시선추적을 이용한 휴먼-컴퓨터 인터페이스

°
김경남, R.S.Ramakrishna
광주과학기술원 정보통신공학과

Human-Computer Interface using Eye-Gaze Tracking

°
Kyung-Nam Kim, R.S.Ramakrishna
Department of Information and Communications
Kwang-Ju Institute of Science and Technology(K-JIST)

Abstract

Human eye-gaze has the potential to be an input mode of future computers. 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 x 10). Failures are due to factors such as errors in iris center detection, low camera resolution, and linear approximation in estimation.

1. 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. A short list includes (a) Electro-Oculography [1], (b) Limbus, Pupil, and Eyelid Tracking [2,3,4,5,6,7,8], (c) Contact Lens Method, (d) Corneal and Pupil Reflection Relationship [3,4,7], (e) Purkinje Image Tracking, (f) Artificial Neural Networks [9] and (g) Head Movement Measurement [8,10,11,12].

Eye movement monitoring techniques fall broadly into two categories - those that measure the position of the eye relative to the head and those that measure the orientation of the eye in space. Techniques belonging to the second category are normally required [14]. Although these may often be fairly accurate, such techniques have some limitations for general purpose applications, which arise from high cost of the embedded technologies, from being intrusive and from usually constrainting the natural mobility of the user.

Computer vision is intrinsically non-intrusive, and does not require any overly expensive equipment. 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.

2. 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. 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.

2.1 Face and Eye Locating

Several facial features should be detected for tracking the face. There have been several studies on detecting eye location : average anthropometric measures, the deformable template and snakes (active contour model) [15], the region-based approach [16], to list a few.

Roughly measured eye location is used as an input to an eye-gaze tracking system. The measurement is refined in the next step, viz., eye movement tracking.

Face has its stable anthropometric and symmetric properties. For instance, eyes are located above the nose and the mouth. Although it depends upon the individual, race, sex, and age, humans have consistent distances between facial features. Human faces are constructed in the same geometrical configuration. These anthropometric data can be utilized in locating the eye.

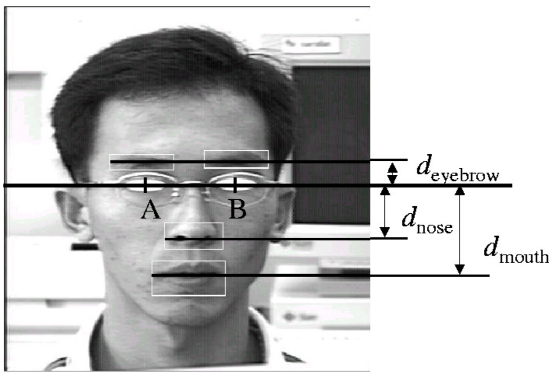

Fig. 1. The Geometric Face Model.

Fig. 1 is based on these observations. The base line passes through the centers of both the eyes. The distances from the center of nose, eyebrows, and mouth to the base line are denoted, respectively by d_{nose}, $d_{eyebrow}$, and d_{mouth} in the figure [17].

If the face is classified from background, then the eye area can be set either by thresholding and geometric constraints, or interactively by the subject.

2.2 Eye Movement Tracking : Iris Center Detection

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.

2.2.1 What to track?

A distinctive feature of the eye image should be measured in any arrangement. The pupil of people having dark or dark-brown eyes can hardly be differentiated from the iris in the captured images. If the image is captured from close range, then it can be used to detect the pupil even under ordinary lighting conditions. 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.

It is important that the procedures should be effective under natural conditions:

- Subject's face is located naturally away from the camera and at a reasonable distance.
- No special apparatus or equipment should be needed.
- No special illumination should be deemed necessary.

Young [13] 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.

2.2.2 Longest Line Scanning (LLS) Algorithm

Human eyes have three degrees of freedom of rotation in 3D space. Actually, the eye image is a projection of the real eye. The iris is nearly a circular plate attached to the approximately spherical eyeball. The projection of the iris is elliptical in shape.

