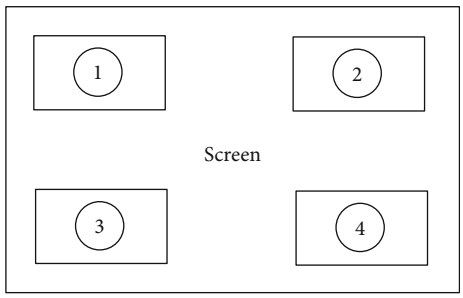

Figure 9: Valid input regions of the screen.

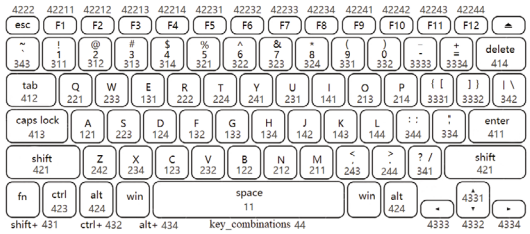

Figure 10: Coding results of each key on the normal keyboard.

recognition accuracy, the user should spend a few minutes completing the calibration step before interaction. The calibration function of the program is divided into three steps.

Step 1. The facial feature points are collected after the user faces the screen for three seconds to estimate the user's head pose.

Step 2. Collect the training data of gaze tracking. The mouse cursor moves according to similar tracks in Figure 13. The number represents the operation order, and the arrow direction is the direction of the mouse movement. The cursor can be controlled by the program or manually according to the actual situation. During the acquisition, the user should avoid head movements, blinking, and other behaviors. When the mouse moves, the program collects the user's eye image and the mouse coordinates as a training sample. After collection, all images are preprocessed to deal with the impact of the environment, and the coordinates of the mouse are normalized to achieve screen size adaptation. All samples were then sent to training; in general, 700-1000 samples are required to train a gaze tracking CNN-based model that meets the requirements. The training model does not need to be retrained when the appearance of the user's eye, the experimental environment, and equipment are not changed much.

More than 500 samples were collected according to the above moving trajectory, and the tracking effect was tested after calibration. The subjects were asked to look at 12 circular targets on the screen in turn. When the subjects looked at each circular target, 10 PPOGs of continuous gaze estimation were recorded. Results are shown in Figure 14; the average error between the PPOG and the APOG is about 2 cm~3 cm.

Step 3. The user closes his eyes for three seconds to collect his/her closed eye images and to train the open and closed eyes binary classification model of SVM with the HOG features of the user's opening eye image collected in the previous step.

The experimental results show that the classification accuracy of a single image is about 90% when the sample size reaches 2000. And this accuracy meets the requirements, because 15 consecutive images are classified as a closed eye state to trigger the interactive command, and the probability of false touch is very low.

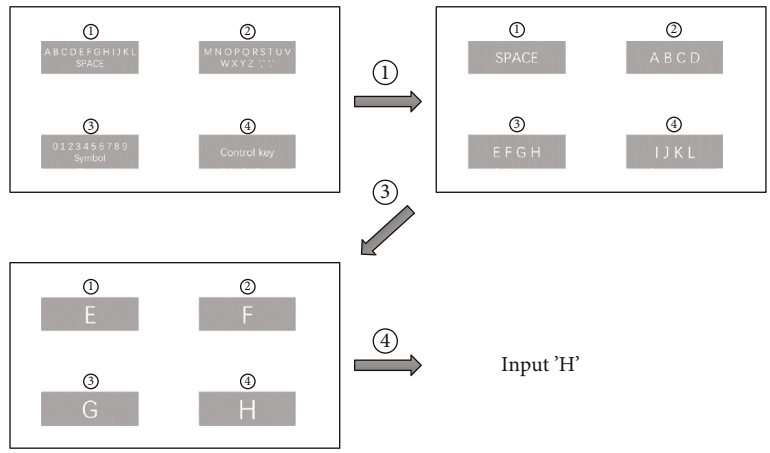

Figure 11: The process and the screen prompt for the input of letter "H."

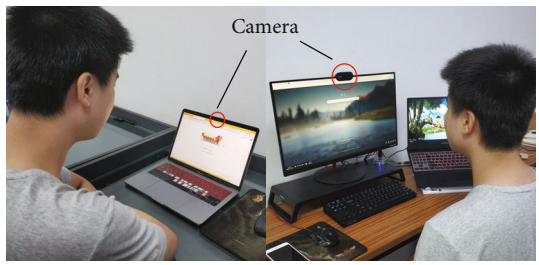

Figure 12: The optional experimental equipment.

