Vision-Based Eye-Gaze Tracking

Kyung-Nam Kim and R.S. Ramakrishna

Department of Information and Communications, K-JIST
1 Oryong-dong, Puk-gu, Kwangju 500-712 Republic of KOREA
Phone: +82-62-970-2280, Fax: +82-62-970-2204
Email: {knkim, rsr}@kjist.ac.kr

Abstract— Human eye-gaze has the potential to be an input mode of future computers. Eye movement has been the primary focus of research in this area. Non-intrusive eye-gaze tracking that allows slight head movement is addressed in this paper. A small 2D mark is employed as a reference to compensate for this movement. The iris center has been chosen for purposes of measuring eye movement.

Two algorithms have been proposed for tracking the iris center: the Longest Line Scanning (LLS) algorithm and the Occluded Circular Edge Matching (OCEM) algorithm. Adaptive template matching is the core of OCEM. It can detect the iris center under normal lighting conditions with unexpected noise. LLS is faster, but it is sensitive to noise and the distribution of edge pixels.

The gaze point is estimated after acquiring the eye movement data. A geometry-based estimation technique and an adaptive estimation technique have been proposed for the purpose. These estimation techniques can detect gaze point with a high success rate at low screen resolution (8×10). Failures are due to factors such as errors in iris center detection, low camera resolution, and linear approximation in estimation.

KEY WORDS : *eye-gaze tracking, human-computer interface, computer vision, and non-intrusive technique*

I. INTRODUCTION

The movement of user's eyes can provide a convenient, natural and high-bandwidth source of input. By tracking the direction of gaze of the user, the bandwidth of communication from the user to the computer can be increased by using the information about what the user is looking at, and even designing objects specially intended for the user to look at.

A variety of eye-gaze (eye-movement) tracking techniques have been reported in the literature [1]. A short list includes (a) Electro-Oculography [2], (b) Limbus, Pupil, and Eyelid Tracking [3], [4], [5], [6], [7], [8], [9], (c) Contact Lens Method, (d) Corneal and Pupil Reflection Relationship [5], [4], [8], (e) Purkinje Image Tracking, (f) Artificial Neural Networks [10] and (g) Head Movement Measurement [11], [12], [9], [13].

Computer vision is intrinsically non-intrusive, and does not require any overly expensive equipment. Non-obtrusive sensing technology - such as video cameras and microphones - has received special attention in this regard. This paper draws on computer vision and image processing techniques for measuring eye-gaze.

The remainder of this paper is organized as follows. The proposed eye movement tracking algorithms are presented in Section 2. Section 3 shows how to predict eye-gaze through an appropriate geometric model and image-to-screen mapping. Experimental results are presented in section 4. Section 5 includes conclusion and further research directions.

II. EYE MOVEMENT TRACKING METHODS

The location of face and eye should be known for tracking eye movements. Exact eye movements can be measured by special techniques. This investigation concentrates on tracking eye movement itself. The primary goal is to detect the exact eye position. Two algorithms have been proposed for iris center detection : *the Longest Line Scanning* and *Occluded Circular Edge Matching* algorithms. The emphasis is on eye movement in this paper, not on face and eye location. If the initial region of the eye is known, then it is easy to track the eye itself.

Rough eye position is not sufficient for tracking eye-gaze accurately. Measuring the direction of visual attention of the eyes requires more precise data from eye image.

A. What to track?

It was decided to track the 'iris' for this reason. Due to the fact that the sclera is light and the iris is dark, this boundary can easily be optically detected and tracked.

Young [14] has addressed the iris tracking problem using a head-mounted camera. There are some issues, however, which have to be emphasized. They arise, due to

the following reasons :

1. Coverage of the top and bottom of the limbus by the eyelids.

2. Poor quality of the images.

3. Excessive coverage of the eyes by eyelids (in some cases).

The techniques proposed in this paper effectively deal with the first two, while the last is an inherently hard problem.

B. Longest Line Scanning (LLS) Algorithm

Actually, the eye image is a projection of the real eye. The projection of the iris is elliptical in shape. The following lemma concernig the ellipse is useful in this regard.