The following lemma concerning 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 following assumptions should be noted :

1. Search window already found includes the iris in its entirety.
2. The longest line inside the iris is not occluded by the eye-lid.
3. There is no noise edge inside the iris.

Assumption 1 and 2 can be generally guaranteed, but 3 is an obstacle to find the longest line inside the iris. This has been solved by preprocessing. The algorithm is given below.

(* **Input** : the block image containing one eye after eye location *)
(* **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 point *)
Detect edges of I_{bin}; (* canny or vertical sobel operator *)
Search lhl between iris edges in I_{bin} during up/down scan;
If more than one lhl **Then** find mid vertical position;
find midpoint of last found line and store in into **Output**;
End.

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

Algorithm 1. Longest Line Scanning (LLS)

2.2.3 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 following problems are noted on a closer look at LLS technique :

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

Fig. 2 shows these problems. 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.

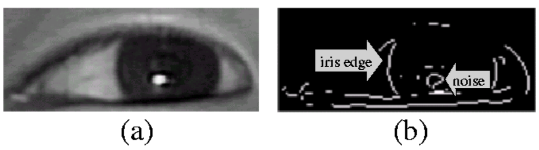

Fig. 2. (a) eye image, (b) its edge image. Look carefully the edge image in which there is some type of noise, and the only clue to find a center of iris are left and right edge of iris boundary, the so called limbus.

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. The approximation also simplifies the discussion without significant loss of accuracy and generality. It also accelerates the matching step. Experimental results justify this simplification.

It should be noted that OCEM does not require any preprocessing as an essential step. The matching step tolerates noise to some extent. The algorithm is outlined below. (Steps which closely resemble those in LLS have been omitted).

(* **Input** : the edge image after LLS,
P_c, the centroid of iris pixels,
P_p, the midpoint, (of horizontal projection)
P_l, the 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 within the circle center matching window **do**
Circular Edge Matching;
Scoring its matched pixels with the corresponding edge pixel;
end for
Find the pixel having maximum score and store it into **Output**;
End.

Algorithm 2. Occluded Circular Edge Matching (OCEM)

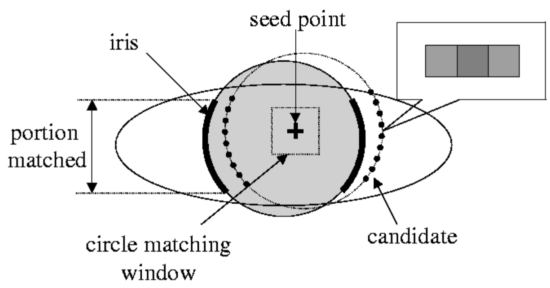

Fig. 3. Matching Process in OCEM

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 : LLS could be the best candidate selection procedure provided it is precise. Further, the computation time of LLS being very short, the overall efficiency of OCEM is not affected at all.

- **Set the Matching Window.** Matching process is performed by moving the center of candidate circle inside the circle matching window (See Fig. 3). The size of this window affects the computation time. A reasonable size is :

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

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). The horizontal extent may be smaller than the vertical one, when the candidate point is selected to be the middle point of projection.

• **Matching** (Fig. 3). 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.* This can be determined by noting how much top and bottom portions of the iris are occluded. A reasonable value of this portion ranges from 1×*radius* to 1.5×*radius*.
2. *Change of radius.* The radius may change slightly across the images. A small change of the radius should be allowed during the computation. The change, $Delta_{radius}$, satisfies the inequality

$$\frac{1}{15} \times radius \le \mathrm{Delta}_{radius} \le \frac{1}{5} \times radius$$

3. *Interlacing.* Matching is highly computationally intensive. Interlaced selection of pixels to be matched accelerates the process albeit losing on accuracy slightly. Experiment shows it works well. Dotted curves of the candidate in Fig. 3 depict interlaced matching.
4. *Distance of neighbors* (See the box in Fig. 3). The distance of these neighbors from the pixel being considered can be extended to more than one.

