Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Vision and Image Understanding 98 (2005) 25–51

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

# A novel non-intrusive eye gaze estimation using cross-ratio under large head motion

Dong Hyun Yoo, Myung Jin Chung*

*Division of Electrical Engineering, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology 373-1, Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea*

Received 15 October 2003; accepted 27 July 2004
Available online 12 October 2004

## Abstract

Eye gaze estimation systems calculate the direction of human eye gaze. Numerous accurate eye gaze estimation systems considering a user's head movement have been reported. Although the systems allow large head motion, they require multiple devices and complicate computation in order to obtain the geometrical positions of an eye, cameras, and a monitor. The light-reflection-based method proposed in this paper does not require any knowledge of their positions, so the system utilizing the proposed method is lighter and easier to use than the conventional systems. To estimate where the user looks allowing ample head movement, we utilize an invariant value (cross-ratio) of a projective space. Also, a robust feature detection using an ellipse-specific active contour is suggested in order to find features exactly. Our proposed feature detection and estimation method are simple and fast, and shows accurate results under large head motion.

*Keywords:* Non-intrusive eye gaze estimation; Large head motion; Cross-ratio; Robust pupil detection

* Corresponding author.
*E-mail address:* mjchung@ee.kaist.ac.kr (M.J. Chung).

1077-3142/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2004.07.011

## 1. Introduction

Eye gaze plays an important role in human communication. It is one of cues when attempting to understand an individual's intentions. Eye gaze can be used as an interface between humans and computerized devices. The eye gaze interface gives us a more natural way to interact with devices than keyboard and mouse. There are several approaches of estimating the direction of eye gaze: reflection of light, electrical skin potential, and contact lenses. These methods are categorized as either intrusive or non-intrusive methods. Non-intrusive methods are preferred because they tend to be more comfortable to use than intrusive methods. Unfortunately, there are still many unresolved problems preventing the use of non-intrusive eye gaze systems in actual applications. The most prevalent of these are: accuracy, restricted head movement, robustness, and ease of calibration. While there have been some reports of systems having accuracy within 1°, many of these confine the range of head movement by the user assuming that the head is fixed, or that the head only moves within a limited small range. Other systems can deal accurately with a certain amount of head movement, but they require additional information, such as knowing the distance between the camera and the eye. To measure this distance, a stereo camera or ultrasonic sensor must be utilized, which complicates the eye gaze system considerably. Furthermore, any complex calibration causes discomfort for the user.

The method proposed in this paper is non-intrusive and is based on the reflection of light, but it differs from existing systems in that it consists of five infrared light-emitting diodes (LEDs) and two cameras. We attach four LEDs to each corner of a monitor, and their glints are utilized to find a mapping function from the glints to an eye gaze point. The fifth LED which is located at the center of one of the camera lenses is crucial to consider large head movement. To estimate where the user's gaze will fall, we utilize the invariant value of projective space, a method that is simple, fast, and accurate even when there is significant head movement by the user.

Exact feature detection is essential to eye gaze estimation systems because feature detection errors are directly related to system accuracy. The pupil is a useful element, but its detection is difficult because its intensity is closely similar to that of the iris. The bright-eye effect suggested by Hutchinson improves the possibility of pupil detection and so it is used in conjunction with a dark-eye image. The pupil's boundary is identified by an active contour method and is fitted with an ellipse to reduce the effect of noise. The proposed method results in robust and precise feature extraction.

Related works and their limitations are discussed in Section 2. An eye model and the definition of eye gaze are explained in Section 3. Our proposed eye gaze estimation system is introduced in Section 4, and an eye gaze estimation method using the proposed system is described in Section 5. In Section 6, a robust feature extraction method is proposed. A face tracking system is shown in Section 7. Experiment results are shown in Section 8. Finally, conclusions are given in Section 9.

## 2. Related works

The methods for estimating eye gaze direction have been categorized into two groups: intrusive and non-intrusive. The intrusive methods perform very well and are reported to be more accurate than the non-intrusive methods currently being practiced. However, the intrusive methods have a severe shortcoming in that the the eye gaze detection devices that must be worn by the users cause discomfort and restrict their movement. Alternatively, many novel non-intrusive methods have recently been proposed with accuracy levels showing drastic improvement. However, there are still numerous unresolved problems related to their use. Let's now look at each type of method in greater detail.

(1) Intrusive methods

Some researchers have attempted to develop eye gaze estimation systems using electric skin potential [1,2]. Methods based on electric skin potential measure body signals and estimate the eye's movement using electrodes attached to the skin around the eye. The contact lens-based approach uses a special contact lens. The exact position of the lens can be recorded by implanting a tiny induction coil into the lens and placing high-frequency electro-magnetic fields around the user's head. Kim et al. [3] suggested a vision-based approach. They estimated the eye motion by tracking the limbus in images, and measured the head's pose with a magnetic sensor attached to the head. By these two results, the eye gaze can be obtained.

(2) Non-intrusive methods

Non-intrusive methods can be classified into two categories: active and passive. Active methods use the reflection of light and vision processing, and passive methods are based solely on vision theory.

Hutchinson [4] computed the direction of the eye gaze by using the glint made by a LED and the center of the pupil in the image captured by a camera. If a user looks directly at the LED, the distance between the glint and the center of the pupil is small. On the other hand, if the user looks away from the LED, the distance is increased. Morimoto et al. [5] utilized the mapping functions from a glint-pupil vector to a screen coordinate of gaze point. The functions are represented by two second-order polynomial equations, and the coefficients of the equations are determined by a calibration procedure. These methods demonstrates enhanced performance when the user's head is in a fixed position, but no if there is any degree of head movement. To improve the performance, Ji and Zhu [6] suggested a neural network-based method. They computed an eye gaze point by mapping pupil parameters to screen coordinates using generalized regression neural networks (GRNN). This system uses pupil parameters, pupil-glint displacement, ratio of the major to minor axes of the ellipse fitting the pupil, ellipse orientation, and glint image coordinates. These methods work very well if the head is in a fixed position. However, because the estimated result varies according to head movement, accuracies of these methods are risked during free head movement.

There are some methods coping with the user's head movement by using additional information. Sugioka et al. [7] formulated the equations for calculating eye gaze direction using the positions of the pupil and the glint. However, the distance between the camera and the eye must be known in order to solve the equations. To solve this problem, they attempted to measure the distance between the camera and the eye using an ultrasonic sensor. Ohno et al. [8,9] proposed a head-free eye gaze tracking system. They developed the eye gaze tracking system using three cameras and one LED. They reported that the 3D position of the center of the cornea curvature can be computed using one camera and a Purkinje-image. However, the distance between the eye and the camera was still required in order to obtain the position, so they utilized a stereo camera system to detect the user's head and to measure the distance. Liu [10] and Pastoor et al. [11] developed an eye gaze estimation system that compensated for head movement by using two cameras: a gaze tracking camera and an eye tracking camera. The gaze tracking camera detects the eye gaze by monitoring the pupil and the reflective glint. The gaze point is computed using the eye gaze compensated by the eye position estimated by the eye tracking camera.

Image-based methods only utilize images to estimate the eye gaze direction. Matsumoto and Zelinsky [12,13] employed a stereo camera system to determine both the eye gaze direction and head direction. The two directions are then combined to estimate the gaze direction. Wang and Sung [14] suggested an accurate image-based eye gaze estimation system using two cameras. One camera is a gaze camera to estimate eye gaze relative to head pose, and the other is a pose camera to obtain head pose. In this method, they estimated the eye gaze relative to head pose using one-circle algorithm.

As discussed, the accuracy of many methods is jeopardized because of their inability to cope with head motion. Although various methods have been applied under significant head movement, they often require additional information to obtain peak efficiency, such as the distance between the camera and the user's eye. To measure this distance, stereo cameras or ultrasonic sensors must be utilized, complicating the eye gaze system by requiring complex calibration processes. Moreover, minor sensor errors at the distance cause the overall estimation results to be lacking in precision.

In this paper, five corneal reflections are generated in order to allow greater head movement with affecting the results in a negative manner. Our proposed method does not require any knowledge concerning the geometric positions of the cameras, the monitor, or the eye. Also, the cameras do not need to be calibrated. Therefore, this method can be easily applied to actual applications.

## 3. Eye model and eye gaze

An eye is a sphere with an average diameter of 24 mm. The outer layer of the eye is formed by the sclera, which is a sturdy whitish opaque membrane that surrounds the cornea and covers the outer portion of the eyeball. It consists of firm connective tissue continuous in its anterior part with a transparent membrane, known as the cornea (Fig. 1). The shape of the cornea is assumed to be a segment of an imaginary sphere with a radius of about 8 mm. The surface of the sclera is coarser than that of

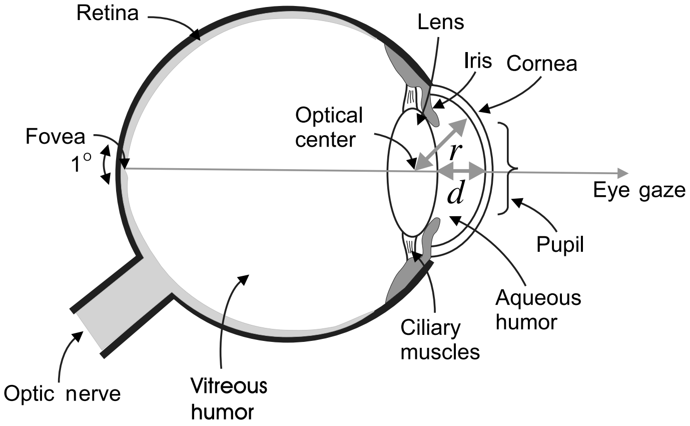


Fig. 1. Eye structure (Illustration taken from [15]).

the cornea, and it also has a smaller curvature. The iris is an opaque tissue with a hole, namely, a pupil, at its center. The diameter of the pupil varies from 2 to 8 mm in response to different ambient light levels and some psychological factors. The innermost layer, the retina, directly receives the photic stimuli. The thickness of the retina in its central part, the macula lutea, is about 0.1 mm. The macula lutea is yellow and is occupied mainly by cones. Within the macular lutea lies the fovea, the part of the retina with the maximum resolving power. The diameter of the fovea is about 0.4 mm (about 1°). The accuracy of the eye gaze system may be limited to 1° since the width of the fovea is approximately equivalent to a view-field of 1°. Eye gaze is determined by the line connecting the fovea to the center of the lens.

