# Vision-Based Eye-Gaze Tracking for Human Computer Interface

**Kyung-Nam Kim*** **and R. S. Ramakrishna***

***Department of Information and Communications,**
**Kwangju Institute of Science and Technology,**
**Kwangju, 500-712, Korea(ROK)**
**{knkim,rsr}@kjist.ac.kr**

## Abstract

Eye-gaze is an input mode which has the potential of an efficient computer interface. Eye movement has been the 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. The gaze point is estimated after acquiring the eye movement data. Preliminary experimental results are given through a screen pointing application.

## 1 Introduction

Human-computer interaction has become an increasingly important part of our daily lives. 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 Electro-Oculography [2], Limbus, Pupil and Eyelid Tracking [3, 4, 5, 6, 7, 8, 9], Contact Lens Method, Corneal and Pupil Reflection Relationship [5, 4, 8], Purkinje Image Tracking, Artificial Neural Networks [10] and 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.

## 2 Tracking of Eye Movement

The location of face and eye should be known for tracking eye movements. We assume this location information has already been obtained through extant techniques. Exact eye movements can be measured by special techniques. This investigation concentrates on tracking eye movement itself. The primary goal of this paper 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.

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 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 can be quite appropriate for people with darker iris color ( for instance, Asians ).

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

The techniques proposed in this section effectively deal with the first one, while the second is an inherently hard problem.

### 2.1 Longest Line Scanning (LLS)

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

**0-7803-5731-0/99/$10.00 ©1999 IEEE**

of the iris is elliptical in shape. The following well-known property is useful in this regard.
*The center of an ellipse lies on the center of the longest horizontal line inside the boundary of the ellipse*
The LLS algorithm is an application of this property. It can be applied to the problem of detection of the iris center.

The algorithm is given below.

```
(* 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 *)
```

**Algorithm 1:** *Longest Line Scanning* (see Figure 1)

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.

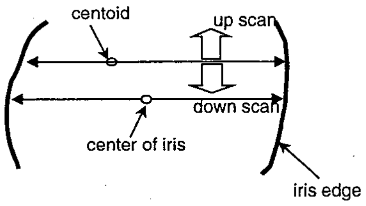


Figure 1: Principles of LLS

## 2.2 Occluded Circular Edge Matching (OCEM)

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

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 ).

```
(* Input:
        the edge image after LLS,
        $P_c$ - the centroid of iris pixels,
        $P_p$ - the midpoint, ( of horizontal 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.
```

**Algorithm 2:** *Occluded Circular Edge Matching*

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 ). 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 ). 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.

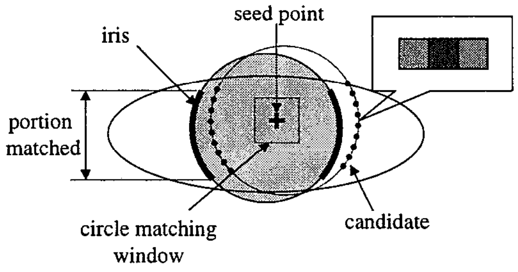


Figure 2: Matching Process in OCEM

## 3 Estimation of Gazing Point

As previous work has reported, gaze estimation with free head movement is very difficult to deal with. 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 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 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.

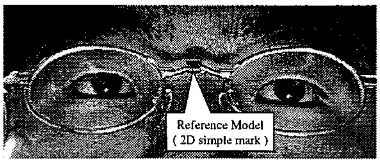


Figure 3: Reference Model: 2D Simple Mark

### 3.1 Geometry around 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. The circle shows the projection of the spherical eyeball. 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. *d* is assumed given or can be intentionally set to fixed value. $r_{ball}$ is the radius of the eyeball, which ranges from 12mm to 13mm ( according to the anthropometric data ). $\Delta_1$ and $\Delta_2$, measured in section 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 $\Delta_1$, $\Delta_2$, 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

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

We found that this $error_{image}$ can be ignored in a real experiment.

### 3.2 Estimation Algorithms

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. 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.

#### 3.2.1 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. $\Delta_{ref}$ is the displacement of the reference model.