• **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.

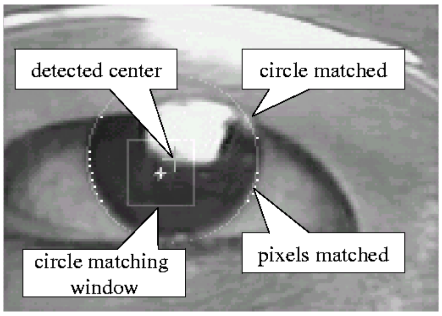

Fig. 4. Experimental Result of OCEM

Fig. 4 shows the iris center being detected correctly.

3. Gaze Estimation

The final goal of the eye-gaze tracking system is to find the direction of eye-gaze. A person's gaze direction is determined by two factors: the orientation of the head, and the orientation of the eyes. While the orientation of the head determines the overall direction of the gaze, the orientation of the eyes determines the exact gaze direction and is limited by the head orientation. The head orientation does not significantly affect the gaze direction (in computer interface applications). The focus is, therefore, 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.

3.1 Geometric Reference

The line of sight (i.e., the gaze direction) is defined as the vector from the eyeball center to the center of the iris in 3D space. The eyeball rotates freely within limited angles. The iris center alone is not sufficient to measure eye-gaze. The iris is not a rigid body but a moving object in the face. It is very difficult to compute the extent of iris movement. The iris movement data are valid only when the face is stationary. This situation can be found in many other eye-gaze tracking systems using HMD, head mounted camera, or some intrusive apparatus. In order to allow head movement, a certain special rigid origin fixed in the subject's head to measure the displacement of iris is required.

A small mark attached to the glasses stuck between two lenses has been adopted for the purpose (Fig. 5). 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.

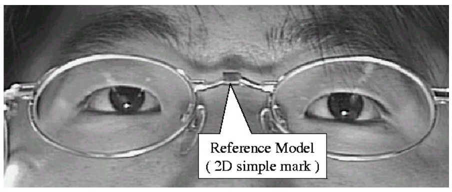

Fig. 5. Reference : 2D Simple Mark

3.2 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*. 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 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.

Fig. 6 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). $\triangle_1$ and $\triangle_2$ are the displacements of the iris center of gaze 1 and gaze2, 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 $\triangle_1$, $\triangle_2$, and g_1, g_2 respectively.

The geometry of the triangle yields

$$s=\frac{(d+r_{ball})}{\sqrt{r^2_{ball}-x^2}}x$$

and

$$g_1=s_1-s_c=\left\{\frac{x_1}{\sqrt{r^2_{ball}-x^2_1}}-\frac{x_c}{\sqrt{r^2_{ball}-x^2_c}}\right\}(d+r_{ball})$$

$$g_2=s_2+s_c=\left\{\frac{x_2}{\sqrt{r^2_{ball}-x^2_2}}+\frac{x_c}{\sqrt{r^2_{ball}-x^2_c}}\right\}(d+r_{ball})$$

Setting

$$\alpha=r_{ball}-\sqrt{r^2_{ball}-x^2}$$

one can write

$$s=\frac{d+r_{ball}}{r_{ball}-\alpha}x$$

If $\alpha = 0$, then

$$s=\frac{d+r_{ball}}{r_{ball}}x$$

Hence, g_1 and g_2 can be rewritten as

$$g_1=\frac{d+r_{ball}}{r_{ball}}(x_1-x_c)=\frac{d+r_{ball}}{r_{ball}}\triangle_1 \qquad (1)$$

$$g_2=\frac{d+r_{ball}}{r_{ball}}(x_2+x_c)=\frac{d+r_{ball}}{r_{ball}}\triangle_1 \qquad (2)$$