Gaze direction is relative to the center of the eyeball. To use eye gaze direction as an interface, the gaze point (fixation point) must be computed. The fixation point is determined by the line of sight and the object crossed by the line. Many conventional eye gaze estimation systems compute the fixation point by calculating the position and the orientation of the eyeball in a 3-dimensional space. These systems require knowledge of the exact position of the eyeball and the orientation of the eye determined by the movement of the eye. The position is combined with the orientation, and the total eye gaze direction relative to the world coordinate is computed in 3D. To obtain the fixation point, the 3D pose of the object relative to the world coordinate should be known.

In this research, a simplified eye model is used. It is assumed that the eye is spherical with an average radius, and that the cornea is a segment of the imaginary sphere with a radius of $r$ mm. The pupil is located at the center of the cornea. The distance between the cornea and the pupil is $d$ mm.

We need to consider the range of head movement when a user is looking at a monitor. Assume a monitor is located on the desk, and a user sits about 40–60 cm in front of the monitor. Head movement is restricted to a particular region while gazing at the monitor. The rotation of the eye is not rigorous because the gaze point of the user is within the monitor screen. Furthermore, the distance between the head and the monitor is greater than the size of the monitor, so the range of eye rotation is small. The range of the head movement can be defined referring to ergonomics [16–19];

(1) The top of a monitor should be slightly above eye level.
(2) Viewing distance is from 400 to 600 mm.
(3) Lateral viewing boundaries should not exceed 30° to either side of the body's center-line.

## 4. System description

To accommodate for ample head movement without risking imprecision, our method requires five IR LEDs and two cameras, as depicted in Fig. 2. Four LEDs are attached to corners of the monitor to produce reflections on the surface of the cornea, and the fifth LED is located at the center of the lens of a zoom lens camera to instigate a bright-eye effect (see Fig. 3). The four LEDs on the monitor are turned

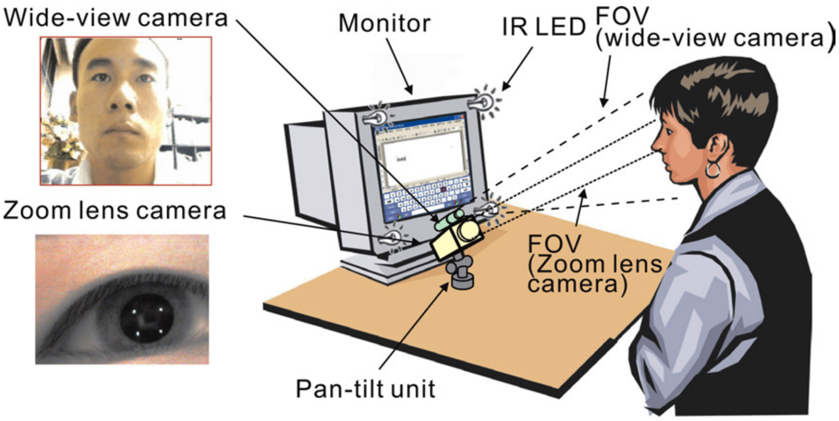


Fig. 2. System configuration: a monitor with four IR LEDs attached to the corners and two cameras, one of which is with an IR LED attached to its lens.

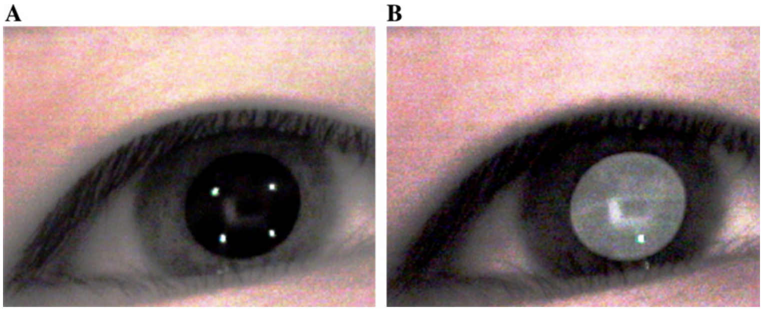


Fig. 3. Input images captured by a camera: (A) dark eye image (corneal reflection) and (B) bright eye image.

on and off together, while the one on the camera is activated independently. These LEDs are controlled through a computer's parallel port. The switching of the LEDs is synchronized with the image capture so as to obtain two images, as in Fig. 3. The LEDs on the monitor generate the reflections on the cornea, and four glints are then shown in the captured image. Since the glints are made by the LEDs placed on the monitor, the polygon made by the glints can be assumed to be the reflection of the monitor. These glints are used to estimate eye gaze direction by mapping from the image space to the monitor screen. The LED placed on the camera creates a diffusion of light on the retina to generate the bright-eye effect, and to make one glint on the cornea.

The cameras are mounted on a pan-tilt unit so that both a face and an eye can be tracked simultaneously. One camera has a high-magnification zoom-lens to capture images of the magnified eye, as seen in Fig. 3. The larger the size of the captured eye, the greater the system accuracy. However, it is somewhat difficult to track the magnified eye by only the zoom lens camera. The field of view (FOV) of the zoom-lens camera is very narrow, so slight head movements cause the eye to disappear from the FOV. Therefore, a wide-view camera is needed to track the eye robustly when the zoom-lens camera fails to do so. A face tracking subsystem with a wide-view camera is also essential. The face tracking system tracks a face continuously, and informs the pan-tilt unit of the position of the eye, so the pan-tilt unit can control the direction of the cameras to capture the eye.

## 5. Estimation of gaze point

The proposed system calculates a gaze point by utilizing the property of projective space. When the LEDs of the monitor are turned on and that of the camera is turned off, the light is reflected on the surface of the cornea generating glints (see Fig. 3A). Conversely, Fig. 3B is made by reverse operation of LEDs. It is assumed that the five glints and the pupil center are observed by a camera, a reasonable assumption considering that a user sees the monitor within the space defined in Section 3. Usually, because the user is situated directly in front of the monitor, the region of head rotation will not be extensive. Fig. 4 shows our concept for estimating eye gaze direction. In the figure, $\mathbf{p}$ is the center of the pupil, and $\mathbf{g}$ is eye gaze point, which is where the user looks. The line of sight intersects the screen at $\mathbf{g}$. Our purpose is to estimate the gaze point from our knowledge of the positions of the five glints, the pupil and the actual size of the monitor screen. We attempt to solve this problem using the invariant value of the projective space. In the figure, assume that there is a virtual tangent plane on the surface of the cornea. The points $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, and $\mathbf{v}_4$ on the plane are the projection points of the four IR LEDs (LED$_1$, LED$_2$, LED$_3$, and LED$_4$) of the monitor. These points are referred to as the virtual projection point, and their projection center is the center of the cornea sphere. If the virtual projection points are approximately coplanar, then the polygon made by the virtual projection points is the projection of the monitor's screen. The virtual projection points and the pupil are projected into five points ($\mathbf{u}_{\mathbf{v}_1}, \mathbf{u}_{\mathbf{v}_2}, \mathbf{u}_{\mathbf{v}_3}, \mathbf{u}_{\mathbf{v}_4}$, and $\mathbf{u}_\mathbf{p}$) in the image plane of the camera.

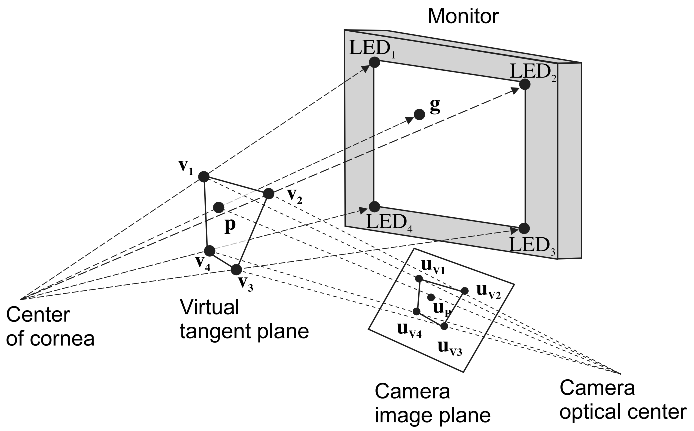


Fig. 4. The relation between the IR LEDs of the monitor (LED$_1$, LED$_2$, LED$_3$, and LED$_4$) and the virtual projection points ($\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, and $\mathbf{v}_4$). The points are the projection of IR LEDs onto the surface of the cornea. **p** is the pupil position in an image and **g** is the gaze point of the eye.

