Monocular Eye Tracking System Using Webcam and Zoom Lens

Kun Ha Suh[1], Yun-Jung Kim[2], Yoonkyoung Kim[1], Daejune Ko[1], and Eui Chul Lee[3,*]

[1] Department of Computer Science, Graduate School, Sangmyung University, Seoul, Republic of Korea
{tjrjsgk,kim_yk2606}@naver.com, Kodeajune@gmail.com
[2] Initiativesix Co., LTD., Seoul, Republic of Korea
yjkim_ceo@initiativesix.com
[3] Department of Computer Science, Sangmyung University, Seoul, Republic of Korea
eclee@smu.ac.kr

Abstract. Many researches about eye tracking have been done in field of human computer interface, virtual reality, and so on. However, most of eye tracking devices are priced high because of its complicated hardware. Hence, in this paper, we propose a simple real-time eye tracking system using web camera. In our system, we use a zoom lens for web camera, which cover the low resolution of eye region by wide-view capturing. In addition, an infrared light illuminator is required because the gaze position is calculated by using the positional relationship between the pupil and specular reflection caused by the illuminator. Our system measures the gaze position using just one eye, and does not require to wear any devices onto user's head. The average error of gaze point estimation was about 0.9 degree at the Z distance of 70cm on the 23inch size monitor. Experimental results showed that the proposed system can be used enough for navigation or pointing interaction.

Keywords: monocular, eye tracking, infrared illuminator, specular reflection.

1 Introduction

Eye tracking is as a technology to track the gaze point of user, which has wide range of applicability. Many applications using eye tracking have been utilized in areas such as computer interface for hand disabled, human computer interface, immersive game, measure the effect of PPL(Product PLacement), and so on. For more accurate and convenient, many kinds of methods have been researched [1][2][3][4].

Previous eye tracking methods can be categorized into two groups: (1) wearable camera based method and (2) remote camera based method. In the wearable camera based methods, some equipments are required to wear for eye tracking [5]. The method of this type has the advantage that a large and accurate eye image for calculating gaze point can be obtained by capturing the eye at close distance. Also, facial movements are allowed because the device moving along the face. However, wearable device may provide discomfort to the users. In addition, when the worn equipment is

* Corresponding author.

James J. (Jong Hyuk) Park et al. (eds.), *Advanced Multimedia and Ubiquitous Engineering*,
Lecture Notes in Electrical Engineering 352, DOI: 10.1007/978-3-662-47487-7_21

moved by nature or mistake after initial calibration, it is difficult to detect the correct gaze point.

In the other category, remote camera based methods is used to track the eye without wearing any equipment. In this case, to guarantee high spatial resolution in far distance complex device or algorithm is needed such as an unusual camera which has PTZ(Panning & Tilting & Zooming) mechanism. These types of method are convenient but cost is high.

According to a different point of view, eye tracking method can be categorized into two groups: (1) binocular gaze detection and (2) monocular gaze detection methods. Commonly, binocular gaze detection method is more accurate than monocular one because the more information for deciding gaze point can be extracted [6]. In monocular gaze detection method, whereas, the gaze point is calculated by only one eye but gaze estimation time is decreased by processing one eye compared to the by binocular gaze detection [7].

To solve problems of wearable and binocular methods, in this paper, a new eye tracking system is proposed. To capture an image which has enough spatial resolution for eye tracking from USB(Universal Serial Bus) webcam, a default lens is replaced to a zoom lens. Eye region is detected by using the properties of the dark pupil and bright SR(Specular Reflection) and the center positions of the pupil and SR is sequentially detected through several processes. A detailed process is provided in Sec. 2.2. After initial user-dependent calibration by gazing only 4 points, gaze positions are successively calculated through adopting the mapping function between the pupil-SR vector and the four corner coordinates of the monitor which extracted by the calibration.

The rest of this paper is explained as follows. The proposed system and method are described in Sec. 2. The experimental results and conclusions are presented in Sec. 3 and 4, respectively.

2 Proposed System and Method

2.1 Proposed System

The proposed system consists of one webcam, one infrared light illuminator and one zoom lens for webcam as shown in Fig. 1.