Although these approximations simplify the estimation, care should be exercised in their use. The approximation error is computed below. With

$$s=\frac{d+r_{ball}}{r_{ball}}x_{\text{approxim}}$$

and

$$s=\frac{(d+r_{ball})}{\sqrt{r^2_{ball}-x^2}}x_{accurate}$$

the approximation error is

$$error_{realworld}=x_{\text{approxim}}-x_{accurate}$$
$$=sr_{ball}\left[\frac{1}{d+r_{ball}}-\frac{1}{\sqrt{(d+r_{ball})^2+s^2}}\right]$$

If k = (realworld-displacement)/(image-displacement}, then

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

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

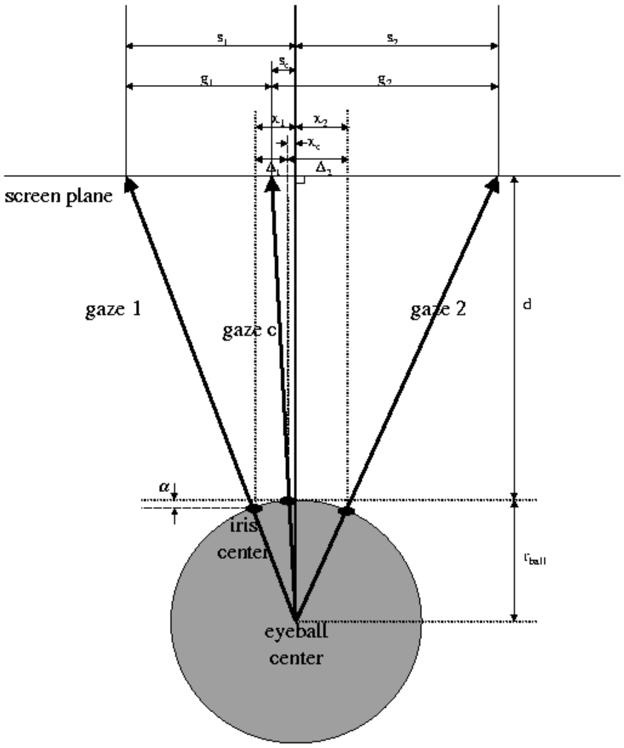

Fig. 6. Geometry of Eye-Gaze

3.3 Image-to-Screen Mapping

The only clue to estimate the eye-gaze is provided by the available image features.

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 use. During initialization, the subject intentionally gazes at predefined screen points. Usually, they consist of center, top left, top middle, top right , bottom left, bottom middle, bottom right, left middle , and right middle 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.

3.3.1 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 (b in Fig. 8). Fig. 7 shows its geometry . S is the distance between two screen points. $\triangle_{ref}$ is the displacement of the reference.

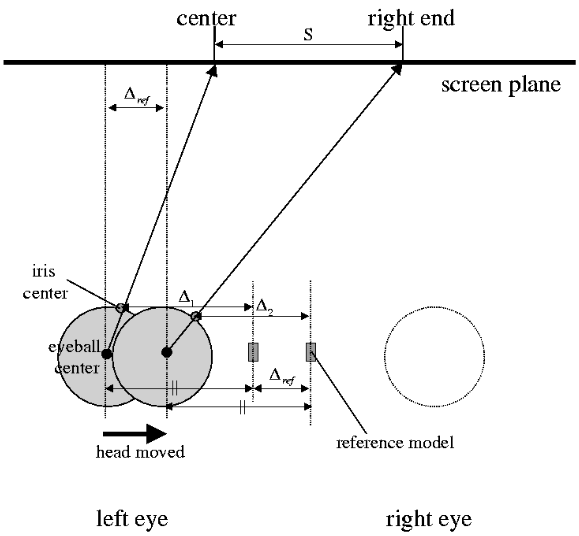

Fig. 7. Geometry including Slight Head Movement

Using equations 1 and 2, we get

$$S_{current} = k\left\{ \frac{(d + r_{ball})}{r_{ball}} (e_{current} - e_{origin}) \right\} + k(ref_{current} - ref_{origin}) + S_{origin} \qquad (3)$$

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 in the following manner (See Fig. 8).