Therefore, there are two projective transforms from the monitor screen to the image plane. If the virtual projection points are approximately coplanar, then the screen coordinates of **g** can be estimated by using the projective invariant. It will be described in Section 5.2.

To obtain the virtual projection points from the glints, a detailed investigation of their geometric relation is needed. Simplifying the problem to a 2-dimensional case (see Fig. 5) will make it easier to understand. Assume that there is an eye in front of a monitor with LEDs and a camera positioned to capture the eye. The user looks at one gaze point (**g**) of the screen. In this figure, the points $\mathbf{r}_1$, $\mathbf{r}_2$, and **c** are the glints made by LED$_1$, LED$_2$, and LED$_C$, respectively, on the surface of the cornea. The reflection of light generates the glints $\mathbf{u}_{\mathbf{r}_1}$, $\mathbf{u}_{\mathbf{r}_2}$, and $\mathbf{u}_\mathbf{c}$ in images. Since the glints are

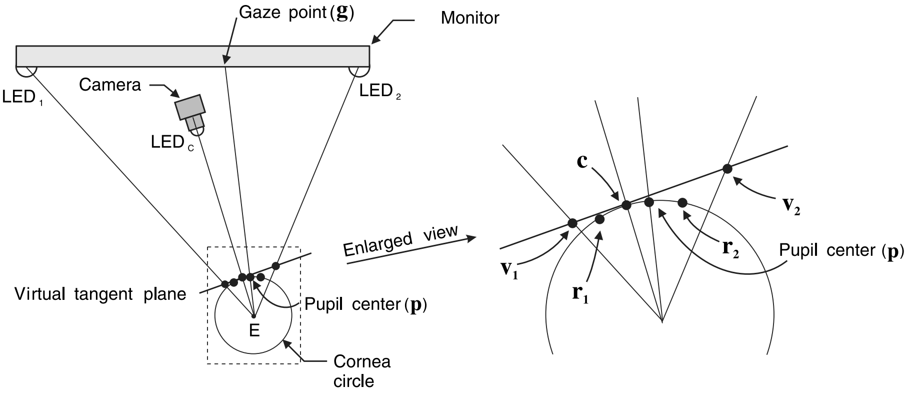


Fig. 5. Geometrical relation of the reflection. $\mathbf{r}_1$, $\mathbf{r}_2$, and **c** are the glints of LED. $\mathbf{v}_1$ and $\mathbf{v}_2$ are the virtual projection points.

produced by reflection, these features are not suitable for use with the property of projective space.

To resolve this problem, the virtual projection points made by the projection are computed. A virtual plane is tangent to the cornea at point **c**. Points $\mathbf{v_1}$ and $\mathbf{v_2}$ on the tangent plane are the cross-points of the tangent plane and the line from the LEDs to the center of the cornea. Glint **c** is on the line connecting the optical center of the camera and the center of the cornea because the $\mathrm{LED_C}$ of the camera is located at the center of the lens. The three points $\mathbf{v_1}$, $\mathbf{v_2}$, and **c** are the cross points of the tangent plane and the lines crossing the same vanishing point which is the center of the cornea. Therefore, the relation between the LEDs on the monitor and the glint in image is considered as two projections: from the monitor to the plane tangent to the cornea, and from the tangent line to the image plane of the camera. Also, the center of the pupil is projected to one point ($\mathbf{u_p}$) in the image plane. As such, we can assume that the two virtual points and the center of the pupil in the image are the projected points of the two LEDs and the gaze point on the screen. Therefore, the gaze point is estimated by the invariant value of the projective space. This concept can also be applied to 3-dimensional cases. The image coordinates of virtual projection points cannot be directly observed but can be estimated using the method described in Section 5.1.

### *5.1. Computation of the virtual projection point*

Virtual projection points should be computed in order to utilize the property of the projective space. It is somewhat challenging to compute the virtual points from real reflected points because the relation is too complicated. Fortunately, the range of head movement can be confined to a specific region in situations where users are seated in front of a monitor and are looking directly at it. Typically, the user is seated 30–50 cm away from the monitor, and the range of movement is less than 30 cm. By confining the head movement, the virtual projection points can be computed approximately. Fig. 6 is a magnified view of Fig. 5. $x_{\text{camera}}$ is the distance between the camera and the center of the cornea, and $x_{\text{LED}}$ is the distance between the LED and the center of the cornea. $x_{R_1}$ and $x_{R_2}$ are the distances between the point $\mathbf{r_1}$ and the camera and between the point $\mathbf{r_1}$ and the LED. If $u_{r_1}, u_c$, and $u_{v_1}$ are the coordinates of the glints ($\mathbf{r_1}$ and **c**) and virtual projection point. In the image of which coordinate system origin is $\Sigma_{\mathrm{i}}$, then

$$u_{\mathrm{c}} = \frac{f}{\tan\phi}, \tag{1}$$

$$u_{r_1} = \frac{rf\cos(\theta_1+\phi) - fx_{\text{camera}}\cos\phi}{r\sin(\theta_1+\phi) - x_{\text{camera}}\sin\phi}, \tag{2}$$

$$u_{v_1} = \frac{rf(\cos\phi - \tan(\theta_1+\theta_2)\sin\phi) - fx_{\text{camera}}\cos\phi}{r(\sin\phi + \tan(\theta_1+\theta_2)\cos\phi) - x_{\text{camera}}\sin\phi}, \tag{3}$$

where $r$ is the radius of the cornea, $f$ is the focal length of the camera, and $\phi$ is the orientation of the camera relative to the cornea.

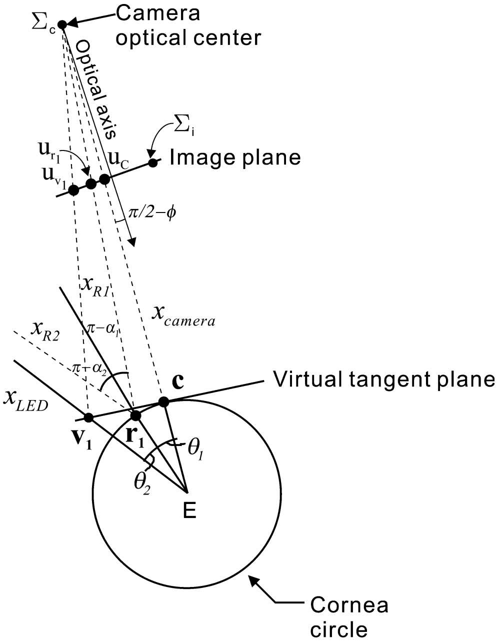


Fig. 6. How to compute a virtual projection point. $\Sigma_c$ is a coordinate system origin which is located at the optical center of the camera, and $\Sigma_i$ is an image coordinate system origin.

$\theta_1$ is the angle between the points $\mathbf{c}$ and $\mathbf{r}_1$, and $\theta_2$ is the angle between the points $\mathbf{r}_1$ and $\mathbf{v}_1$. When the user operates a computer, the two angles are very similar because the distance between the monitor and the eye is longer than the distance between the camera and the LED. Conforming with trigonometry

$$\frac{x_{R_1}}{\sin\theta_1} = \frac{x_{\text{camera}}}{\sin\alpha_1}, \tag{4}$$

$$\frac{x_{R_2}}{\sin\theta_2} = \frac{x_{\text{LED}}}{\sin\alpha_2}. \tag{5}$$

According to the law of reflection, the incident angle is equal to the reflection angle, so

$$\sin\theta_2 = \frac{x_{\text{camera}}}{x_{\text{LED}}}\frac{x_{R_2}}{x_{R_1}}\sin\theta_1. \tag{6}$$

Because the radius of the cornea is smaller than the distance of $x_{\text{camera}}$, $x_{\text{LED}}$, $x_{R_1}$, and $x_{R_2}$, $x_{\text{camera}}$, and $x_{\text{LED}}$ are close to $x_{R_1}$ and $x_{R_2}$, respectively. Eq. (6) is approximated to the following equation:

$$\sin\theta_2 \approx \sin\theta_1. \tag{7}$$

Because the range of the angles is $[0, \pi/2)$, the angles are almost identical. The angle $\phi$ is close to $\pi/2$ because the high-magnification camera has ability to track a relatively small eye. By approximation, Eqs. (1)–(3) can be simplified as follows:

$$u_c \approx 0, \tag{8}$$

$$u_{r_1} \approx \frac{-rf\sin\theta_1}{r\cos\theta_1 - x_{\text{camera}}}, \tag{9}$$