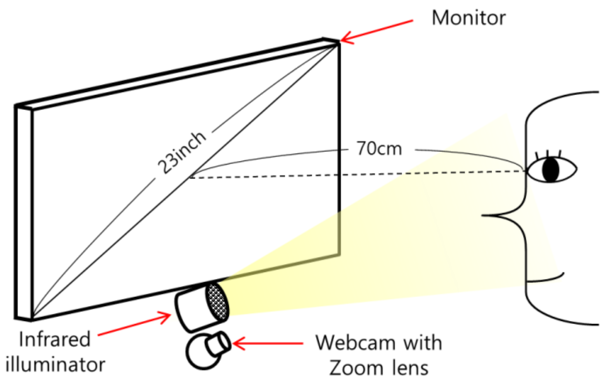


Fig. 1. Proposed eye tracking system

The used infrared light illuminator includes 50 IR-LEDs (Infrared-Light Emitting Diode) whose wavelength and illumination angle are 850nm and 30 degrees, respectively [8].

In our experiments, we used a Logitech Webcam C600 whose spatial resolution is 1600×1200 pixels [9]. To obtain the image of dark pupil and bright SR, we converted the camera to an infrared one. The infrared cutting filter of the camera is removed and infrared passing filter is attached to its place. However, since the used camera alone is not enough to ensure sufficient spatial resolution to calculate position accurately, the zoom lens is attached to the camera. As a result, it is possible to detect the accurate pupil and SR at a distance of 70cm. Also, infrared illuminator and camera are located on the bottom of the monitor as shown in Fig. 1 to avoid pupil occlusion by eyelid.

2.2 Proposed Method

The processing stage of proposed system includes three main steps. First, detecting eye region is an essential prerequisite to detect the location of pupil and SR. In our method, we use the left side eye for eye tracking. To detect an eye region, the input image is divided into sub-blocks of 64×60 size (in Fig. 3 (a)) [7]. In each sub-block, the values of maximum and minimum pixel level are measured. Then, the difference between them is calculated. Based on the difference, the sub-blocks are sorted in descending order and upper three regions are selected as an eye candidate region (in Fig. 3 (b)). Histogram stretching is applied into each candidate region then the number of black pixels is counted. Consequently, the region of including the most black pixels is regarded as eye region.

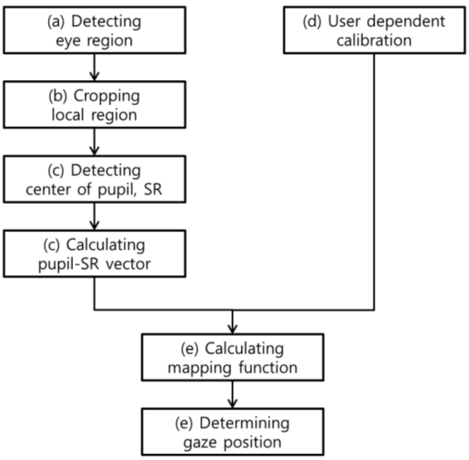
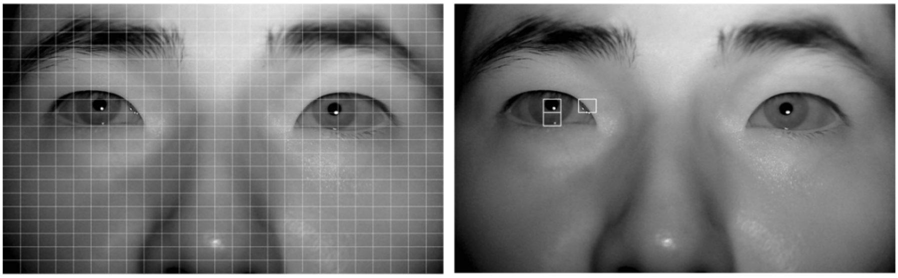


