ELSEVIER

Optics and Lasers in Engineering 42 (2004) 91–108

OPTICS and LASERS in ENGINEERING

Design of a computer game using an eye-tracking device for eye's activity rehabilitation

Chern-Sheng Lin[a,*], Chia-Chin Huan[a], Chao-Ning Chan[a], Mau-Shiun Yeh[b], Chuang-Chien Chiu[a]

[a] *Department of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan, ROC*
[b] *Chung-Shan Institute of Science and Technology, Lung-Tan, Tao-Yuan, Taiwan*

Received 1 October 2002; received in revised form 1 June 2003; accepted 1 June 2003

Abstract

An eye mouse interface that can be used to operate a computer using the movement of the eyes is described. We developed this eye-tracking system for eye motion disability rehabilitation. When the user watches the screen of a computer, a charge-coupled device will catch images of the user's eye and transmit it to the computer. A program, based on a new cross-line tracking and stabilizing algorithm, will locate the center point of the pupil in the images. The calibration factors and energy factors are designed for coordinate mapping and blink functions. After the system transfers the coordinates of pupil center in the images to the display coordinate, it will determine the point at which the user gazed on the display, then transfer that location to the game subroutine program. We used this eye-tracking system as a joystick to play a game with an application program in a multimedia environment. The experimental results verify the feasibility and validity of this eye-game system and the rehabilitation effects for the user's visual movement.

Keywords: Eye mouse; Eye-tracking system; Rehabilitation; Eye's activity; Cross-line tracking algorithms

1. Introduction

Eye-tracking devices were initiating in military applications for guiding weapons systems while freeing the pilot's hands for guiding the aircraft through combat aerial maneuvers. Usually, these devices include a detection device to detect the bio-electromagnetic signals or image sequences generated by eye movements. There have

*Corresponding author.

0143-8166/$ - see front matter
doi:10.1016/S0143-8166(03)00075-7

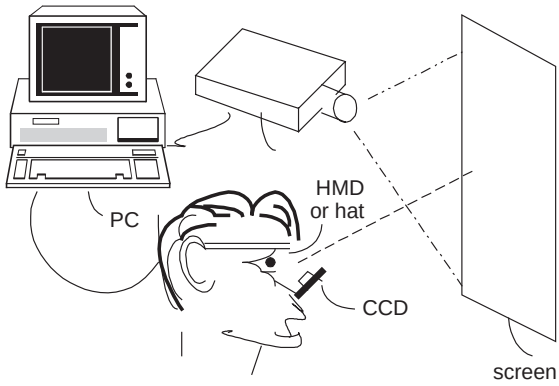


Fig. 1. Computer game setup using the eye-tracking method.

been various eye-tracking systems developed over the past 30 years. The image-type eye-tracking system that we adopted in this study consists of a charge-coupled device (CCD) camera, an image capture card and an LCD projector, as shown in Fig. 1.

The eye-tracking device has become one of the most important human–machine interfaces in which eye movements are related to the information-processing demands of a task. For example, in the eye-tracking experiments by Lin et al. [1], a new search method, superior to the existing methods, was used. The eye opening and closing actions activate additional commands, used for controlling robots. In order to save much computer search time on the image of the face, this system uses many search checkers to search the binary shadows. When coming across extraneous features, it can quickly skip over them without performing a detailed template comparison. This method is called diagonal-box checker search. This system allows the handicapped an opportunity to perform some simple tasks without the help of others. Wagner and Galiana [2] adopted template matching with correlation calculation to find the locations of eyes. Spindler and Chaumette [3] used an off-line calibration and gaze coordinate filtering for a visual servo task. Grattan and Palmer [4,5] developed a microcomputer-based system for the disabled that relies upon differential reflection from the eyeball. The input blink is used to activate the data presentation as a binary tree or matrix scan.