$$u_{v_1} \approx \frac{-rf\tan 2\theta_1}{r - x_{\text{camera}}}. \tag{10}$$

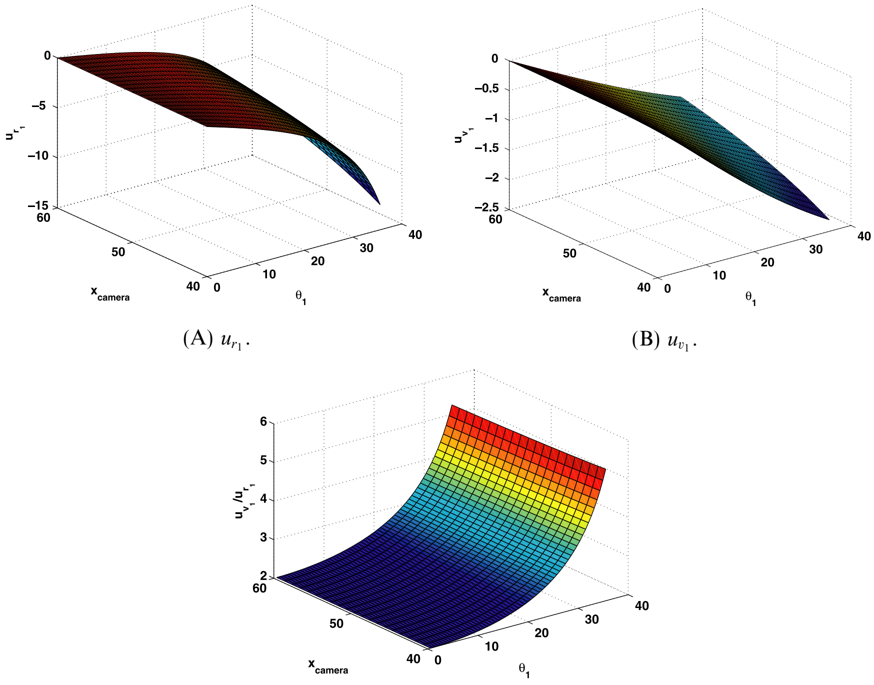
The equations are the functions of $x_{\text{camera}}$ and $\theta_1$. The values of the features according to $x_{\text{camera}}$ and $\theta_1$ are plotted in Fig. 7. $\theta_1$ is less than 10° within the previously confined, so the ratio of $u_{v_1}$ and $u_{r_1}$ is approximately 2.

If $\theta_1$ is small enough, then Eqs. (9) and (10) can be simplified as follows:

$$u_{r_1} \sim \frac{-rf\theta_1}{r - x_{\text{camera}}}, \tag{11}$$

$$u_{v_1} \sim \frac{-2rf\theta_1}{r - x_{\text{camera}}}. \tag{12}$$

Consequently, we have the relating equation: $u_{v_1} = 2u_{r_1}$. However, $u_c$ can be deviated from the center of an image in an actual situation. To compensate for this problem,



(A) $u_{r_1}$.

(B) $u_{v_1}$.

(C) $u_{v_1}/u_{r_1}$.

Fig. 7. The variation of the features in the image according to the eye's location.

we eliminate the offset by subtraction of $u_c$ from $u_{r_1}$ and $u_{v_1}$. Therefore, the virtual projection point can be computed by the following equation:

$$u_{v_1} = u_c + \alpha(u_{r_1} - u_c), \tag{13}$$

where $\alpha$ is a constant close to 2. The value is determined through the simple calibration process described in Section 5.5. Fig. 8 depicts the computation of the virtual projection points from the glints.

### 5.2. Estimation of an eye gaze point by cross-ratio

The gaze point is estimated using a cross-ratio invariant to the projective transform. Fig. 9 demonstrates how the cross-ratio in an image is computed. The points

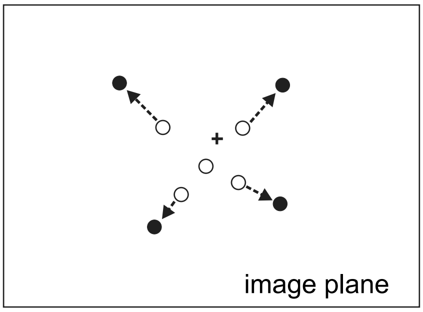


Fig. 8. Computation of virtual projection points in an image plane. The white circles (○) are the glints, and the black circles (●) are the virtual projection points of the glints. The cross (+) marks the pupil

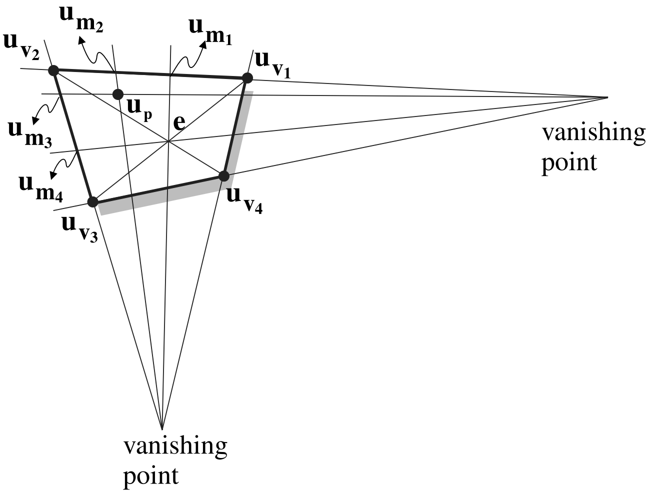


Fig. 9. How a cross-ratio in an image is computed. $\mathbf{u}_{v_1}$, $\mathbf{u}_{v_2}$, $\mathbf{u}_{v_3}$, and $\mathbf{u}_{v_4}$ are the virtual projection points on the cornea. $\mathbf{u_p}$ is the center of the pupil in the image. $\mathbf{e}$ is an intersection of diagonal lines of the polygon $\mathbf{u}_{v_1}$, $\mathbf{u}_{v_2}$, $\mathbf{u}_{v_3}$, and $\mathbf{u}_{v_4}$.

$\mathbf{u}_{\mathbf{v}_1}, \mathbf{u}_{\mathbf{v}_2}, \mathbf{u}_{\mathbf{v}_3}$, and $\mathbf{u}_{\mathbf{v}_4}$ are the virtual projection points worked out by the method described in the previous section. $\mathbf{u}_\mathbf{p}$ is the center of the pupil, and $\mathbf{e}$ is the cross-point of the line $\overline{\mathbf{u}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_3}}$ and the line $\overline{\mathbf{u}_{\mathbf{v}_2}\mathbf{u}_{\mathbf{v}_4}}$.

The $x$-coordinate of the eye gaze point is computed by the following method. First, a vanishing point must be calculated by line $\overline{\mathbf{u}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_4}}$ and line $\overline{\mathbf{u}_{\mathbf{v}_2}\mathbf{u}_{\mathbf{v}_3}}$, allowing the two points $\mathbf{u}_{\mathbf{m}_1}$ and $\mathbf{u}_{\mathbf{m}_1}$ to be calculated. The vanishing point is the cross-point of $\overline{\mathbf{u}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_4}}$ and $\overline{\mathbf{u}_{\mathbf{v}_2}\mathbf{u}_{\mathbf{v}_3}}$. $\mathbf{u}_{\mathbf{m}_1}$ is the cross-point of line $\overline{\mathbf{u}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_2}}$ as well as the line that connects point $\mathbf{e}$ with the vanishing point. $\mathbf{u}_{\mathbf{m}_2}$ is the cross-point of the line $\overline{\mathbf{u}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_2}}$ and the line that connects $\mathbf{u}_\mathbf{p}$ with the vanishing point. If $\mathbf{u}_{\mathbf{v}_\mathbf{i}} = (x_i^v, y_i^v)$ $(i = 1, 2, 3, 4)$ and $\mathbf{u}_{\mathbf{m}_\mathbf{i}} = (x_i^m, y_i^m)$ $(i = 1, 2, 3, 4)$ are the coordinates of each point in the image, the equation for the cross-ratio of the four points on the line $\overline{\mathbf{u}_{\mathbf{v}_1}\mathbf{u}_{\mathbf{v}_2}}$ is

$$CR^x_{\text{image}} = \frac{(x_1^v y_1^m - x_1^m y_1^v)(x_2^m y_2^v - x_2^v y_2^m)}{(x_1^v y_2^m - x_2^m y_1^v)(x_1^m y_2^v - x_2^v y_1^m)}. \tag{14}$$

Similarly, the cross-ratio of the monitor screen, shown in Fig. 10, is calculated using the following equation:

$$CR^x_{\text{screen}} = \frac{(w - \frac{w}{2})\hat{x}_g}{(w - \hat{x}_g)\frac{w}{2}} = \frac{\hat{x}_g}{w - \hat{x}_g,} \tag{15}$$

where $w$ is the width of the monitor screen and $\hat{x_g}$ is the $x$-coordinate of the estimated eye gaze point $\mathbf{g}$.

