Web Cam Based Moving Object Detection

[1]Mrs.M.Karpagam , [2]Ms.P.Babychitra , [3]Ms.R.Loheswari , [4]Mrs.K.Swathi
[1]*AP/CSE dept, Mangayarkarasi College of engineering*, Madurai, Tamilnadu.
[2,3,4]*CSE Dept, Mangayarkarasi College of engineering*, Madurai, Tamilnadu.
.

***Abstract*—**
This examination proposes a quick eye following technique, to react to a circumstance that require a high preparing pace yet less precision. In contrast to different examinations, this investigation utilizes a webcam with a low goals of 640 × 480, which diminished the expense of gadgets extensively. We additionally built up the comparing calculation to suit the low-quality picture. We utilize a productive calculation to recognize the pupils which depends on shading power change to diminish the computation load. The preparing speed surpasses the prerequisite of eye following for saccade eyeball development. The aftereffect of trial demonstrates that the proposed technique is a quick and minimal effort strategy for eye following

Indexterms: *eye tracking, gaze estimation, webcam, OpenCV*

I. INTRODUCTION

Eye tracking is the process of estimating either the point of gaze (where the subject is looking) or the motion of the eye in relation to the head. It has become popular R PAPER BEFORE STYLING Before you begin to format your paper, first write and with the development of augmented reality (AR) and virtual reality (VR) in recent years. However, to adapt these two technologies, researchers are focusing on accurate eye tracking, which require high-quality cameras, gyroscopes and some other specialized devices. It causes the cost of eye tracking technology to remain high. Moreover, due to some other issues such as computation load and convenience, eye tracking technology is still not applied generally. We do not really need a very high accuracy of eye tracking for many situations which are possible to apply in reality, such as the analysis of reading behavior on web pages. For this situation, it is not necessary to know which word the user is reading. It will be very helpful if we have a light-weight eye tracking system which can detect which section was read and how long it took, these are enough to help us analyze the reading behavior of users. Based on the situation described, we propose a rapid webcam-based eye tracking method which is light-weight, low-cost and convenient, the system will be executed

II. EYEBALL MOVEMENT AND SPEED

There are mainly two ways of eyeball movement when human's gaze is moving. One is smooth movement, during this movement, humans can pay attention to track the moving objects, and the maximum speed of smooth movement is 30 degrees/s [4].

Taking a traditional eye tracking application scenario as an example, to track user's gaze on the display screen. The angle from the middle of the display screen to the side is about 30 degrees, it will take a total of 1 second under smooth movement. Therefore, the eye tracking system needs a speed which is higher than 1 frame/s to capture the gaze. The second eyeball movement is saccade movement, humans will ignore the pictures in the process of moving, attention will be shifted in another direction in a very short time. The speed of saccade movement is very fast, which is about 400~600 degrees/s [4]. At a maximum speed of 600 degrees/s, it will take 0.05 seconds for 30 degrees, but in this case, the visual system of human being needs time to get out of this kind of state of ignoring, the required time will be 0.17 seconds, then the user will pay enough attention on the target. Therefore, the total time taken is 0.22 seconds. In this situation, eye tracking system needs a speed which is

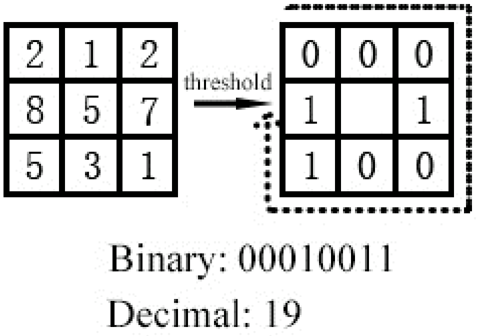
III. RAPID FACE DETECTION AND EYE DETECTION

On the subject of face detection, the detection methods based on Boosting algorithm have obvious speed advantage [5]. Therefore, we chose the AdaBoost-based cascaded classifier which is contained in OpenCV to get a high processing speed.

Local binary pattern (LBP) feature [6] is a commonly used detection operator in image analysis. LBP operator processes a local 3 × 3 pixels region, by comparing the central pixel of the region with the surrounding 8 pixels. Set the value on corresponding pixel to 1 if the previous value is greater than the center's, otherwise set it to 0, and then the binary numbers on the compared 8 pixel points are arranged in sequence to form an 8-bit binary number. This 8-bit binary number is the LBP value of the center pixel.