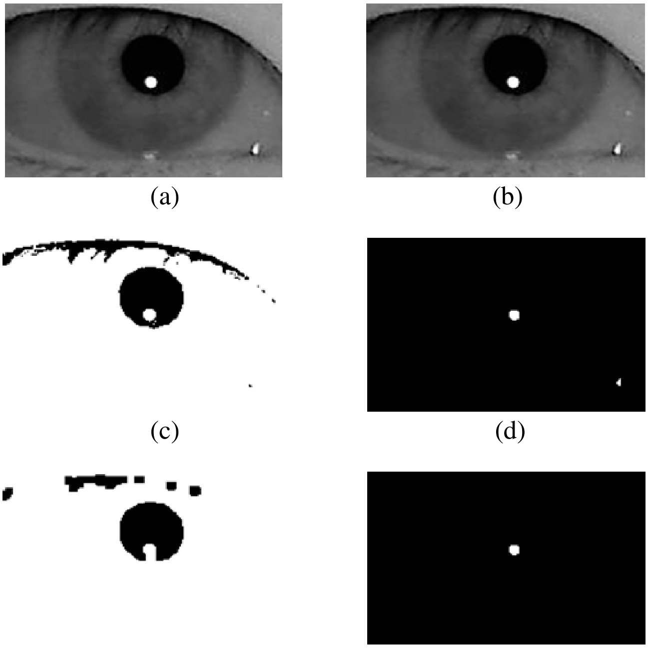
Fig. 2. Flow chart of the proposed method



(a) (b)

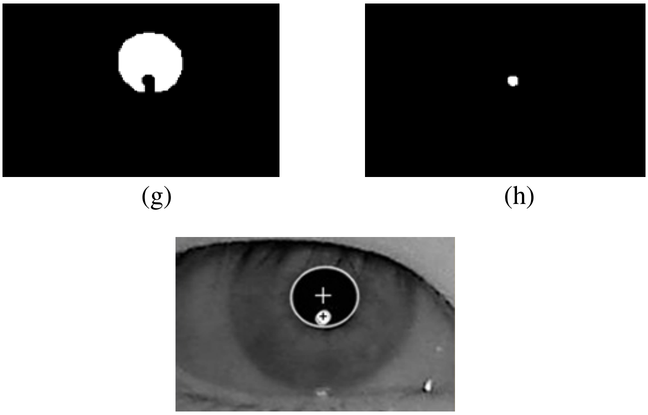
Fig. 3. Example of the eye region detection process. (a) Divided sub-blocks on the original image, (b) Three candidate regions of eye.

Next, based on the detected eye region, local region of 192×120 size is cropped from the original images. And then, calculating the center of the pupil and SR is divided sub-steps again as shown in Fig 4. For that, the binarization, morphological operation, and component labeling are sequentially preformed for pupil and SR, respectively after histogram stretching. Then, the final positions of pupil center and SR center are obtained through ellipse fitting algorithm [10].



(a) (b) (c) (d) (e) (f)

Fig. 4. Results of each step for calculating center of the pupil and SR. (a) Original local-region. (b) Histogram stretching result of (a). (c) Binarization result of (b) for the pupil. (d) Binarization result of (b) for the SR. (e) and (f) Each morphological operation result of (c) and (d), respectively. (g) and (h) Each component labeling and conditional filtering results of (e) and (f), respectively. (i) Detected center of the pupil and SR using ellipse fitting algorithm.



(g) (h)

(i)

Fig. 4. (*continued*)

Finally, we use the just one eye for eye tracking by analyzing magnitude and direction of the pupil-SR vector as shown in Fig. 5 [8]

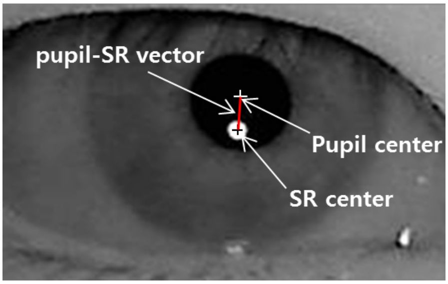


Fig. 5. Example of the pupil-SR vector

Before calculating the gaze position, user-dependent calibration should be performed as the step (d) of Fig. 2. In our method, 4-points calibration scheme is adopted. When a user gaze four corners of monitor ((m_{x1}, m_{y1}), (m_{x2}, m_{y2}), (m_{x3}, m_{y3}), (m_{x4}, m_{y4})), four pupil-SR vectors ((x_{pr1}, y_{pr1}), (x_{pr2}, y_{pr2}), (x_{pr3}, y_{pr3}), (x_{pr4}, y_{pr4})) of the eye can be defined. Then, by using the above coordinates, the gaze position on the monitor plane (G_x, G_y is calculated through following two formulas.