Lemma. *The center of an ellipse lies on the center of the longest horizontal line inside boundary of ellipse* □
The LLS algorithm is an application of this lemma. It can be applied to the problem of detection of the iris center.

The algorithm is given below.

Algorithm 1 *Longest Line Scanning :* Figure 1(a)
---
```
(* Input : the block image containing one eye *)
(* Output : the iris center *)
```
$Thresh_{iris}$: the threshold of iris color
I_{bin} : the binary block image
```
 Begin
    Threshold Input into I_bin by Thresh_iris;
    Find centroid of iris pixels; (* as a candidate *)
    Detect edges of I_bin; (* canny or vertical operator *)
    Search lhl between iris edges in I_bin;
    If more than one lhl Then find mid vertical position;
    Store midpoint of last found line into Output;
 End.

(* Note : lhl stands for longest horizontal line *)
```

Experimental results are shown in Figure 1(b). Searching and decision after edge detection enhances computational efficiency. Except when preprocessing fails, it computes the center of the iris quite accurately. But it is sensitive to distribution of edge pixels.

C. Occluded Circular Edge Matching (OCEM) : a better solution

Although the LLS method detects the center of the iris, it is not sufficient for measuring eye-gaze precisely. The follwing problems are noted on a closer look at LLS technique :
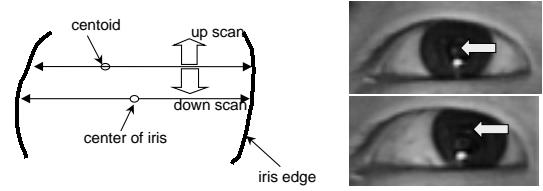
(a) Principles of LLS   (b) The Result Images of LLS

Fig. 1. Longest Line Scanning (LLS)

- intra-iris noise
- rough iris edge
- occlusion of longest line by eyelids

The only clues to find the center of the iris are left and right edge pixels of the iris boundary, the so called limbus. In order to estimate the original position and shape of the iris boundary, the circular edge matching (CEM) method can be adapted. As mentioned earlier, the iris is naturally occluded by eyelids to some extent, depending upon the individual or the status of the subject. CEM should be adaptively modified. Only the visible portions of the edge without occluded portions need to be processed in the matching step.

The angle of rotation of the eyeball and the eccentricity of the ellipse are not large, when the subject sits and operates the computer in front of the screen. This observation justifies a circular approximation to the ellipse. Experimental results justify this simplification.

The algorithm is outlined below. (Steps which closely resemble those in LLS have been omitted).

Algorithm 2 *Occluded Circular Edge Matching*

```
(* Input : the edge image after LLS,
           P_c, the centroid of iris pixels,
           P_p, the midpoint, ( of horzontal projection )
           P_l, the resulting point computed by LLS *)
(* Output : the iris center *)
 Begin
   Select the candidate point out of P_c, P_p, and P_l;
   Set the circle center matching window;
   for all pixels in the circle center matching window do
      Circular Edge Matching;
      Scoring its pixels matched with the edge pixel;
   end for
   Store the pixel having maximum score into Output;
 End.
```

The algorithm is described in the sequel.

• **Select the Candidate Point.** This is the initial point. The center of the circle matching windows is a good candidate. A carefully selected candidate point reduces the size of search window and the computation time. Three different kinds of candidate point are considered below :

1. *Centroid.*
2. *Midpoint (of horizontal projection)*
3. *The point computed by the LLS algorithm.*

• **Set the Matching Window.** Matching process is performed by moving the center of candidate circle inside the circle matching window (See figure 2(a)). The size of this window affects the computation time. A reasonable size is :

$$Horizontal(Vertical)Size = \frac{2}{5}r_{iris} \sim \frac{4}{5}r_{iris}$$

The radius of the iris, r_{iris}, can be obtained from the result of the previous image frame (*radius*, unless otherwise indicated refers to this radius).

• **Matching** (Figure 2(a)). The left and right curvatures of the iris candidate are matched with those of the iris to be detected in the edge image. Several factors must be carefully determined for better performance :

1. *The portion to be matched.*
2. *Change of radius.*
3. *Interlacing.* Interlaced selection of pixels to be matched accelerates the process albeit losing on accuracy slightly.
4. *Distance of neighbors*

• **Scoring and Decision.** Every match is scored. The iris candidate with maximum score is chosen as the final solution. Its center is the center of the eye.