One calculation example is shown in Fig. 1.
A more formal description of the LBP operator can be given as (1).

(*xc*, *yc*) is the central pixel with intensity *ic*, and *ip* is the intensity of the *p*th neighbor pixel, *s*(*x*) is the sign function defined as (2).



LBP feature can be used in face detection. The face is divided into N rectangular regions. The image is processed using LBP transform to obtain the LBP histogram for each region. The N histograms are stitched into a new histogram. The features of the human face are represented by the features of the histogram. Based on LBP feature, we can use Boosting learning algorithm to obtain a fast classifier for face detection. In this study, we used the LBP-based cascade classifier which is contained in OpenCV, it was trained by AdaBoost algorithm.

There is another kind of classifiers which are also contained in OpenCV for face detection, they are trained based on Haar-like feature. Comparing with Haar-like feature-based classifier, LBP feature-based classifier has the advantages which are shown below.
1) The calculation load is lower, which makes LBP features-based classifier that has a low requirement for devices.
2) LBP feature-based classifier is light-weight lows.
3) The speed of detection using LBP features-based classifier is higher.
In a study [7] to compare LBP and Haar-like features for face detection, the author compared the performance of two features in face detection, the result showed that there is no big gap on accuracy between LBP feature and Haar-like feature, but for speed, LBP feature-based classifier has a significant advantage. Based on this study, we chose LBP feature-based classifier for face detection.
Once a face is detected, clip the part of face from the original image for further processing.
For eye detection, we also use the corresponding classifier contained in OpenCV. To decrease the calculation load, we limit features, such as the height of eyes and the distance between eyes. After eye detection, clip the part of eyes from the original image to decrease the calculation load in further processing. features, such as the height of eyes and the distance between eyes. After eye detection, clip the part of eyes from the original image to decrease the calculation load in further processing.

IV. PUPIL DETECTION

After completing face and eye detection, we need to locate the pupil to estimate the user's gaze.

There are many ways to locate the pupil, including machine learning and methods based on geometry features. Machine learning is still too heavy in many situations, which is not suitable for general application. It will be more efficient to calculate the center of pupil through geometry features. A traditional pupil detection method based on geometry features is Hough transform, which detects the pupil by detecting a circle from the image of the eyes, it is an effective method for locating the pupil, but it is more suitable in a situation where the distance between the camera and eyes is short, in [3], the distance is only about 30 to 50 mm. Moreover, it requires a high image quality. In this project, we use a webcam with a low resolution of 640 × 480, and the distance between the webcam and the eyes is about 300 to 500 mm. We used Hough transform to locate pupils, but the effect was not good. In the experiment, there was an incomplete pupil exposure, and due to low-quality images, the confidence levels of the curves on the image may be competitive, for the curve of pupils, it may not have a sufficient confidence level to exceed others', which caused wrong results.

In this study, due to the limitation of devices and speed requirement, the negligible errors are acceptable. Based on the situation, we propose a method for locating the pupil by checking the color intensity change on the image of eyes. The specific processing steps are as follows.
1) Locating the eyes using the cascade classifier contained in OpenCV and clip the part of eyes from the original image to decrease the calculation load in further processing.
2) Smoothing the image of the eyes using bilateral filter. Due to the low-quality image, there are many noisy points on the images, and also because of the light intensity and angle, the color intensity on the pupil is not stable. Therefore, it is necessary to smooth the image first for further processing which is checking the color intensity change on the image of eyes. Bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. The calculation load of bilateral smoothing is comparatively high, but after two times of image segmentation and due to the low-resolution image from the webcam, the resolution

of image is very low, which makes the calculation load in this step acceptable.

3) From the center point of the image of the eye, verify the color intensity horizontally, and locate the dark part on the horizontal line which is closest to the center, calculate the average of intensity on the dark line. As illustrated in Fig. 2 and Fig. 3.

4) From the first pixel on the dark line, verify the surrounding pixels to get all the pixels which have similar color intensity to the average of the dark line. Repeat this process to detect all the continuation of dark pixels.