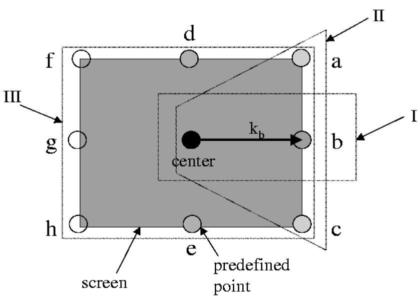

Fig. 8. Schemes in Geometry-Based Estimation

I. k_b is determined using the gaze data b itself and center.

II. k_b is determined using the gaze data a,b,c and center.

III. k_b is determined using the gaze data from a to h and center.

The value S refers to the gazing point. The situation is the same as in the initialization step. Depending on the scheme (I, II, and III), S can be computed from equation 3.

3.3.2 Adaptive Estimation

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

Another slightly more complicated scheme is described below.

- First store the initialization data and compute the horizontal center line and the vertical center line. In Fig. 9, left middle, center, and right middle points and top middle, center, and bottom middle points , respectively, determine the horizontal center line, and the vertical center line.
- Second, when a subject gazes at some screen point, find which region (I, II, III, or IV) the point is located in. For instance, in Fig. 9, assume that the region I was selected. The coordinates of the gazing point are then determined by linear approximation (from a,b,d, and center).

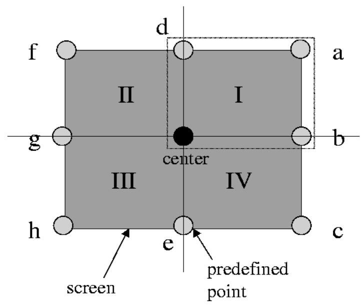

Fig. 9. Schemes in Adaptive Estimation

4. Experimental Results

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

4.1 Configuration

Before presenting experimental results, initial values and evaluation methods are indicated.

1. **Initial Values** (including anthropometric data)}
   - *radius of eyeball*, r_{ball} : 12 mm
   - *radius of iris* : 6 mm

• *Size of Screen* : 315(H) x 240(V) mm
• *Distance from screen to subject, d* : 400 ~ 700 mm

2. **Evaluation Method**

• *Initialization* : subject gazes at 9 predefined screen points
• *Evaluation* : subject again gazes at the region separated by 3 x 3, 4 x 5, 8 x 10 screen resolution.

4.2 Experimental Results and Analysis

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

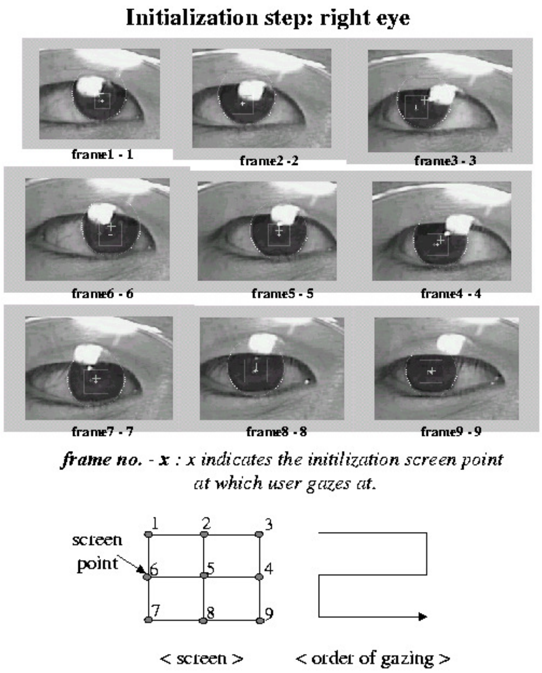

Fig. 10. Initialization step : right eye. Because each image is a search window that adaptively changes its size, Figure looks disarranged.

OCEM is employed for eye movement tracking and both the Adaptive and Geometry-Based estimation techniques are utilized for estimation. Fig. 10 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 Fig. 11. Both eyes are tracked simultaneously (Similar results were obtained for left eye).