Figure 2(b) shows the iris center being detected correctly.
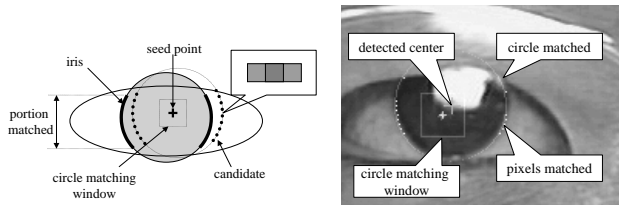
(a) Matching Process in OCEM   (b) Experimental Result

Fig. 2. Occluded Circular Edge Matching

III. Gaze Estimation

The focus is on estimating the orientation of the eyes with slight head movement. It is very important to estimate it from the image features and values measured at the stage of eye movement tracking. The direction of eye-gaze, including the head orientation is considered in this investigation. A new geometric model incorporating a reference has been devised. The geometry consisting of subject's face, camera, and computer screen has been explored so as to understand eye-gaze in this environment. Finally, a couple of estimation methods have been proposed.

A. Geometric Reference

A small mark attached to the glasses stuck between two lenses has been adopted for the purpose of the special rigid origin (Figure 3). This provides the following geometric information.

• The position of subject's face
• The origin of the iris center movement

It cannot offer any orientation information at all, because it is like a small spot. Nevertheless, it can compensate for slight head movement.
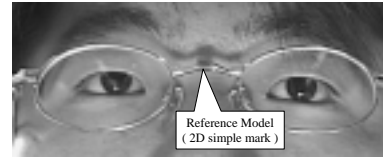

Fig. 3. Referecne Model: 2D Simple Mark

B. Geometry of Eye-Gaze

Although free head movement is allowed in ordinary situations, some restrictions are imposed on experimental setup for simplicity. People tend to fix their head parallel to the screen plane when they use the computer. Eyes and camera are assumed to lie in the normal to the screen, and natural, but slight, head movement is allowed.

Information provided by eye movement tracking before gaze estimation consists of

• *The position of reference model.* This involves the position of the face (rigid body).
• *the radius of iris.* This is useful in estimating the distance from the camera to the subject.

• *the vector from the reference model to the iris center.*
This carries information on eye movement.

The imaging system is assumed to employ orthogonal projection. Assuming orthogonal rather than weak perspective projection simplifies the discussion. Only one (horizontal or vertical) dimension is considered for purposes of analysis.

Figure 4 depicts how much displacement of the iris center in the projection reflects the displacement of the eye-gaze. Three gazes (gaze 1, gaze 2, and gaze c) and projection of each iris center and gaze are shown in the figure. Gaze c is the reference gaze. d is the distance from the eyeball surface to the screen plane. r_{ball} is the radius of the eyeball, which ranges from 12mm to 13mm (according to the anthropometric data). Δ_1 and Δ_2 are the displacements of the iris center of gaze 1 and gaze 2, respectively, from that of gaze c in the projection. g_1 and g_2 are the displacements of gaze 1 and gaze 2 respectively, from gaze c. The input and output are Δ_1, Δ_2, and g_1, g_2 respectively.

If $\alpha = r_{ball} - \sqrt{r_{ball}^2 - x^2} = 0$, then

$$g_1 = \frac{d + r_{ball}}{r_{ball}}(x_1 - x_c) = \frac{d + r_{ball}}{r_{ball}}\Delta_1 \qquad (1)$$

$$g_2 = \frac{d + r_{ball}}{r_{ball}}(x_2 + x_c) = \frac{d + r_{ball}}{r_{ball}}\Delta_2 \qquad (2)$$

Although these approximations simplify the estimation, care should be exercised in their use. The approximation error is computed below. If $k = \frac{realworld-displacement}{image-displacement}$, then

$$error_{image} = \frac{1}{k}sr_{ball}\left[\frac{1}{d + r_{ball}} - \frac{1}{\sqrt{(d + r_{ball})^2 + s^2}}\right] \qquad (3)$$

This $error_{image}$ can be ignored in a real experiment.