These cross-ratios are equal because the cross-ratio is invariant in projective space. Subsequently, we are able to obtain the $x$-coordinate of the estimated eye gaze point

$$\hat{x}_g = \frac{w \cdot CR^x_{\text{image}}}{1 + CR^x_{\text{image}}}. \tag{16}$$

The $y$-coordinate of the eye gaze point is computed in the same manner. The cross-ratio of the image is

$$CR^y_{\text{image}} = \frac{(x_2^v y_3^m - x_3^m y_2^v)(x_4^m y_3^v - x_3^v y_4^m)}{(x_2^v y_4^m - x_4^m y_2^v)(x_3^m y_3^v - x_3^v y_3^m)}. \tag{17}$$

The cross-ratio of the screen is

$$CR^y_{\text{screen}} = \frac{(h - \frac{h}{2})\hat{y}_g}{(h - \hat{y}_g)\frac{h}{2} = \frac{\hat{y}_g}{h - \hat{y}_g}} \tag{18}$$

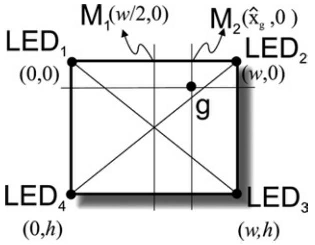


Fig. 10. The screen of a monitor. $w$ is the width of the screen and $h$ is its height. $\mathbf{g}$ is the estimated eye gaze point and its position is $(\hat{x_g}, \hat{y_g})$.

where $h$ is the height of the monitor screen and $\hat{y}_g$ is the $y$-coordinate of the estimated eye gaze point $\mathbf{g}$. The $y$-coordinate of the eye gaze point is

$$\hat{y}_g = \frac{h \cdot CR^y_{\text{image}}}{1 + CR^y_{\text{image}}}. \tag{19}$$

### 5.3. Simulation of the constancy of the proposed method

To confirm the constancy of the estimated gaze point under head motion, we carried out a simulation. We assumed the simulation environment as shown in Fig. 11 in which there is a monitor, a camera, and an eye (cornea). The user is 40–50 cm in front of the monitor. The eye is modeled as in Fig. 1. The eyeball is a sphere with a radius of 12 mm and it is understood that the radius of the cornea ($r$) is 8 mm, and the distance from the cornea to the lens ($d$) is 2.794 mm from anthropometric data. These values reflect the average dimension of an adult'eye. In this simulation, the gaze point is fixed at one point of the monitor screen, and the location of the eye is changed in the $x$-direction (from left to right). Fig. 12 shows the cross-ratio and the estimated gaze point according to the eye's location when the positions of the glints in the image are used to compute the cross ratio. The cross-ratio and the estimated result vary significantly according to the eye's location with the amount of the variation of the estimated gaze point being about 35 mm. Assuming the monitor has a width of 300 mm and a resolution of 1024 pixels, this variation means the maximum error may be about 119 pixels. Moreover, the estimated value is different from the ground truth value (0 mm), so the value should be scaled and offset.

A similar simulation was also carried out using the proposed method of the virtual points and the cross-ratio. The result is shown in Fig. 13. The cross-ratio varies in a small range, and the estimated result changes by less than 1 mm. Note that because the estimated value is close to the ground truth value (0 mm), no scale or offset

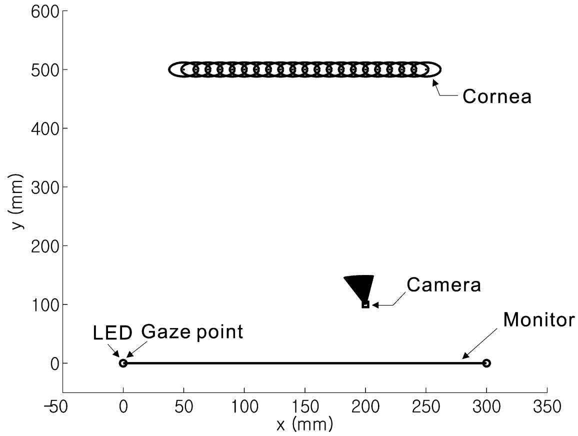
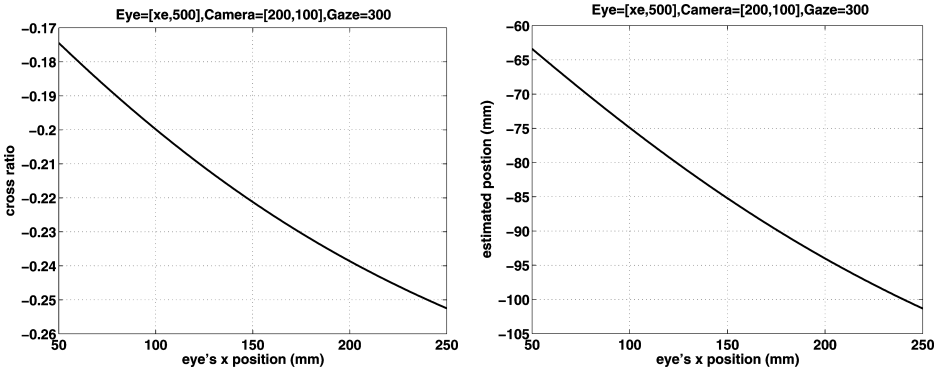


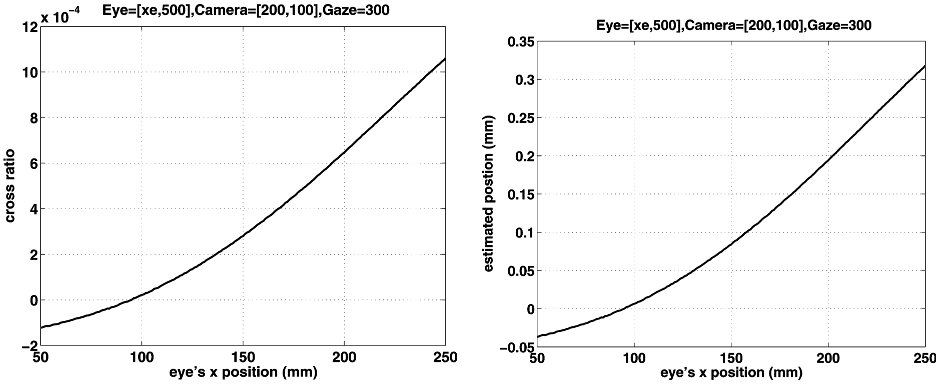
Fig. 11. Simulation environment.



(A) Cross-ratio.

(B) Estimated gaze point.

Fig. 12. The variation of the cross-ratio according to the eye's location. The positions of the glints in the image are used to compute the cross-ratio.



(A) Cross-ratio.

(B) Estimated gaze point.

Fig. 13. The variation of the cross-ratio according to the eye's location. The positions of the virtual projection points are used to compute the cross-ratio.

is needed. Therefore, this eye gaze estimation method can be used when there is substantial head movement.

### *5.4. Sensitivity study on the computation of the virtual projection point*

In Section 5.1, the virtual projection points of glints are approximately computed using Eq. (13). Because the computation is not quite exact, we have to investigate how the inaccuracies resulting from the estimated virtual projection points will affect the precision of the gaze points. In Eq. (13), the virtual projection point varies according to $\alpha$. Accordingly, we check the estimated gaze point changing $\alpha$ from 1.8 to 2.5. The variation of the error of the estimated gaze point is depicted in Fig. 14. Usually, $\alpha$ is less than 2.5 because the $\theta_1$ is not greater than 20° under an average operating environment (see Fig. 7). An estimation error of about 15 mm may be generated according to the cornea position and gaze direction.

### 5.5. Calibration

In Eq. (13), $\alpha$ is close to 2, as mentioned before, but it may differ slightly according to the user. To increase the accuracy of the proposed method, a calibration process to compute the exact value of the coefficient can be used.

When a user looks at a LED on the monitor, the center of the pupil and the glints caused by the LEDs of the monitor and the camera are detected. Because the user looks directly at the LED, the position of the virtual projection point of the glint should be same as the pupil's center. With this constraint, we can compute the coefficient by the following equation:

$$\alpha = \frac{d(\mathbf{u_p}, \mathbf{u_c})}{d(\mathbf{u_{r_1}}, \mathbf{u_c})}, \tag{20}$$

where $d(\mathbf{x_1}, \mathbf{x_2})$ is the Euclidean distance between the points $\mathbf{x_1}$ and $\mathbf{x_2}$. $\mathbf{u_p}$, $\mathbf{u_c}$, and $\mathbf{u_{r_1}}$ are the image coordinates of the pupil center, the glint made by the LED of the camera, and the glint made by the LED of the monitor, respectively. Because there are four virtual projection points in our system, all four coefficients are required. The process is repeated four times as the user looks at each LED of the monitor.

## 6. Robust feature extraction

To estimate the precise eye gaze point, the five glints made by the light reflections and the center of the pupil must be obtained from the image sequences. Exact feature extraction is imperative to achieve high-performance from any eye gaze estimation system. Furthermore, the method should be able to operate swiftly so that the system can work in real time. In this section, a robust and rapid feature extraction method is proposed.

### 6.1. Detection of glints

The glints made by the corneal reflections can be detected by the following image processing method because their intensity values are usually the highest in the image.

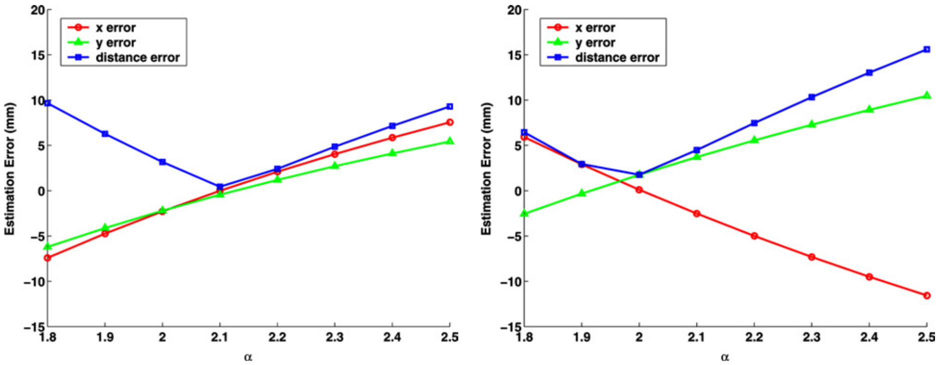
*Step 1: Segment the input image (dark-eye image) by a threshold to obtain a binary image.*

An input image, as in Fig. 15A, is segmented by a threshold.