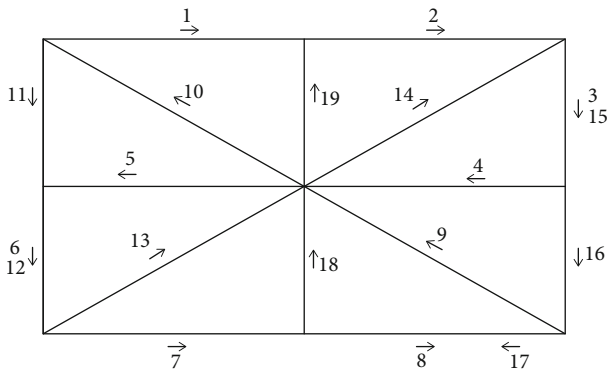

Figure 13: An example of gaze data calibration.

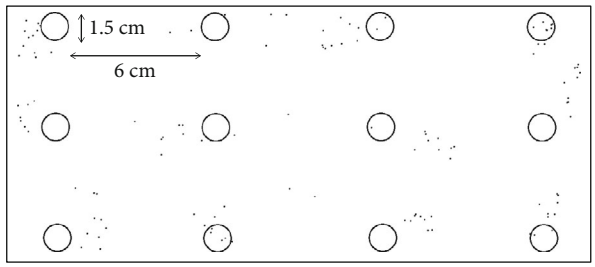

Figure 14: Calibrated gaze data.

3.2. Training for the Interaction. Four 20-year-old college students were selected to participate in the test. None of the four users had been exposed to eye tracking-based interaction, and they need to be trained before the formal test. The training includes click, moving mouse, and typing through auto eye control. After explaining how to operate and what should be paid attention to, the training program began.

Take the training of controlling the mouse based on gaze estimation as an example: the training is divided into three stages; the first stage is to be familiar with the methods (see Section 3.1) of the cursor's operation. In the second stage, the program randomly generates blocks with different positions and sizes on the screen. The user needs to control the cursor to move to the specified block. When the cursor touches the block, it disappears and the next block is generated. In the third stage, blocks are also generated. The difference is that the user needs to blink and click correctly before the next block is generated. Each subject was trained for 20 minutes a day on average, and the next test could be carried out after training for more than three days.

3.3. Interoperability Test of the Proposed Interaction System

3.3.1. Test for the Noncontact Mouse Control. As shown in Figure 15, place 25 blocks of the same length and width on the screen with the same intervals as the targets. The subject needs to use the methods described above to control the mouse with eye movements to click on these targets. Squares with two different sizes are prepared, which are 2 cm × 1 cm and 1 cm × 0.5 cm. Four subjects were asked to select each square line by line to click; the time for every subject was recorded. If one square is clicked correctly, the subject can continue to click the next one in sequence. If all the squares are clicked correctly, you will stop counting time. Each size of the squares is tested three times; that is, 75 correct clicks are required. The experimental result recording after three times is shown in Tables 1 and 2. The highest and lowest accuracy on the smaller click square is 100% and 80%, respectively. And the slowest tester spent 10.85 seconds for one click. It can be seen for the two tables, on

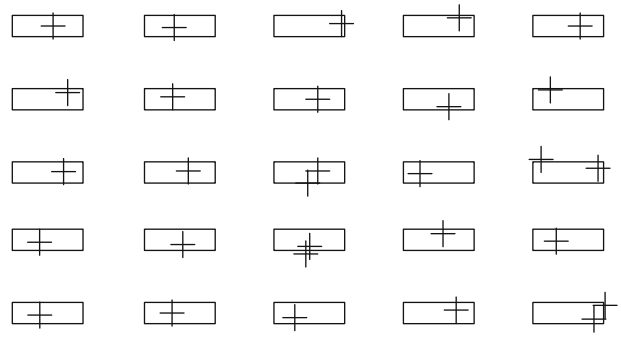

Figure 15: A target click test design.

Table 1: Test results of the proposed mouse click in the square (2 cm × 1 cm).

Subject	Total time (s)	Accuracy (%)	Speed (s per click)
1	417	100.0	5.56
2	556	89.3	7.41
3	489	97.5	6.52
4	507	93.3	6.76

Table 2: Test results of the proposed mouse click in the square (1 cm × 0.5 cm).

Subject	Total time (s)	Accuracy (%)	Speed (s per click)
1	585	100.0	7.80
2	760	80.0	10.13
3	639	96.0	8.52
4	814	90.6	10.85

the square (2 cm × 1 cm), that the speed of the noncontact mouse click basically meets the real-time requirement.