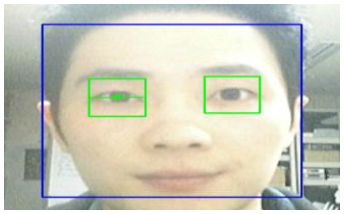


Fig. 2. The original image in the step of checking color intensity

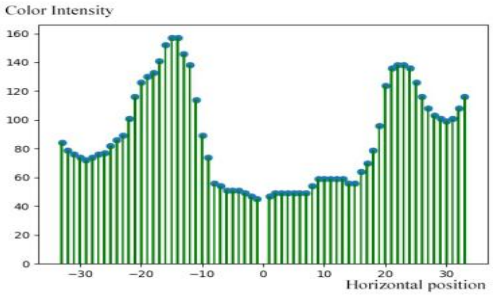


Fig. 3. The color intensity change of the left eye

The dark block formed by the dark pixels is the result of pupil detection.

5) Calculate the center of the dark block, which is the approximate center point of the pupil.

For the accuracy of pupil detection, it may depend on multiple factors, such as the light intensity, light angle, size of the eyes and extent of the pupil area exposed. The positioning error on the center of pupil in this method is very small, which can work well in most of the situations and is sufficient for the rough gaze estimation in this study. And it can effectively avoid the erroneous results caused by the bright spots on the pupils.

V. GAZE ESTIMATION

There are two kinds of eye movements when a human being moves his or her gaze. First, when the angle change is small, only the eyeballs will move and the head will remain motionless. The second is that the eyeballs and the head will move together when the angle change is large.

There is an angle threshold for switching between the two movement modes, and this threshold is slightly different for each person, between about 15 degrees and 30 degrees. The uncertainty of the threshold causes great difficulty for gaze estimation. For traditional eye tracking, there are two cases. First, the experiment will require the user not to turn the head, and the second is to combine the system with a gyroscope to obtain an accurate head rotation angle to calibrate the direction of gaze.

In this study, it is difficult to determine the exact head rotation angle because of the limitation of devices and speed requirement. Therefore, we propose a method of rough estimation for the users' gaze.

A. Initialization

Define a rectangular area as the user's attention area, this rectangle should be positioned in the front of the user's head left and right within 30 degrees. Define the upper left corner as Corner 1 and the lower right corner as Corner 2, use Corner 1 and Corner 2 to initialize the system. Capture the images of the user when the user peers at Corner 1 and Corner 2 freely, which means the user can move the eyeballs and the head together. Define the upper left corner of the image as (1,1), to the right is x positive direction and down is y positive direction, to establish the coordinate axis and calculate the position of the pupils, as shown in Fig. 4.

Then we can get the face motion range F {($xf1$, $yf1$), ($xf2$, $yf2$)}, (xf, yf) is the center point of the face. And the eye motion range E {($xe1$, $ye1$), ($xe2$, $ye2$)}, (xe, ye) is the midpoint of the line connecting the two pupils' center points. Then we capture the images of the user when the user peers at Corner 1 and Corner 2 without the head's movement and record the position of the pupils to obtain the eye motion range D {($xd1$, $yd1$), ($xd2$, $yd2$)}, (xd, yd) is the midpoint of the line connecting the two pupils' center points.

In this study, we could not accurately determine the angle of rotation of the eye, we assumed that the approximate horizontal movement of the pupil in the image was proportional to the distance change of the gaze.

As shown in Fig. 5, since the image is two dimensional, the pupil is moved on the image by line segment b. However, the shape of the eyeball is an approximate sphere. The line segment a is proportional to the moving distance of the gaze in reality. The line segment b will always be shorter than the line segment a. When the rotation angle is larger, the gap between a and b will be greater, which will significantly affect the final result. Provided that the diameter of the eyeball is 10 mm, when the angle of eyeball rotation is 30 degrees, the difference between the line segment a and the line segment b is less than 0.8 mm. This is still acceptable in our rough estimation.

B. Capture the Subsequent Images

Capture the subsequent images using the camera, process the image using face detection, eye detection and pupil detection. In a given frame after starting gaze estimation, the center point of the face is (*xfp*, *yfp*), and the midpoint of the line connecting the two pupils' center points is (*xep*, *yep*).