The threshold is determined through experiments. Fig. 15B shows the segmented result.

*Step 2: Process the binary image with binary morphology.*

In this binary image, there are four 1-value blobs made by the reflection of infrared light, and there might exist extra 1-value blobs caused by noise and different



(A) Cornea center = (152, 112, and 500). (B) Cornea center = (10, 0, and 500).

Fig. 14. Sensitivity study. The camera is at (152, 0, and 100) and the gaze point is (152, 117).

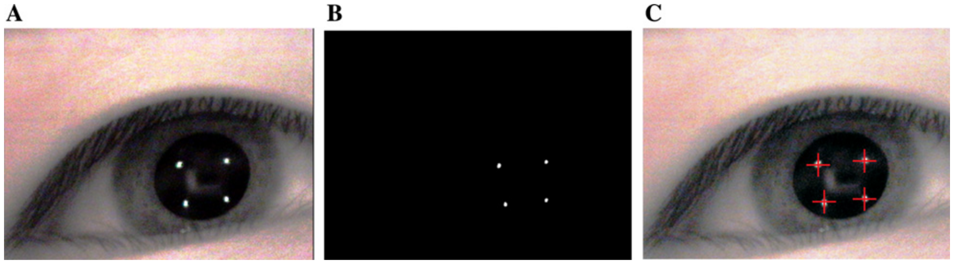


Fig. 15. Detection of the corneal reflections: (A) dark-eye image, (B) segmented result, and (C) detected glints.

levels of illumination. Furthermore, one glint may be divided into more than two blobs due to an unsuitable threshold. To overcome these problems, the binary image must be processed several times by binary morphology like erosion and dilation.

*Step 3: Group each region.*

Region grouping should be done using a sequential connected components algorithm to compute each area. From all areas calculated, the four largest regions are selected and all others are neglected.

*Step 4: Compute the center position of each region.*

Once the four regions have been detected via the previous steps, the center of each region can be computed.

Fig. 15C shows the detected glints which are marked by red crosses.

### *6.2. Detection of pupil center*

Successful pupil detection is more difficult to achieve than glint detection because the intensity of the pupil is similar to the adjacent region. To overcome the difficulty, we referred to the robust pupil detection method suggested by Ebisawa [20]. In his method, when an LED is located in the optical axis of a camera, the pupil is shown as a bright region in an image from the camera, as in Fig. 3B. Conversely, when the LED is located away from the axis, the pupil becomes darker, as in Fig. 3A.

To detect the pupil robustly, we use two images: one is taken when the LEDs of the monitor are turned on and the LED of the camera is turned off, and the other is taken when the LEDs of the monitor are turned off and the LED of the camera is turned on. If these images are obtained from the camera within a short interval, then the intensity difference within the pupil region is large while that of the regions outside of the pupil is quite small in the two images. Therefore, the difference image of the two images has high-intensity values in the pupil region, and the pupil region can be extracted by a segmentation method that is similar to glint detection. From this binary image, the center of the region can be computed easily.

The simple segmentation method may fail to determine the exact pupil region because the segmented result is very sensitive to the threshold value. Also, there are some holes made by corneal reflections in the difference image. Fig. 16 shows three examples resulting from segmentation. Although the first image indicates results from a proper threshold, there are still four punctures in the pupil region. When the difference image is segmented by a high-threshold, as in the second image, many additional holes in the pupil region result. Because of these holes, the mass center of the segmented region is slightly different from the actual pupil center. In the third image, the boundary of the pupil region is unclear and there are small blobs due to an improper threshold and noise. Because of noise, the computation of mass center cannot detect the exact center of the pupil. To reduce the effect of noise, we utilize the fact that the pupil boundary looks like an ellipse. Ellipse fitting can lessen the effect of noise, and improve the detection accuracy of the system. An improper threshold may induce extra blobs outside of the pupil region. Moreover, if there are many blobs, the segmentation method, just like the corneal detection (i.e., labeling and mass center computation), takes a long time. Therefore, a method using labeling and mass center computation is not suitable to real-time application. To resolve this problem, the boundary of the pupil region is determined, and then the boundary is fitted by an ellipse. For ellipse fitting, the direct ellipse fitting algorithm [21] is utilized.

Boundary detection must be computed for ellipse fitting. However, previous edge detection methods with convolution operator such as Sobel and Canny are

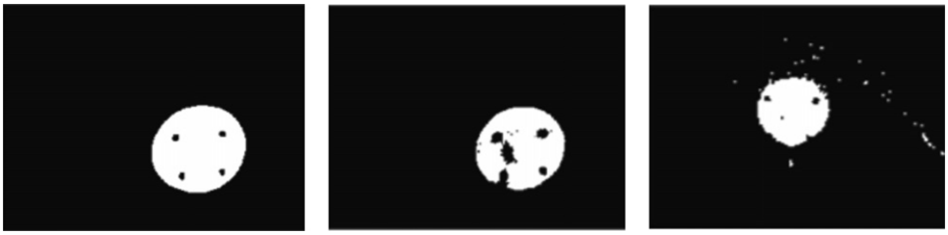


Fig. 16. Segmented results of three cases.

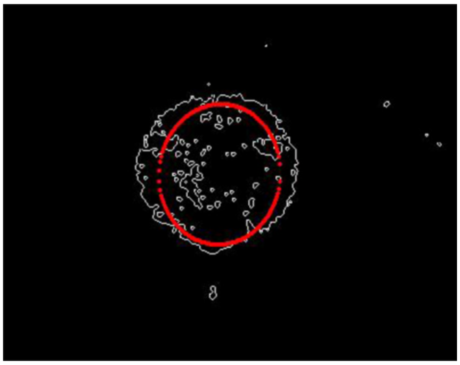


Fig. 17. Ellipse fitting to the edge detected by a Sobel operator. The red ellipse is the fitted result.

unsuitable for extracting the pupil boundary. The reason for this is shown in Fig. 17, in which the edge image is detected by a Sobel operator. The inner edges are detected as well as the boundary of the pupil region, because there are many holes in the pupil region. It is difficult to detect the boundary of the pupil region using only the edge operator. Therefore, the fitted ellipse is not accurate.

The boundary detection method of this research is similar to active contour methods like active shape model (ASM), but it is ellipse-specific in order to lessen the computational load. The pupil center is detected in two stages. A rough position is computed in the first stage, and then the precise position is determined in the second stage. The algorithm utilizes the constraint that the pupil is very close to the circle; in other words, the long axis and the short axis of the ellipse have similar lengths.

*Step 1: Obtain the difference image of the bright-eye image and the dark-eye image.*

*Step 2: Segment the difference image by a threshold.*

The resulting image is a binary image in which each pixel has 1-value if the intensity of the corresponding pixel of the difference image is more than the threshold; otherwise, the pixel has 0-value.

*Step 3: Compute the rough center and size of the pupil.*

The binary image is resized to lower scale in order to reduce computational time. The rough center $(\bar{x}, \bar{y})$ of the pupil region is determined by averaging the position $(x_i, y_i)(i = 1, \ldots, n)$ of the 1-value pixels in the down-scaled image. The rough size of the pupil region is equal to the number of the 1-value pixels.

*Step 4: Place the initial contour.*

The initial contour is a circle that consists of a number of points. The points are placed at the same degree between two adjacent points. The contour's center is located at the rough center of the pupil region, and the size of the contour is initialized by the rough size computed in the previous step.

*Step 5: Determine the boundary of the pupil region.*

The edge is established by a 1D line search along the normal vector at each point of the contour. The search lines are along the normal vectors to the initial circle in points, and a simple template such as [0 0 0 1 1 1] is used for edge detection.

*Step 6: Compute the position of the pupil.*

Once the boundary of the pupil is detected, it is fitted by an ellipse to reduce the effects of noise. We apply the direct ellipse algorithm [21].

The 1D edge search is depicted in Fig. 18. The contour consists of a number of points that are placed at the same degree interval $\theta$. On each point, the edge is detected along the search line that crosses the point and is normal to the initial contour. To reduce computation time, the length of the search line should be reduced. However, if the length is too short, the edge detection may fail. To resolve this problem, the edge location of the previously detected point is utilized to determine the initial position of the next point. The distance of the initial point from the contour's center is determined by that of the previous detected point. In Fig. 18, small triangles mark the initial positions of contour points and a cross denotes the detected edge position after a 1D search.

Fig. 19 shows the results of the detected boundary from the three examples of Fig. 16. In this figure, the small circle represents the initial position of a simplified active contour, and the cross-marks outline the boundary of the pupil region. The exact

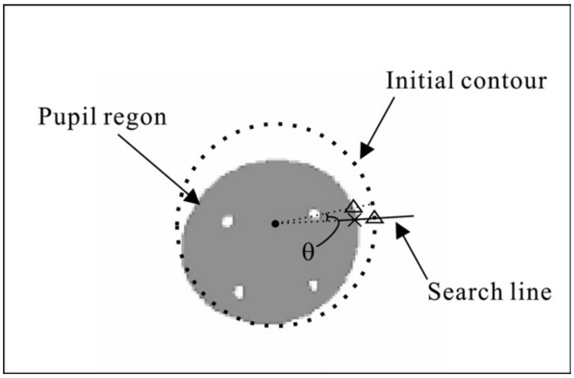


Fig. 18. Edge detection of a pupil region by a 1D search.