3.3.2. Character Input Test. Let the user use the two proposed typing methods (see Section 3.2) to type a sentence with 56 characters including spaces. If the input character is wrong, you need to delete the character and reenter it. The testing results are shown in Tables 3 and 4.

3.4. Result Analysis and Discussion. According to the experimental results, the average distance between CNN's PPOG and the user's APOG is 2.5 cm after calibration. It can be seen that the average error of the model is 3.18° when the distance is 45 cm between the user and the screen. In fact, compared with other appearance-based gaze estimation models [35], the performance of this result is not the best. However, this just shows the universality of the proposed interaction mode in the upper application of other appearance-based gaze estimation models. For example, combined with the proposed interaction rules, the model [23], which does not need additional calibration with large error, can still achieve accurate human-computer interaction applications.

The results of the interaction test show that the overall accuracy of the eye control mouse is 93.33%, and the click speed is 7.9 seconds. In terms of typing, the overall accuracy

Table 3: Results of character input test based on the virtual keyboard.

Subject	Total time (s)	Accuracy (%)	Speed (s per character)
1	454	98.2	8.10
2	529	91.0	9.44
3	468	94.6	8.35
4	480	96.4	8.57

Table 4: Results of character input test based on the character coding.

Subject	Total time (s)	Accuracy (%)	Speed (s per character)
1	384	98.2	6.85
2	443	96.4	7.91
3	428	98.2	7.64
4	422	94.6	7.53

of the virtual keyboard is 95.05%, and the speed is 8.6 seconds per character. The overall accuracy of characters input based on gaze tracking is 96.85%, and the speed is 7.48 seconds per character.

Compared to several appearance-based eye tracking methods proposed in recent years, this work has the advantages of flexible interaction and complete interaction functions, etc. The specific comparison results are shown in Table 5.

The experimental results also show that the accuracy and speed largely depend on subjects' proficiency. For example, Test No. 1 practiced for 3 hours before the test, which was 2 hours more than other subjects, so he got the best results. Basically, the subject can use the three mouse movement methods to make the mouse click anywhere on the screen. But it will take more time to click near the edge of the screen than at the center.

In typing, compared with the virtual keyboard, the character input method based on eye tracking coding was faster. But the character input speed still cannot fully meet the requirements of real-time interaction. A feasible way to speed up typing is to introduce animation effect into the input process to guide the user to quickly and accurately move the line of sight to enter characters.

Furthermore, the results show that the size of the screen has an impact on interaction efficiency. Generally speaking, a large screen has a wide range of gaze estimation, and it has advantages in the accuracy and speed of character input. However, it may take more time to challenge the final landing point of the mouse if you want to select a target. Therefore, the screen size should not be too large or too small. Our experiments indicated that the appropriate size is 14 to 19 inches.

In addition, it should be noted that subjects who use eye tracking interaction intensively in a short period of time will be tired. Most of the subjects experienced fatigue about 10 minutes after using the system for the first time. And they developed dizziness, nausea, and other symptoms 15 minutes later. They had to put off the test for a day. After a variety of

Table 5: Comparison results.

Literatures	How to interact	Key methods	Accuracy	Speed	Comparison conclusions
[18]	The screen is divided into 3×3 or 5×5, which can monitor the user's gaze on the screen in real time and can be used as 9 or 25 interactive channels	Improved pupil detection algorithm based on color intensity	3×3: 94% 5×5: 78%	In real time	It had poor accuracy and insufficient flexibility for practical applications.
[24]	Use the nine key inputs widely applied in straight board mobile phones	Using CNN to estimate the gaze in nine directions, representing nine keys	90%	20 letters per minute for a skilled user	It was twice as fast compared to our work, but it had insufficient flexibility and limited functions.
[26]	Move the mouse through the head posture, and control the mouse switch, click and scroll through blinking and other facial expressions	Head posture estimation, face detection, and expression recognition	Not mentioned	Not mentioned	It requires various operations to adjust the position of the head when moving the mouse, which reduces the interaction efficiency and usability.

test training, the intensive training can last for 20 minutes without fatigue.

This work also arranged the daily testing task with the proposed human-computer interaction system, including browsing the web and watching videos. All the subjects were able to complete the tasks successfully.