C. Image-to-Screen Mapping

In this section, the techniques to determine gazing points on the computer screen are discussed. The *Geometry-Based Estimation* is, indeed, based on the geometry of the eye-gaze discussed in the previous section. *Adaptive Estimation* determines the eye-gaze with the help of displacements in the image.

Regardless of which of these techniques is actually employed, image-to-screen mapping requires that the system be initialized first. It should be calibrated while in
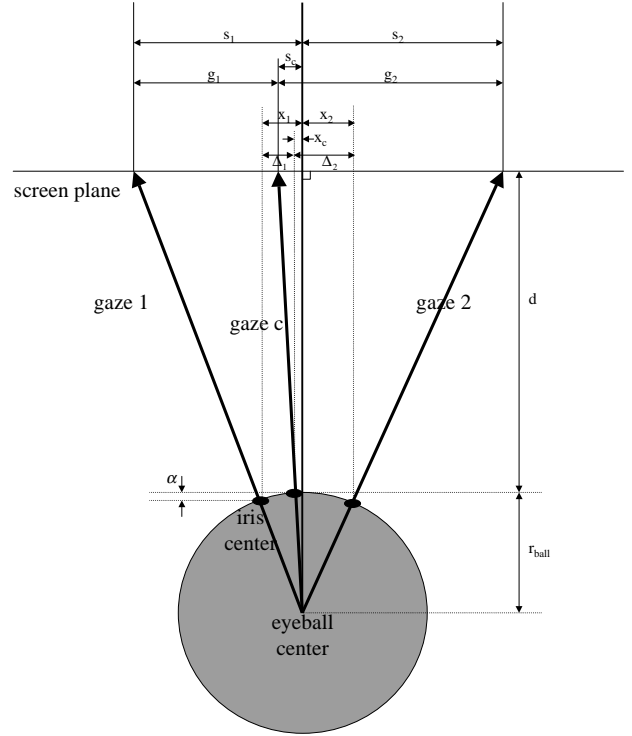

Fig. 4. Geometry of Eye-Gaze

use. During initialization, the subject intentionally gazes at predefined screen points. From the resulting eye movement data, other gazing points can be estimated. During the calibration, because subject moves continuously, changes in the parameters (such as the radius or iris, the distance, or the head position arising due to subject movements) are incorporated in the estimation process, thereby reconstructing the parameter set.

C.1 Geometry-Based Estimation

The subject first gazes at the center of the screen, and then, slightly moves and gazes at the right end of the screen Figure 5 shows its geometry . S is the distance between two screen points. Δ_{ref} is the displacement of the reference model.

Using equations 1 and 2, we get

$$S = k\left\{\frac{d + r_{ball}}{r_{ball}}(\Delta_2 - \Delta_1) + \Delta_{ref}\right\} \qquad (4)$$

During initialization, the value of k is expected to be different depending on the direction towards each predefined screen points. The different value of k can be computed at this initialization stage. The value S refers
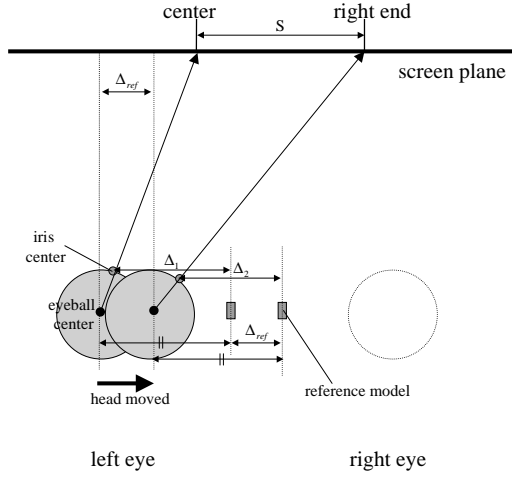
Fig. 5.   Geometry including Slight Head Movement

to the gazing point. The situation is the same as in the initialization step.