Using equations 1 and 2, we get

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

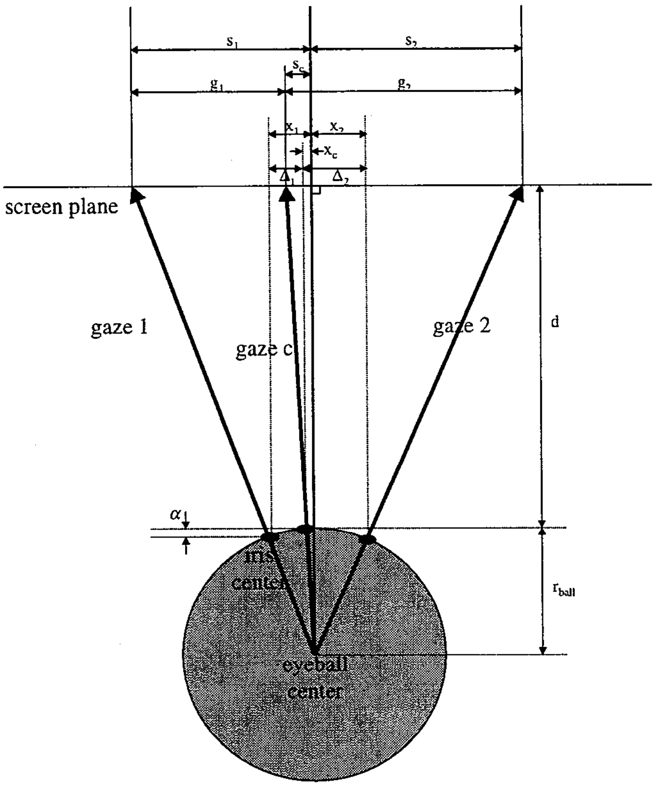


Figure 4: Geometry around Eye-Gaze

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

```
(* Input:
      $d$ - the distance from the eyeball to the screen,
      $r_{ball}$ - the radius of the eyeball,
      $\Delta_1$, $\Delta_2$ - the distance from the iris to the reference,
      $\Delta_{ref}$ - 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.
```

**Algorithm 3:** *Geometry-Based Estimation*

**3.2.2 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 4.

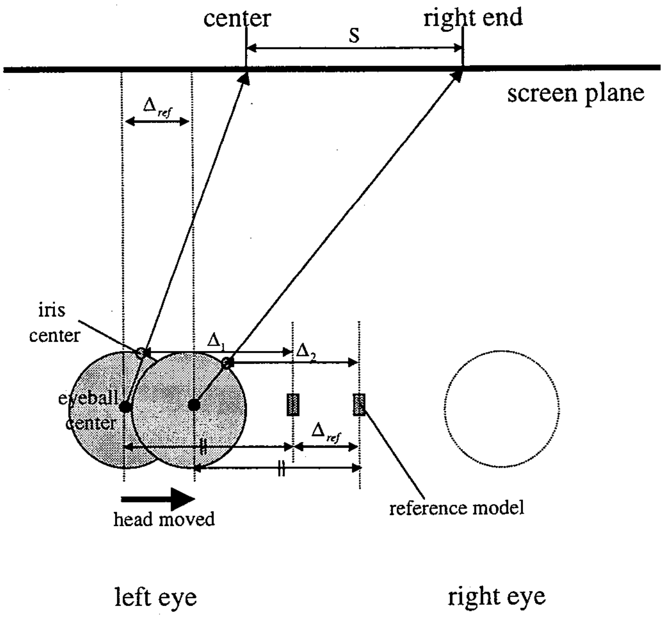


Figure 5: Geometry including Slight Head Movement

```
(* 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.
```

**Algorithm 4:** *Adaptive Estimation*

# 4 Experimental Results

In this section, preliminary experimental results pertaining to eye movement tracking and the corresponding ( estimated ) gazing points are presented, though we have performed several experiments.