4. Conclusions

In order to achieve simple, affordable, and accurate auto eye control human-computer interaction application, this work applied a webcam to obtain eye image and a lightweight CNN to complete the appearance-based gaze estimation. Using the results of gaze estimation, eye control mouse and character input methods are designed to replace the traditional human-computer interaction devices such as the mouse and keyboard hardware to complete the accurate interactive operation with the computer. The benefits of our method are as follows: (1) the proposed human-computer interaction system enables ordinary people to operate the computer when both hands are occupied. (2) It can also effectively help people with behavioral disorders to solve the input dilemma in human-computer interaction. (3) It realizes the goal of accurate interaction with computers by using imprecise gaze tracking. However, if a completely unfamiliar user uses this system, he needs to spend time to understand the principles of the proposed interaction methods and master the skills before he can skillfully achieve various interactive operations.

Although the interactive system in this work has a certain threshold, it is difficult to meet the requirements of real-time interaction for a long time. However, simple equipment requirements make the proposed interactive system easy to popularize, especially for some people with behavioral disorders with limited living conditions. An ordinary laptop with a camera allows them to complete simple interactive tasks such as browsing videos and information like ordinary people in their spare time.

In the future, we will study the optimization accurate eye positioning method for the user wearing glasses to extend the system to a more general situation. For people with behavioral disorders, the setting and interactive operation of various commands need to be in line with their habits to make the system more accessible. Therefore, developing an interactive system for people with behavioral disorders is also the follow-up task of this study. Finally, the research results can be extended to regulate driving behavior and more noncontact interactive control applications.

Data Availability

All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest

There are no conflicts of interest.

Acknowledgments

This work is supported by grants from the Zhuhai Basic and Applied Basic Research Foundation (ZH22017003200027PWC).

References

[1] O. Palagin, V. Romanov, I. Galelyuka, V. Hrusha, and O. Voronenko, "Wireless smart biosensor for sensor networks in ecological monitoring," in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 679–683, Bucharest, Romania, 2017.

[2] M. Dahmani, M. E. H. Chowdhury, A. Khandakar et al., "An intelligent and low-cost eye-tracking system for motorized wheelchair control," *Sensors*, vol. 20, no. 14, article 3936, 2020.

[3] N. Dirix, H. Vander Beken, E. de Bruyne, M. Brysbaert, and W. Duyck, "Reading text when studying in a second language: an eye-tracking study," *Reading Research Quarterly*, vol. 55, no. 3, pp. 371–397, 2020.

[4] F. Guo, M. Li, Y. Chen, J. Xiong, and J. Lee, "Effects of highway landscapes on drivers' eye movement behavior and emergency reaction time: a driving simulator study," *Journal of Advanced Transportation*, vol. 2019, 9 pages, 2019.

[5] Z. Li, S. Zhao, L. Su, and C. Liao, "Study on the operators' attention of different areas in university laboratories based on eye movement tracking technology," *IEEE Access*, vol. 8, pp. 8262–8267, 2020.

[6] Y. Tamura and K. Takemura, "Estimating focused object using smooth pursuit eye movements and interest points in the real world," *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 21–23, 2019.

[7] H. Y. Lai, G. Saavedra-Pena, C. G. Sodini, V. Sze, and T. Heldt, "Measuring saccade latency using smartphone cameras," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 3, pp. 885–897, 2020.

[8] J. Rapela, T. Y. Lin, M. Westerfield, T. P. Jung, and J. Townsend, "Assisting autistic children with wireless EOG technology," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3504–3506, San Diego, CA, USA, 2012.

[9] M. Khamis, F. Alt, and A. Bulling, "The past, present, and future of gaze-enabled handheld mobile devices: survey and lessons learned," in *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 1–17, Barcelona, Spain, 2018.

[10] A. Bissoli, D. Lavino-Junior, M. Sime, L. Encarnação, and T. Bastos-Filho, "A human–machine interface based on eye tracking for controlling and monitoring a smart home using the internet of things," *Sensors*, vol. 19, no. 4, p. 859, 2019.

[11] R. Jacob and S. Stellmach, "What you look at is what you get," *Interactions*, vol. 23, no. 5, pp. 62–65, 2016.

[12] V. Clay, P. König, and S. Koenig, "Eye tracking in virtual reality," *Journal of Eye Movement Research*, vol. 12, no. 1, pp. 1–18, 2019.

[13] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Pursuit calibration: making gaze calibration less tedious and more flexible," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 261–270, St. Andrews Scotland, United Kingdom, 2013.

[14] W. D. Chang, "Electrooculograms for human–computer interaction: a review," *Sensors*, vol. 19, no. 12, article 2690, 2019.