When the head movements are much slower than the iris movements, Xie et al. [6] presented a method for compensating for the head movements using cascaded tracking of the white and dark regions in the eye's image. Martin and Schovance [7] suggested a very complicated eye movement control model with muscle mass and tendon tension parameters. Allison et al. [8] used an integrated head and eye-tracking system to investigate compensatory vestibulo-ocular responses. The measurement of three-dimensional (3-D) viewing behavior requires a relatively large computational effort. Talmi and Liu [9] used highlight detection and displacement vector transformation in the determination of the 3-D eye position. In 3-D multimedia system visualization, eye blinking can be detected by calculating the values of the squared frame differences (SFD) of the facial image blocks [10]. Human

visual tracking and evaluation systems comprise a field of significant interest and importance.

Here a new image-processing method, based on a new cross-line tracking and stabilizing algorithm, is proposed for calculating the location and diameter of the pupil using inexpensive optical hardware. The CCD camera used in this system was built from an observation camera component set. On the back of the camera, a mini sender is mounted which transmits the video signal to a frame grabber at the computer. The user must set the camera behind the user's face and focus the camera onto one of the user's eyes. The system can judge the gazing direction from the information of the gray level of the eye with special image-processing algorithms. However, this method need not require the user's head to remain totally motionless [11], because the user's eye will not go out of focus when the user's head move.

We will discuss the construction of the system, the image-processing approach and the blink and eye close functions. The computer automatically processes the eye-image captured by the CCD and acquires the users eye activity. This system is able to digitally record where and what someone is looking at. We used this eye-tracking system as a 'visual' joystick to play a game using a multimedia application program.

2. Cross-line tracking and stabilizing algorithm

The practical application of this eye-tracking device is discussed next. First, a special CCD driven by an image capture card captures the eyeball image. The eyeball image is then transmitted to a PC for pre-image processing. The pre-image processing includes two steps: transferring a RGB image color formula into a gray scale image represented by 8 bits. A threshold value is set and the gray scales of all pixels are divided into two groups. The gray scale of one group, which is over the threshold value, is set to 255. The other group, which is less than the threshold value, is set to 0. Therefore, the transferred image is a black and white image and only shows a pupil and eyelashes. Subsequently, the brightness of the image in the x and y directions is evaluated. According to the evaluation of the brightness in the x and y directions, the position of the centroid of the pupil is obtained. Thus, the total number of black pixels, which represents the area of the pupil in the image, can be calculated to identify the eye blink conditions. However, this method is time consuming and cannot avoid the noise problem. Furthermore, many advanced template matching algorithms and feature extraction methods have been developed for locating the positions of eyes [12–15].

A cross-line tracking method is used to process the image of the eye especially for the user who has problems in ocular motility dysfunction or eye movement disorders. In this new method we can only process one row and one column of an image instead the entire image matrix and find the gazing point when user's eye sway. The first step is to find the position of the center point of the pupil. Suppose that the pattern of a pupil is black and circular, then any line that is perpendicular and passes through the midpoint of the subtense of circle will also pass through the center of circle. Before the cross-line thresholding process can be performed, the

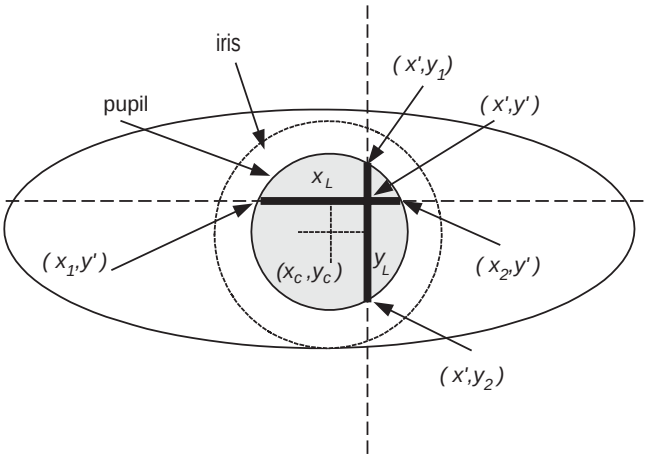


Fig. 2. Cross-line eye-tracking method.