Algorithm 3 *Geometry-Based Estimation*

```
(*  Input :
 d, the distance from the eyeball to the screen,
 r_ball, the radius of the eyeball,
 Δ_1, Δ_2, the distance from the iris to the reference,
 Δ_ref, the displacement of the reference,
 S_origin, the screen position of the original gaze *)
(*  Output :
 S_current, the current gaze's screen position *)
Begin
    (* Initialization *)
        find k_x at each different x direction
        from predefined gazes;
    (* Estimation *)
        decide x, the direction of current gaze;
        retrieve k_x and find S_current;
End.
```

C.2 Adaptive Estimation

This technique adaptively uses only the displacement of the iris center and the displacement of the reference model. Based only on initialization data, it determines gazing point by linear approximation. It involves the following algorithm :

IV. EXPERIMENTAL RESULTS

In this section, experimental results pertaining to eye movement tracking and the corresponding (estimated) gazing points are presented.

Algorithm 4 *Adaptive Estimation*

```
(*  Input : the image positions of the iris and the
        reference at current gaze and each predefined gaze,
        k_calib, calibrating factor *)
(*  Output : S, the current gaze's screen position *)
Begin
    (* Initialization *)
        find the gaze vector of each predefined gaze
        in image and set an original gaze;
    (* Estimation *)
        compute the gaze vector of current gaze;
        compensate for slight head movement using k_calib;
        compute S;
End.
```

A. Experimental Results and Analysis

LLS and OCEM estimated gazing points at 3×3, 4×5, and 8×10 screen resolutions by geometry-based and adaptive estimation methods. We only present in detail the results of experiments with 8 × 10 Screen Resolution.

B. 8 × 10 Screen Resolution

OCEM is employed for eye movement tracking and both the Adaptive and Geometry-Based estimation techniques are utilized for estimation. Figures 6(a) show results of the right eye movement tracking at initialization. Each image of subject's iris being tracked, while he gazes at 10 screen points randomly (from small rectangle 1 to 10), is given in Figure 6(b). Both eyes are tracked simultaneously. (Similar results were obtained for left eye).
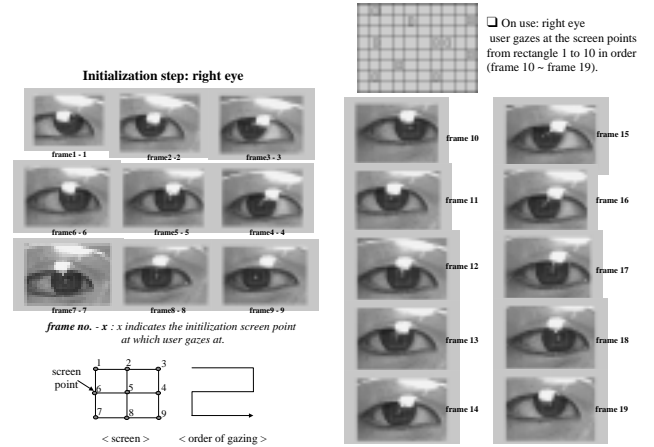
(a) Initialization step : right eye.

(b) Subject's random gazes : right eye

Fig. 6.   8 × 10 Screen Resolution

The frames from 1 to 9 are for the initialization step

TABLE I

Estimated Gazing Points in 8 × 10 Screen Resolution

Frame	Original Point	Geometry Based		Adaptive
		Scheme I	Scheme III	
frame10	32	32	32	32
frame11	15	15	15	15
frame12	54	54	54	54
frame13	37	37	37	37
frame14	38	**28**	**28**	**28**
frame15	20	20	20	20
frame16	50	50	50	50
frame17	67	67	67	**77**
frame18	62	62	62	62
frame19	2	2	2	2

and the remainder is the data when the subject gazes at the screen point in which he is interested.

The subject gazes at the screen points in the following order : $32 \rightarrow 15 \rightarrow 54 \rightarrow 37 \rightarrow 38 \rightarrow 20 \rightarrow 50 \rightarrow 67 \rightarrow 62 \rightarrow 2$. They correspond to each frame from 10 to 19. Gazing points are estimated by linear interpolation on the estimation results of both the left and the right eye. Table I shows the estimation results employing Geometry-Based Estimation and Adaptive Estimation.