[15] S. Bi, Y. Gu, J. Zou, L. Wang, C. Zhai, and M. Gong, "High precision optical tracking system based on near infrared trinocular stereo vision," *Sensors*, vol. 21, no. 7, p. 2528, 2021.

[16] A. Kar and P. Corcoran, "A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms," *IEEE Access*, vol. 5, pp. 16495–16519, 2017.

[17] Y. L. Wu, C. T. Yeh, W. C. Hung, and C. Y. Tang, "Gaze direction estimation using support vector machine with active appearance model," *Multimedia Tools and Applications*, vol. 70, no. 3, pp. 2037–2062, 2014.

[18] C. Zheng and T. Usagawa, "A rapid webcam-based eye tracking method for human computer interaction," in *2018 International Conference on Control, Automation and Information Sciences*, pp. 133–136, Hangzhou, China, 2018.

[19] A. A. Akinyelu and P. Blignaut, "Convolutional neural network-based methods for eye gaze estimation: a survey," *IEEE Access*, vol. 8, pp. 142581–142605, 2020.

[20] D. Cazzato, F. Dominio, R. Manduchi, and S. M. Castro, "Real-time gaze estimation via pupil center tracking," *Paladyn, Journal of Behavioral Robotics*, vol. 9, no. 1, pp. 6–18, 2018.

[21] M. Tonsen, J. Steil, Y. Sugano, and A. Bulling, "InvisibleEye," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–21, 2017.

[22] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Mpiigaze: real-world dataset and deep appearance-based gaze estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 162–175, 2019.

[23] K. Krafka, A. Khosla, P. Kellnhofer et al., "Eye tracking for everyone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2176–2184, Las Vegas, NV, USA, 2016.

[24] C. Zhang, R. Yao, and J. Cai, "Efficient eye typing with 9-direction gaze estimation," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19679–19696, 2018.

[25] N. Valliappan, N. Dai, E. Steinberg et al., "Accelerating eye movement research via accurate and affordable smartphone eye tracking," *Nature Communications*, vol. 11, no. 1, pp. 1–12, 2020.

[26] K. Meena, M. Kumar, and M. Jangra, "Controlling mouse motions using eye tracking using computer vision," in *2020 4th International Conference on Intelligent Computing and Control Systems*, pp. 1001–1005, Madurai, India, 2020.

[27] J. Xiahou, H. He, K. Wei, and Y. She, "Integrated approach of dynamic human eye movement recognition and tracking in real time," in *2016 International Conference on Virtual Reality and Visualization*, pp. 94–101, Hangzhou, China, 2016.

[28] P. B. M. Thomas, T. Baltrušaitis, P. Robinson, and A. J. Vivian, "The Cambridge face tracker: accurate, low cost measurement of head posture using computer vision and face recognition software," *Translational Vision Science & Technology*, vol. 5, no. 5, pp. 1–7, 2016.

[29] R. Menges, C. Kumar, K. Sengupta, and S. Staab, "eyegui: a novel framework for eye-controlled user interfaces," in *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, pp. 1–6, Gothenburg Sweden, 2016.

[30] C. Kumar, R. Menges, and S. Staab, "Eye-controlled interfaces for multimedia interaction," *IEEE Multimedia*, vol. 23, no. 4, pp. 6–13, 2016.

[31] S. Han, R. Liu, C. Zhu et al., "Development of a human computer interaction system based on multi-modal gaze tracking methods," in *2016 IEEE International Conference on Robotics and Biomimetics*, pp. 1894–1899, Qingdao, China, 2016.

[32] M. Zhao, H. Gao, W. Wang, and J. Qu, "Research on human-computer interaction intention recognition based on EEG and eye movement," *IEEE Access*, vol. 8, pp. 145824–145832, 2020.

[33] P. P. Banik, M. K. Azam, C. Mondal, and M. A. Rahman, "Single channel electrooculography based human-computer interface for physically disabled persons," in *2015 International Conference on Electrical Engineering and Information Communication Technology*, pp. 1–6, Savar, Bangladesh, 2015.

[34] M. Yildiz and H. Ö. Ülkütaş, "A new PC-based text entry system based on EOG coding," *Advances in Human-Computer Interaction*, vol. 2018, 8 pages, 2018.

[35] Y. Cheng, H. Wang, Y. Bao, and F. Lu, "Appearance-based gaze estimation with deep learning: a review and benchmark," pp. 1–21, 2021, https://arxiv.org/abs/2104.12668.