pupil pixel contrast must be enhanced using a weighting operation. Assume that D is the distance from the center point of the cross-line to a pixel (x_i, y_i):

$$f(x_i, y_i)' = D\alpha + f(x_i, y_i), \tag{1}$$

where $f(x_i, y_i)$ is the gray level of a pixel at the location (x_i, y_i). $f(x_i, y_i)'$ is the gray level value after the weighting operation. α is a weighting factor which is adjustable. Accordingly, the pixel in the pupil will obtain the value $f(x_i, y_i)'$, smaller than a specified data.

As shown in Fig. 2, assume that the center of cross-line is (x', y') and the end-points of the horizontal and vertical segments are $(x_1, y'), (x_2, y'), (x', y_1), (x', y_2)$. The position of the center point of the pupil (x_c, y_c) can then be expressed as

$$x_c = (x_1 + x_2)/2,$$

$$y_c = (y_1 + y_2)/2.$$

In the pupil, the length of the horizontal and vertical black segment x_L, y_L can be expressed as

$$x_L = (x_2 - x_1),$$

$$y_L = (y_2 - y_1).$$

If the user closes his eye, then we may find there is no horizontal or vertical black segment. So we can let

$$x_c = x'_c,$$

$$y_c = y'_c,$$

where (x'_c, y'_c) is the center point of the pupil in the last state.

In this system, the image-processing rate is 12–15 frames/s and the maximum eye movement velocity is 70 pixels/s. In addition, the length of the horizontal and vertical black segment x_L, y_L always remains in the range of 40–80 pixels. If the new cross-line center is set at (x', y') this will be the present center point of the pupil (x_c, y_c). The new cross-line is guaranteed to intersect the center of the pupil. The eye-tracking task can then continue.

The above method is very fast, but there are two defects:

(a) Noise yields because we only use the information from the cross-line of the entire pupil.
(b) The user's eye will sway while the user is watching a target.

So we must add a stabilizing algorithm in the tracking method. If we select the center of the cross-line as the cursor position, the cursor will sway too fast and be unsuitable for use. In order to solve the cursor sway problems, the following steps are introduced to increase the cursor stability.

Let the moving vector of the cursor be

$$\vec{v}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \tag{2}$$

where $x(t)$ and $y(t)$ are the variations in the coordinates of the pupil center at $t-1$ and t.

From the following equation, we can judge if the eyeball movement directions are coincident. In other words, if the direction of movement is reliable, the moving vector must be in accord with Eq. (3).

$$\frac{\vec{v}(t)}{||\vec{v}(t)||} = \frac{\vec{v}(t-n)}{||\vec{v}(t-n)||}. \tag{3}$$

If the sampling data can satisfy the above equation many times, we can infer that the gazing direction movements, n, are the number of coincident movement directions. The larger the value of n, the more reliable the eyeball movement direction. Let $\vec{v}_o(t)$ be the output vector:

$$\vec{v}_o(t) = \begin{cases} \dfrac{\vec{v}(t) - \vec{v}(t-1)}{m(k-n)} & \text{if } n \leqslant k, \\ \vec{v}(t) & \text{if } n > k. \end{cases} \tag{4}$$

The value of k and m can be determined by the user, and the recommended values are $m = 1$ and $k = 3$.

The above equation neglects eye wavering when someone is gazing at a target. A tolerance value is therefore added to Eq. (3) to account for the wavering condition. The stabilizing algorithm can be expressed as

$$\frac{\vec{v}(t-n)}{||\vec{v}(t-n)||} - p \leqslant \frac{\vec{v}(t)}{||\vec{v}(t)||} \leqslant \frac{\vec{v}(t-n)}{||\vec{v}(t-n)||} + p, \tag{5}$$

where p is the tolerance value.