C. Analysis

The proposed eye-gaze tracking methods are seen to be quite successful at screen resolutions up to 8×10 screen resolution. Geometry Based Estimation is better than Adaptive Estimation in estimating eye-gaze. A small number of failures result from large head movements and error in eye movement tracking.

Eye-Gaze tracking methodology using video technology has its inherent limitations : camera resolution limits measurement of eye movement. In the experiments, one has about 20 pixels only for estimating the entire vertical computer screen range.

Some solutions are :

- Using camera with higher resolution.
- Placing the camera closer to subject's face.
- Employing two cameras, one for head tracking, the other for eye movement tracking.

V. Conclusions and Further Directions

Non-intrusive vision-based eye-gaze tracking methods involving eye movement tracking (the iris center tracking) and gaze estimation have been investigated in this paper. Neither special lighting source nor cumbersome equipment is required for their implementation. Practical feasibility of the techniques has been demonstrated by using them as one type of computer interface (the substitute for a pointing device). The subject is allowed to move slightly, in a natural way. The eye-gaze is computed by finding correspondences between points in a model of face and points in the camera image.

More robust initialization and calibration for different subjects, different computers and different environments are essential for these techniques to be employed routinely in computer interfaces.

References

[1] Laurence R. Young and David Sheena, Survery of Eye Movement Recording Methods, *Behavior Research Methods and Instrumentation*, Vol. 7, No. 5, pp. 397-429, 1975.

[2] Arie E. Kaufman, Amit Bandopadhay, Bernard D. Shaviv, An Eye Tracking Computer User Interface , *Research Frontier in Virtual Reality Workshop Proceedings*, IEEE Computer Society Press, pp. 78-84. October 1993,

[3] Glenn A. Myers, Keith R. Sherman, and Lawrence Stark, Eye Mornitor, *IEEE Computer Magazine*, Vol. March, pp. 14-21, 1991.

[4] Yoshinobu Ebisawa, Improved Video-Based Eye-Gaze Detection Method, *IEEE IMTC '94*, Hamamatsu, May, 1998.

[5] Thomas E. Hutchinson, K. Preston White, Jr., Worthy N. Martin, Kelly C. Reichert, and Lisa A. Frey, Human-Computer Interaction Using Eye-Gaze Input, *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 19, No. 6, pp. 1527-1534, 1998.

[6] C. Colombo, S. Andronico, and P. Dario, Prototype of a Vision-Based Gaze-Driven Man-Machine Interface, *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, August, 1995.

[7] Christophe Collet, Alain Finkel, and Rachid Gherbi, CapRe: A Gaze Tracking System in Man-Machine Interaction, *Proceedings of IEEE International Conference on Intelligent Engineering Systems*, September, 1997.

[8] Baoshen Hu and MingHua Qiu, A New Method for Human-Computer Interaction by using Eye Gaze, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, October, 1994.

[9] Rainer Stiefelhagen, Gaze Tracking for Multimodal Human-Computer Interaction, *Diplomarbeit*, Universität Karlsruhe, September, 1996.

[10] Shumeet Baluja and Dean Pomerleau, Non-Intrusive Gaze Tracking Using Artificial Neural Networks, *CMU Technical Report*, CMU-CS-94-102, School of Computer Science, Carnegie Mellon University, January, 1994.

[11] Philippe Ballard and George C. Stockman, Computer Operation via Face Orientation, *Pattern Recognition, Vol.1. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference*, 1992.

[12] Thanarat Horpresert, and Yaser Yacoob, and Larry S. Davis, Computing 3-D Head Orientation from a Monocular Image Squence, *Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, October, 1996.

[13] A. H. GEE AND R. CIPOLLA, Determining the Gaze of Faces in Images, *Technical Report*, CUED/F-INFENG/TR 174, Department of Engineering, University of Cambridge, March, 1994.

[14] DAVID YOUNG, HILARY TUNLEY, AND RICHARD SAMUELS, Specialised Hough Transform and Active Contour Methods for Real-Time Eye Tracking, *Cognitive Science Research Paper*, 386, School of Cognitive and Computing Sciences, University of Sussex, July, 1995.