C. Gaze Estimation

Divide the attention area into equal number of columns (n) by rows (n), as shown in Fig. 6. Estimate the gaze of the user.

In the Horizontal direction, we define Tfh as a threshold of negligible horizontal face movement, if the face movement is not more than Tfh, set the horizontal face movement to 0.

If $xfp - (xf2 - xf1) / 2 > Tfh$:

$\text{Column} = (xep - xe1) \square\ (xe2 - xe1) \square\ (1 \square\ n)\ (\square\ )$

If $xfp - (xf2 - xf1) / 2 \leq Tfh$:

$\text{Column} = (xep - xd1) \square\ (xd2 - xd1) \square\ (1 \square\ n)\ (\square\ )$

In the vertical direction, we define Tfv as a threshold of negligible vertical face movement, if the face movement is not more than Tfv, set the vertical face movement to 0.

If $yfp - (yf2 - yf1) / 2 > Tfv$:

$\text{Row} = (yep - ye1) \square\ (ye2 - ye1) \square\ (1 \square\ n)\ (\square\ )$

If $yfp - (yf2 - yf1) / 2 \leq Tfv$:

$\text{Row} = (yep - yd1) \square\ (yd2 - yd1) \square\ (1 \square\ n)\ (\square\ )$

D. Output gaze position

After getting the results of horizontal direction and vertical direction, output the corresponding gaze position.

VI. SIMULATION AND ANALYSIS

In this study, we did the simulations to test the system on a monitor. The CPU of the computer used in the simulations is Intel Core i7-4790S. The operating system is Windows 10, the program's average cost of memory is 130 megabytes. The size of the monitor used in the simulations is 23.6 inches (556 mm × 399 mm), the distance between the eyes and the monitor is about 500 mm, the resolution of the webcam used in the ssimula We did two simulations, one in which the attention area was divided equally into 9 cells (3 × 3) and the other one was divided into 25 cells (5 × 5). The result is shown in TABLE I.

The program achieved a high accuracy in the 3 × 3 partitioning mode, but the accuracy rate decreased in the 5 × 5 partitioning mode. It is found in the experiment that in the 5 × 5 division mode, the program is easy to detect a neighboring area in the vertical direction, the possible reason is that the vertical angle of the eyes' movement is too small to be detected, and the accuracy of pupil detection is not sufficient to reflect this kind of movement.

The processing speed of the system reached 8.2 frames/s, which greatly exceeds the requirement for saccade eye movement.

VII. CONCLUSION

This study uses a webcam as the main device for eye tracking and achieves a high accuracy of 94% on a screen divided into 9 sections and 78% for 25 sections. Comparing to traditional accurate eye tracking method, the proposed method is convenient and low-cost. The program is lightweight and easy to apply, especially in mobile devices. We decreased the calculation load by clipping the image and limiting the search area based on geometric features. The pupil detection method proposed in this study also effectively and quickly completed the task. The memory cost of the system is low, and the processing speed greatly exceeds the requirement for saccade eye movement. Therefore, the proposed method is a fast eye tracking method which is suitable for general application in human computer interaction.

REFERENCES

[1] Melman Haim, and N. Eden, "Eye tracking system," US20180074581. 2018.

[2] J. Chen, and Q. Ji, "Probabilistic gaze estimation without active personal calibration," IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, vol. 42, pp. 609-616, 2011.

[3] W. Fuhl, M. Tonsen, A. Bulling, and E. Kasneci, "Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art," Machine Vision & Applications, vol. 27(8), pp. 1-14, 2016.

[4] L. Wang, "Analysis of human's eyeball movement," Science & Technology Vision, vol. 20, pp. 185, 2014.

[5] Y. Derhalli, M. Nufal and T. Alsharabati, "Face detection using boosting and histogram normalization." Electrical and Electronics Engineering Conference IEEE, pp. 1-6, 2016.

[6] J. Zheng, J. Hua, Z. Tang and F. Wang, "Face detection based on LBP," International Conference on Electronic Measurement & Instruments IEEE, pp. 421-425, 2017

[7] K. Kadir, MK. Kamaruddin, H. Nasir, "A comparative study between LBP and Haar-like features for Face Detection using OpenCV," International Conference on Engineering Technology and Technopreneurship IEEE, pp. 335-339, 2015.