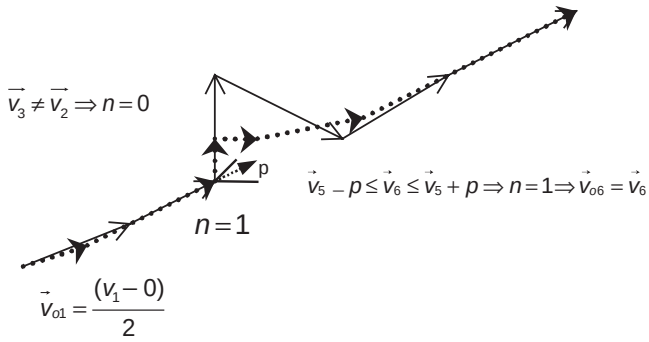


Fig. 3. One example of the stabilizing algorithm.

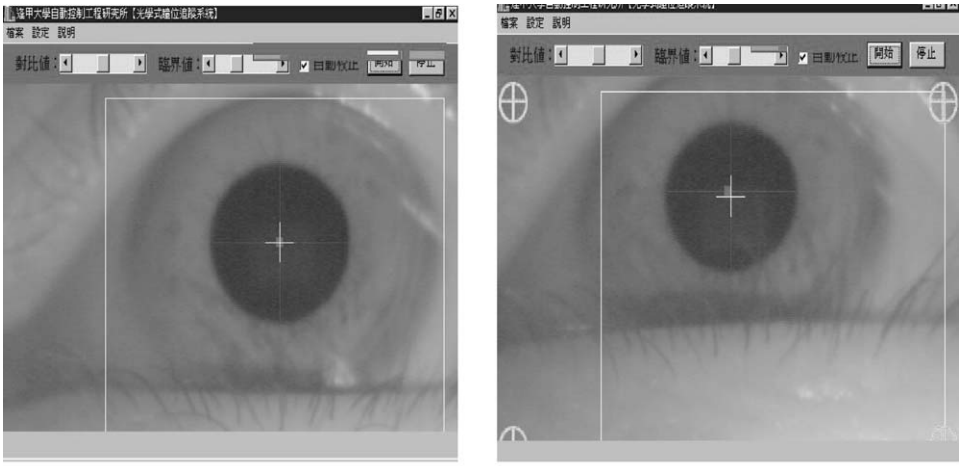


Fig. 4. Center of the pupil (short cross-hairs) and output gazing point (long cross-hairs): (A) superposition; (B) separately.

One example of the stabilizing algorithm is shown in Fig. 3. In Fig. 3 $m = 1$ and $k = 2$, the solid line is the ith velocity $\vec{v}_i$ of the center of the pupil. The dashed line is the ith velocity of the gazing point $\vec{v}_{oi}$, as output according to the algorithm. $\vec{v}_{o1} = (v_1 - 0)/2$ represents the first output, here the summation of coincide times $n = 0$. The 2nd direction is different from the 3rd direction, so n still is 0. That is $\vec{v}_3 \neq \vec{v}_2 \Rightarrow n = 0$. The 5th direction is the same with the 6th moving direction and velocity deviation is smaller than tolerance value p, so $n = 1$:

$$\vec{v}_5 - p \leqslant \vec{v}_6 \leqslant \vec{v}_5 + p \Rightarrow n = 1 \Rightarrow \vec{v}_{o6} = \vec{v}_6.$$

If the direction coincides and lasts for specified intervals, the output gazing point will align with the center of pupil very fast (Fig. 4(A)). However, if the direction is inconsistent (or the direction of v_3 is larger than the tolerance value p), the output