The subject gazes at the screen points in the following order :

32 → 15 → 54 → 37 → 38 → 20 → 50 → 67 → 62 → 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 1 shows the estimation results

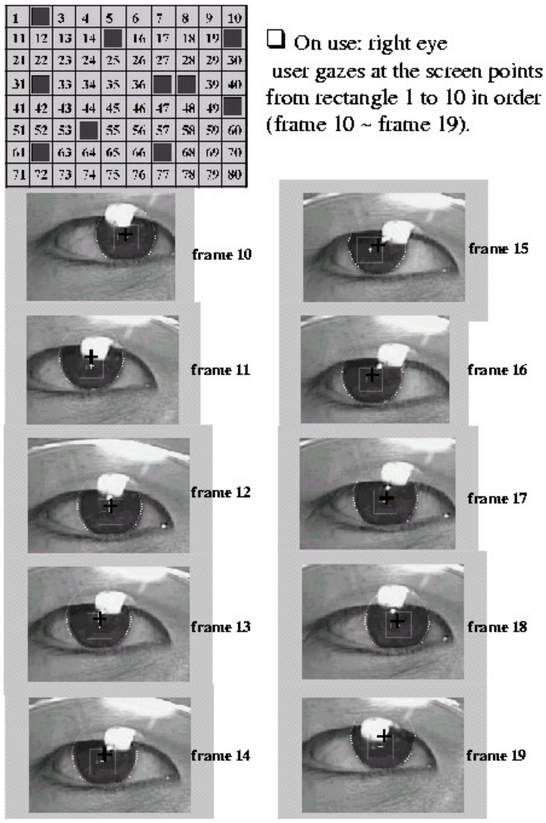

Fig. 11. Subject's random gazes : right eye

Frame	Original Point	Geometry-Based		Adaptive
		Scheme I	Scheme II	
frame 10	32	32	32	32
frame 11	15	15	15	15
frame 12	54	54	54	54
frame 13	37	37	37	37
frame 14	38	**28**	**28**	**28**
frame 15	20	20	20	20
frame 16	50	50	50	50
frame 17	67	67	67	**77**
frame 18	62	62	62	62
frame 19	2	2	2	2

Table 1. Estimated Gazing Points in 8 x 10 Screen Resolution

The proposed eye-gaze tracking methods are seen to be quite successful at screen resolutions of 3 x 3 and 4 x 5. They perform satisfactorily at 8 x 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.

5. 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. 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.

The *Longest Line Scanning* (LLS) and the *Occluded Circular Edge Matching* (OCEM) have been proposed to deal with this problem. OCEM is particularly successful in accurately detecting the iris centers in the image sequence.

Two estimation techniques, both of which are based on the geometry around the eye-gaze have been proposed, *Geometry-Based Estimation* and *Adaptive Estimation*. A small 2D mark has been adopted as the reference which is the origin of the displacement of the eye movement. It should be extended to cover the free head movement. This may require concomitant compensation and non-linear estimation.

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] 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,

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] A. H. Gee and R. Cipolla, Determining the Gaze of Faces in Images, *Technical Report*, Department of Engineering, University of Cambridge, March, 1994.

[13] 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.

[14] 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.

[15] Kin-Man Lam and Hong Yan, An Improved Method for Locating and Extracting the Eye in Human Face Images, *IEEE Proceedings of ICPR*, pp. 411-415, November, 1995.

[16] Jyh-Yuan Deng and Feipei Lai, Region-Based Template Deformation and Masking for Eye-Feature Extraction and Description, *Pattern Recognition*, Vol. 30, No. 3, pp. 403-419, 1997.

[17] Shi-Hong Jeng, Hong Yuan Mark Liao, Chin Chuan Han, Ming Yang Chern, and Yao Tsorng Liu, Facial Feature Detection using Geometrical Face Model: An Efficient Approach, *Pattern Recognition*, Vol. 31, No. 3, pp. 273-282, 1998.