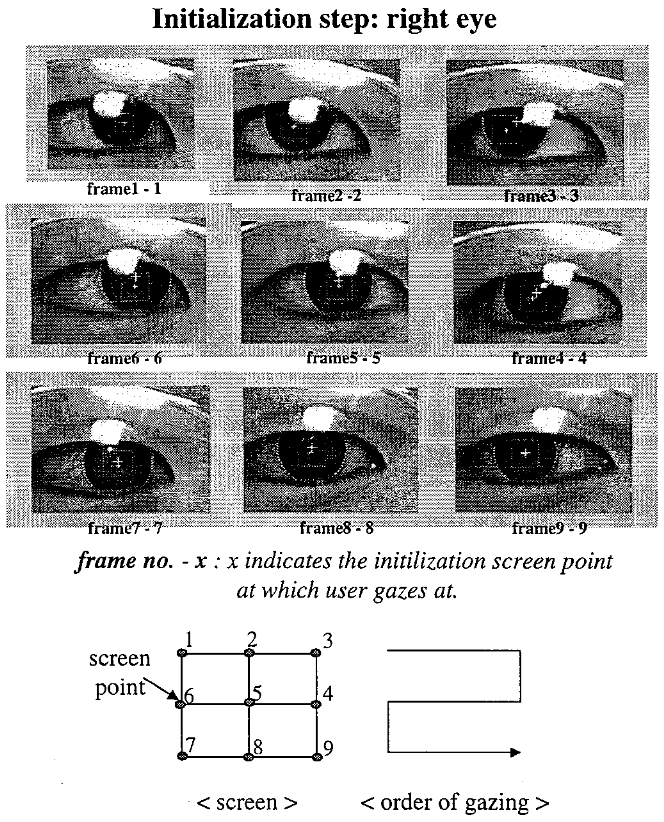
## 4.1 Preliminary Experimental Results

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.

OCEM is employed for eye movement tracking and both the Adaptive and Geometry-Based estimation techniques are utilized for estimation. Figures 6 show results of the right eye movement tracking at initialization. Each image of subject's iris being tracked, while gazing at 10 screen points randomly ( from small rectangle 1 to 10 ), is given in Figure 7. Both eyes are tracked simultaneously. (Similar results were obtained for left eye).

The frames from 1 to 9 are for the initialization step and the remainder is the data when the subject gazes at the screen points of interest.

The subject gazes at the screen points in the following order : 32 → 15 → 54 → 37 → 38 → 20 → 50 → 67 →



*frame no. - x : x indicates the initilization screen point at which user gazes at.*

Figure 6: Initialization step : right eye (different image size due to different search window)

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 employing Geometry-Based Estimation and Adaptive Estimation.

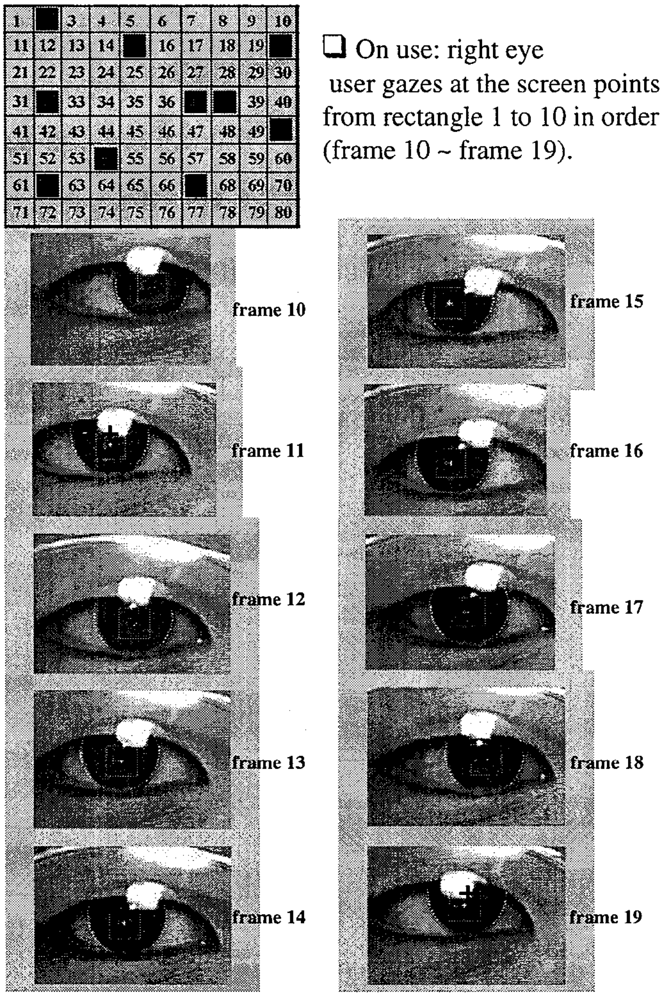
### 4.2 Analysis

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.

Table 1: 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 |



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ■ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | ■ | 16 | 17 | 18 | 19 | ■ |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | ■ | 33 | 34 | 35 | 36 | ■ | ■ | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | ■ |
| 51 | 52 | 53 | ■ | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | ■ | 63 | 64 | 65 | 66 | ■ | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |

☐ On use: right eye
user gazes at the screen points from rectangle 1 to 10 in order (frame 10 ~ frame 19).

Figure 7: Subject's random gazes : right eye

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 Conclusion

Non-intrusive vision-based eye-gaze tracking methods involving eye movement tracking (not the eye location, but 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.

More robust experiments 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, Survey 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.