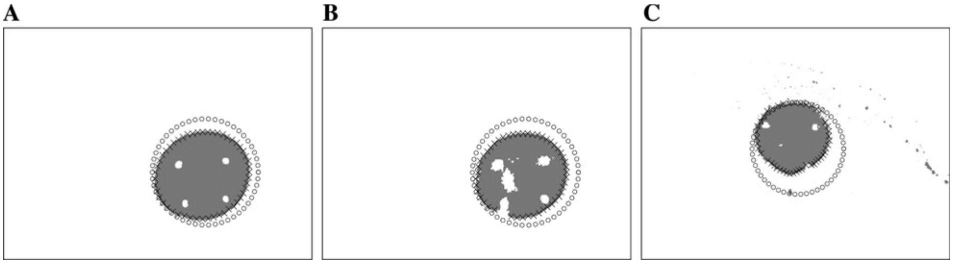


Fig. 19. Boundary detection results of the examples from Fig. 16. The small circle is an initial position of a simplified active contour. The crosses mark the boundary of the pupil region.

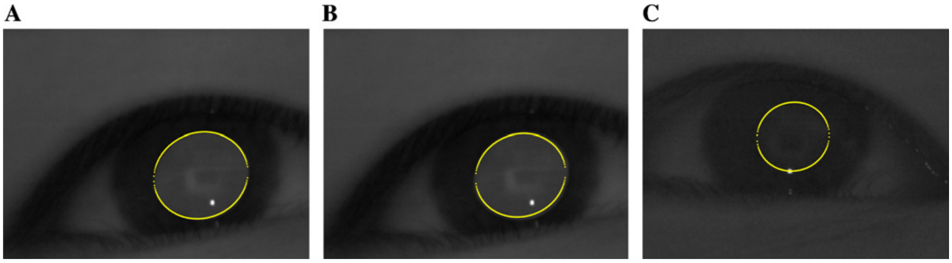


Fig. 20. Fitting results of Fig. 16.

fitting results of Fig. 16 are revealed in Fig. 20 in which the yellow dots signify the contour of the pupil region.

## 7. Face tracking subsystem

A face tracking subsystem is required to track an eye robustly. A face is tracked in images captured by a wide-view camera mounted on a pan-tilt unit, and the pan-tilt unit controls the direction of the camera according to the position of the face. Using this subsystem, the high-zoom camera can capture the eye robustly.

The face tracking algorithm is based on the Gaussian modeling of face color and mean shift algorithm [22,23]. In the first stage, face color is modeled by joint Gaussian distribution because it is known that face color is modeled well by this model [24]. We utilize normalized-RGB space to diminish the effect of illumination.

Face color distribution can be based on a Gaussian model $G(m, \Sigma^2)$, where $m = (\bar{r}, \bar{g})$ with

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i, \tag{21}$$

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i, \tag{22}$$

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix} = \begin{bmatrix} \sigma_r^2 & \rho\sigma_r\sigma_g \\ \rho\sigma_g\sigma_r & \sigma_g^2 \end{bmatrix}, \tag{23}$$

$$\rho = \frac{\sigma_{rg}}{\sigma_r\sigma_g}, \tag{24}$$

where $r_{\mathrm{i}}$ and $g_{\mathrm{i}}$ are normalized-R and -G values, respectively, and $N$ is the number of pixels intended for face color. $\Sigma$ is the covariance matrix of $r_{\mathrm{i}}$ and $g_{\mathrm{i}}$ in a facial region. We can compute the probability of a pixel color at $(x, y)$ by the following equation:

$$I(x,y) = \frac{1}{2\pi\sigma_r\sigma_g\sqrt{1-\rho^2}} \times \exp\left[-\frac{1}{2(1-\rho^2)}\left[\left(\frac{r-\mu_r}{\sigma_r}\right)^2 - 2\rho\left(\frac{r-\mu_r}{\sigma_r}\right)\left(\frac{g-\mu_g}{\sigma_g}\right) + \left(\frac{g-\mu_g}{\sigma_g}\right)^2\right]\right] \tag{25}$$

where $r$ is the normalized-R value and $g$ is the normalized-G value at $(x,y)$.

Then, we can obtain the center of a facial region by the mean shift algorithm:

$$M_{00} = \sum_x \sum_y I(x,y), \tag{26}$$

$$M_{10} = \sum_x \sum_y xI(x,y), \; M_{01} = \sum_x \sum_y yI(x,y). \tag{27}$$

The mean search window is located at

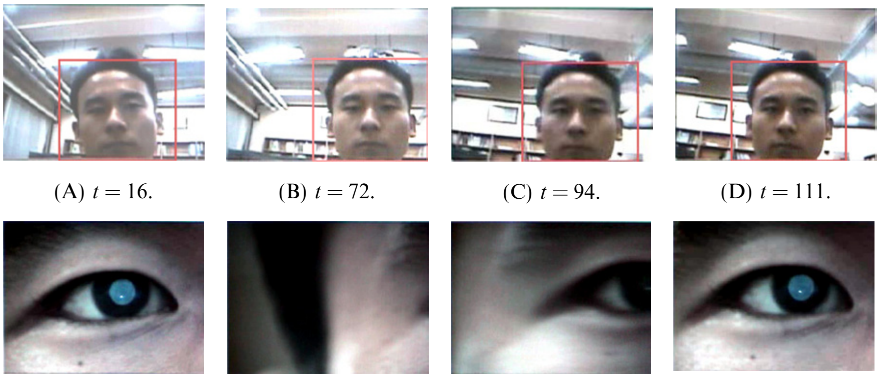
$$x_c = \frac{M_{10}}{M_{00}}, \; y_c = \frac{M_{01}}{M_{00}}, \tag{28}$$

where $I(x,y)$ is the probability value at position $(x,y)$ in the image, and $x$ and $y$ range over the search window.

In the image, the face is detected and then the position of the eye is computed relative to the center of the detected facial region. Fig. 21 shows the tracking results. The user moves his head from right to left. At time $t = 72$, the head movement causes the eye to be out of the image. At $t = 94$, the eye reappears in the image.

## 8. Experimental results

Various experiments were carried out to show the feasibility of the proposed method. Fig. 22 portrays the experimental setup. The camera was a near infrared CCD camera and the image size was 640 × 480 pixels. The computer used was a Pentium IV-1.4GHz system and the monitor had a 15-in. screen. The resolution of the screen was 1024 × 768 pixels. There are 5 × 5 target points in the screen of the monitor. The user sat 40–60 cm away from the screen and looked at one of the target points. This system estimated which point the user looked at. About 200 trials involving significant head motion were carried out, the results of which are shown in Fig. 23. The cross-indicates the estimated result, and the number indicates the target number. To compare the proposed method to Hutchinson's method, which is the conventional method, the estimated result of the conventional method is also shown in the figure. In the conventional method, the gaze point is estimated by the glint of one IR LED and the position of the pupil. In Fig. 23, $x$- and $y$-axes are coordinates of the screen, and dots remark the estimated eye gaze points. In Tables 1 and 2, the average, standard deviation, and maximum of the estimation errors are listed. The



(A) $t = 16$. (B) $t = 72$. (C) $t = 94$. (D) $t = 111$.

(E) $t = 16$. (F) $t = 72$. (G) $t = 94$. (H) $t = 111$.

Fig. 21. Face and eye tracking results: (A–D) are the images obtained by a wide-view camera, and (E–H) are the images simultaneously obtained by a high-zoom camera.

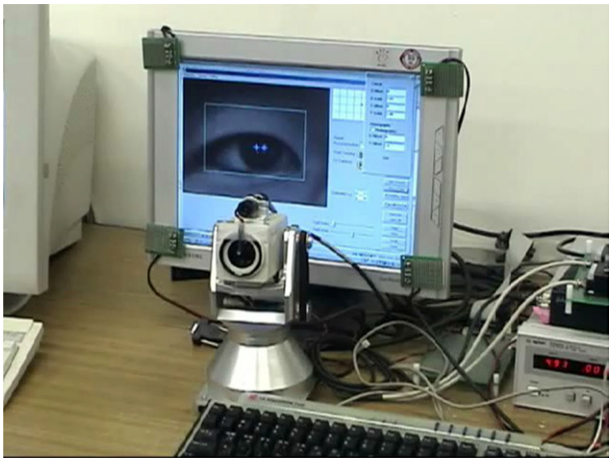
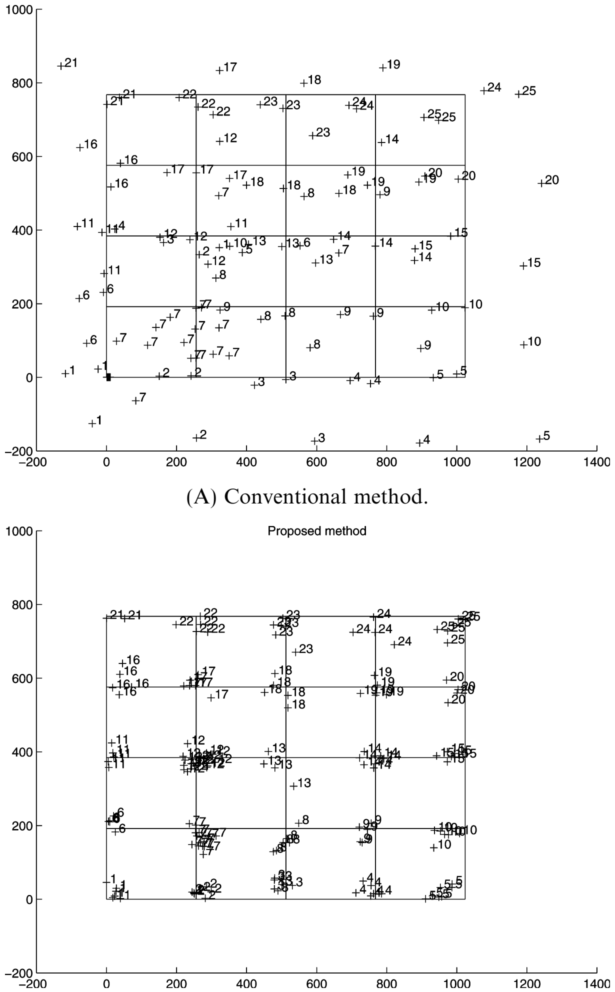


Fig. 22. Experimental setup: a monitor with four IR LEDs and two cameras mounted on a pan-tilt unit. A small wide-view camera is attached to the top of the high-zoom camera.

average error of the proposed method is 0.98 and 0.82° in $x$- and $y$-coordinates, respectively.