$$\mathrm{M} = \mathrm{T} \cdot \mathrm{V} \tag{1}$$

$$\begin{bmatrix} m_{x1} & m_{x2} & m_{x3} & m_{x4} \\ m_{y1} & m_{y2} & m_{y3} & m_{y4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{pr1} & x_{pr2} & x_{pr3} & x_{pr4} \\ y_{pr1} & y_{pr2} & y_{pr3} & y_{pr4} \\ x_{pr1}y_{pr1} & x_{pr2}y_{pr2} & x_{pr3}y_{pr3} & x_{pr4}y_{pr4} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} G_x \\ G_y \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{pr} \\ y_{pr} \\ x_{pr}y_{pr} \\ 1 \end{bmatrix} \tag{2}$$

3 Experimental Results

The proposed eye tracking system was tested with a Intel Core i5(2.30Hz) CPU and 4GB RAM. The program was implemented using Visual C++ with OpenCV library [11]. The size and the spatial resolution of used monitor in experiment were diagonally 23 inch (16:9) and 1920×1080 pixels, respectively. A total of 5 persons participated in the experiments and to validate the accuracy of the proposed system, we did a test to gaze the 9 reference points. Each person performed the test 5 times.

Experimental results showed that the RMS (root mean square) error is approximately the angular error of 0.9 degrees by 46 pixels, which despite using the webcam, it has fairly high accuracy.

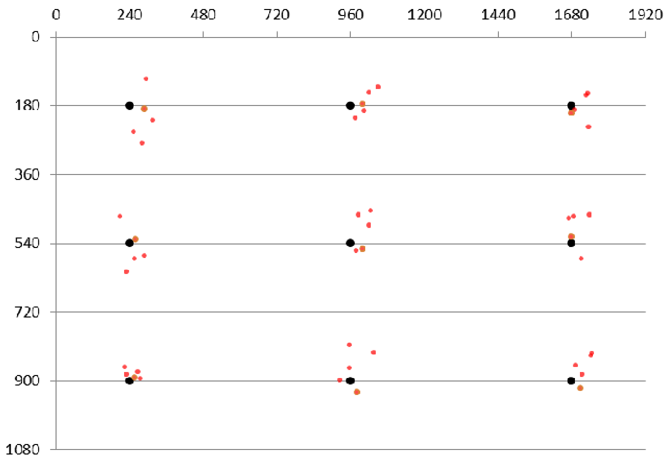


Fig. 6. Five examples of the test for measuring gaze estimation accuracy

4 Conclusion

In this paper, we proposed a new eye tracking system using webcam. The proposed system is very cheap and simple because they do not use any complex or expensive hardware. Besides, it is very convenient because there is no need to wear any equipment.

In future works, we will study a method for increasing the accuracy of the gaze point and simplify the procedure of user-dependent calibration.

Acknowledgement. This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1014) supervised by the IITP(Institute for Information & communications Technology Promotion).

References

1. Matsumoto, Y., Ino, T., Ogsawara, T.: Development of intelligent wheelchair system with face and gaze based interface. In: 10th IEEE International of Robot and Human Interactive Communication, pp. 262–267 (2001)
2. Zhu, Z., Ji, Q.: Eye and gaze tracking for interactive graphic display. Machine Vision and Applications 15, 139–148 (2004)
3. `http://www.tobii.com/en/eye-experience/eyex` (accessed on March 18, 2015)
4. Kules, B., Capra, R., Banta, M., Sierra, T.: What do exploratory searchers look at in a faceted search interface? In: 9th ACM/IEEE CS Joint Conference on Digital Libraries, pp. 313–322 (2009)
5. Lee, E.C., Park, K.R.: A study on eye gaze estimation method based on cornea model of human eye. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 307–317. Springer, Heidelberg (2007)
6. Cho, C., Lee, H., Gwon, S., Lee, J., Jung, D., Park, K., Kim, H., Cha, J.: Binocular gaze detection method using a fuzzy algorithm based on quality measurements. Optical Engineering 52(5), 053111 (2014)
7. Lee, E., Park, K.: Non-wearable eye tracking method using single camera with single illuminator. In: 2nd International Conference on Internet, pp. 170–173 (2010)
8. `http://www.koditec.com/sub211.html?cid=142&pid=254` (accessed on March 18, 2015)
9. `http://support.logitech.com/product/2-mp-webcam-c600` (accessed on March 18, 2015)
10. `http://www.sciencedirect.com/science/article/pii/S0169260798001059` (accessed on March 18, 2015)
11. `http://sourceforge.net/projects/opencvlibrary` (accessed on March 18, 2015)