There are some notable causes of error. One is feature detection. The accuracy of this system depends on the accuracy of feature detection. However, detecting the pupil's center is particularly difficult, as the pupil's intensity is similar to adjacent regions. In addition, feature detection is also affected by environmental conditions such as illumination. The inexact feature detection results in estimation error. Second, the polygon of the glints is still quite small. It is about 45 pixels by 35 pixels. Because the distance between two glints can be short according to the head position, the small region of the glints may cause the accuracy to be low. One pixel in an image



(A) Conventional method.

(B) Proposed method.

Fig. 23. Experimental result. The resolution of the screen was 1024 × 768 and the screen had 5 × 5 target points.

Table 1
The average, standard deviation, and maximum of the estimation error using the conventional method

| Error | Average | Standard deviation | Maximum |
|---|---|---|---|
| X (pixel) | 77.4 | 71.5 | 407.0 |
| Y (pixel) | 70.3 | 76.2 | 304.0 |
| X (mm) | 23.1 | 21.3 | 121.2 |
| Y (mm) | 20.9 | 22.7 | 90.5 |

Table 2
The average, standard deviation, and maximum of the estimation error using the proposed method

| Error | Average | Standard deviation | Maximum |
|---|---|---|---|
| X (pixel) | 28.8 | 21.9 | 113.0 |
| Y (pixel) | 24.0 | 18.6 | 97.8 |
| X (mm) | 8.6 | 6.5 | 33.7 |
| Y (mm) | 7.2 | 5.9 | 29.3 |

corresponds to several tens of pixels in the screen. Certain situations cause the glints to be too close to each other. One such instance is when a user is too far from the monitor. In this situation, the polygon made by the glints may be too small. However, this situation is not normal because the distance between the user and the monitor is usually from 40 to 60 cm, as defined in Section 3. If the user is far from the monitor and the region made by the glints is too small, then we have to use a lens with higher magnifying power. The second instance is when a user is at the side of the monitor. However, this case is also anomalous because the user cannot see the screen well. Another source of error is the approximation of the virtual projection point. To utilize the property of projection space, virtual projection points are computed. By the virtual projection point and the cross-ratio, the estimated result is better than that of the conventional method when head movement is considered. However, in order to simplify the computation of the virtual projection point, an approximation for the reflection of light was used, but this approximation causes estimation errors. The error is particularly large when the user looks at the corner of the screen.

The real-time issue is very important to eye gaze estimation systems. This system runs in 15 frames/s because the feature detection takes a long time. To reduce the search space, the region of the pupil is identified first, and then the glints are detected in the region of the pupil. This way, the computation time is decreased drastically, and feature detection is not affected by exterior noise.

## 9. Conclusions

In this paper, a novel method of estimating the eye gaze point of a human eye was suggested. Four IR LEDs were attached to a computer monitor, and light from these LEDs was reflected on the surface of the cornea. This reflection generated glints in images captured by a camera. To estimate the eye gaze point, the pupil's center position and the glints of the corneal reflection were detected in images and a cross-ratio was used. To use the cross ratio, virtual projection points were suggested. The advantage of the proposed method is that it does not require any geometrical knowledge concerning LEDs, the monitor, the camera, or the eye. Also, it does not require a complicated calibration process. This method works well under large head movement in real-time, and consists of relatively simple devices.

The pupil's center position is difficult to detect. Therefore, an ellipse-specific active contour was proposed. The experimental results show that our method is able to track human eye gaze simply and quickly in real-time.

Expanding upon our research, we are now working on improving the accuracy of the proposed method. In the computation of the virtual projection point, the approximation is rough. To increase the performance, the approximation must be improved. Also, we have proposed a robust pupil detection method, but it is still difficult to detect the pupil exactly. Improving this aspect of the system is important because once the feature detection is fast and accurate, the face tracking system will no longer be required. In this paper, our proposed method requires five light sources and two cameras, but if the systems can be improved so that feature detection and gaze estimation are possible with only one camera, the system will be lighter and cheaper. Therefore, we are now considering a better feature detection method.

## References

[1] Y. Kuno, T. Yagi, Y. Uchikawa, Development of fish-eye VR system with human visual function and biological signal, in: IEEE Internat. Conf. on Multi-sensor Fusion and Integration for Intelligent Systems, 1996, pp. 389–394.

[2] Y. Kuno, T. Yagi, Y. Uchikawa, Development of Eye-gaze input interface, in: Proc. of 7th Internat. Conf. on Human–Computer Interaction jointly with 13th Symposium on Human Inferface, 1997, p. 44.

[3] D.H. Kim, J.H. Kim, D.H. Yoo, Y.J. Lee, M.J. Chung, A human–robot interface using eye-gaze tracking system for people with motor disabilities, Trans. Control Automat. Syst. Eng. 3 (4) (2001) 229–235.

[4] T.E. Hutchinson, Human–computer interaction using eye-gaze input, IEEE Trans. Syst. Man Cybernet. 19 (6) (1989) 1527–1533.

[5] C.H. Morimoto et al., Keeping an eye for HCI, in: Proc. on Computer Graphics and Image Processing, 1999, pp. 171–176.

[6] Q. Ji, Z. Zhu, Eye and tracking for interactive graphic display, in: Internat. Symp. on Smart Graphics, 2002.

[7] A. Sugioka, Y. Ebisawa, M. Ohtani, Noncontant video-based eye-gaze detection method allowing large head displacements, in: IEEE Internat. Conf. on Medicine and Biology Society, 1996, pp. 526–528.

[8] T. Ohno, N. Mukawa, A. Yoshikawa, FreeGaze: a gaze tracking system for everyday gaze interaction, in: Proc. of Eye Tracking Research and Applications Symposium, 2002, pp. 125–132.

[9] T. Ohno, N. Mukawa, A. Yoshikawa, Just blink your eyes: a head-free gaze tracking system, in: *CHI2003*, 2003, pp. 950–951.

[10] J. Liu, Determination of the point of fixation in a head-fixed coordinate system, Internat. Conf. Pattern Recogn. 1 (1998) 501–504.

[11] S. Pastoor, J. Liu, S. Renault, An experiment multimedia system allowing 3-D visualization and eye-controlled interaction without user-worn devices, IEEE Trans. Multimedia 1 (1) (1999) 41–52.

[12] Y. Matsumoto, A. Zelinsky, Real-time face tracking system for human–robot interaction, in: IEEE Internat. Conf. on System, Man, and Cybernetics, 1999, pp. 830–835.

[13] R. Newman, Y. Matsumoto, S. Rougeaux, A. Zelinsky, Real-time stereo tracking for head pose and gaze estimation, in: Fourth IEEE Internat. Conf. on Automatic Face and Gesture Recognition, 2000, pp. 122–128.

[14] J. Wang, E. Sung, Study on eye gaze estimation, IEEE Trans. Syst. Man Cybernet.-Part B: Cybernetics 32 (3) (2002) 332–350.

[15] S.E. Palmer, Vision Science, MIT Press, Cambridge, MA, 1999.

[16] S. Pheasant, Bodyspace, Taylor & Francis, London, 1986.

[17] M.S. Sanders, E.J. McCormick, Human Factors in Engineering and Design, McGraw-Hill, New York, 1993.

[18] K. Kroemer, H. Kroemer, Ergonomics: How to Design for Ease and Efficiency, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[19] W. Karwowski, W.S. Jarras, The Occupational Ergonomics Handbook, CRC Press, Boca Raton, FL, 1999.
[20] Y. Ebisawa, Improved video-based eye-gaze detection method, IEEE Trans. Instrum. Meas. 47 (4) (1998) 948–955.
[21] A. Flizgibbon, M. Pilu, R.B. Fishe, Direct least square fitting of ellipses, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 476–480.
[22] K. Fukunaga, Introduction to Statistial Pattern Recognition, Academic Press, New York, 1990.
[23] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 790–799.
[24] J. Yang, A. Waibel, A real-time face tracker, in: *Proc. 3rd IEEE Workshop Application Computer Vision*, vol. 17, 1996, pp. 142–147.