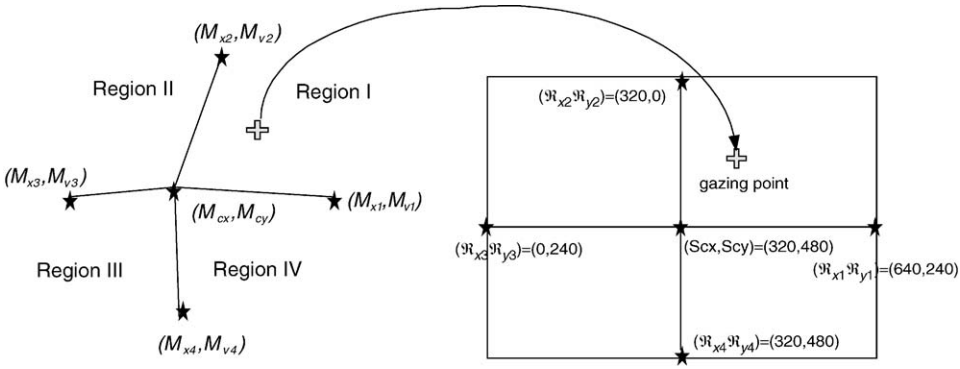


Fig. 5. Relationship of five calibration points of eye image, screen center and four margin points.

gazing point will not follow the center of pupil (Fig. 4(B)). This method avoids cursor swaying caused by signal noise or an inexperienced user.

3. Coordinates mapping of gazing point

System calibration is very important for transforming the pupil center coordinates in the image into gazing point coordinates in the screen [16,17]. In this system the CCD is mounted in the user's head, so the coordinates mapping is not very sensitive to head movement. When calibration begins, the user is asked to focus his gaze upon the center of the target, which will appear on the screen. As shown in Fig. 5, the screen center and four margin points (S_{cx}, S_{cy}), $(\Re_{x1}, \Re_{y1})$, $(\Re_{x2}, \Re_{y2})$, $(\Re_{x3}, \Re_{y3})$, $(\Re_{x4}, \Re_{y4})$ were adopted for calibration. Beginning at center, then from the top to bottom, left to right, every two seconds the target will shift to the next position. In a standard calibration process, a user shifts his gaze to the target until he has completed the calibration sequence of five gazing points. The corresponding locations for the center of the pupil (M_{cx}, M_{cy}), (M_{x1}, M_{y1}), (M_{x2}, M_{y2}), (M_{x3}, M_{y3}), (M_{x4}, M_{y4}) are obtained in the same sequence. The coordinate transformation coefficients (β_{xn}, β_{yn}) are then obtained as follows:

$$(\beta_{xn}, \beta_{yn}) = \left(\frac{\lambda x(\Re_{xn} - S_{cx})}{M_{xn} - M_{cx}} + \delta x, \frac{\lambda y(\Re_{yn} - S_{cy})}{M_{yn} - M_{cy}} + \delta y \right), \quad n = 1, 2, 3, 4, \tag{6}$$

where $\lambda x, \lambda y$ are the coordinate proportional factors and $\delta x, \delta y$ are the coordinate shift factors.

The coordinate transformation coefficients (β_{xn}, β_{yn}) also indicate the ability of the user to use the eye-game system. In our system, coordinate transformation coefficients under 6 indicates an excellent ability, 6–15 indicates a good ability, greater than 15 indicates poor ability.

The values (M_{cx}, M_{cy}), (M_{x1}, M_{y1}), (M_{x2}, M_{y2}), (M_{x3}, M_{y3}), (M_{x4}, M_{y4}) in the image indicate the margins which should not be exceeded when a user is looking at the screen. If the system detects that the margins have been exceeded, it will

recommend that the user keep his head as still as possible through an alarm. If a margin is exceeded continually, then a compensation shift is needed. For example, if (M_x, M_y) exceeds the margin (M_{xs}, M_{ys}) 12 times (or for 1 s), the compensation shift (A_x, A_y) can be expressed as follows:

$$A_x = M_x - M_{xs},$$
$$A_y = M_y - M_{ys}. \tag{7}$$

Immediately after the calibration process, the user will be able to move the aiming cursor with his eye at an update rate of 12–15 frames/s. The relationship of gazing point (S_x, S_y) corresponding to the pupil center (M_x, M_y) can be expressed as

Region I

$$S_x = S_{cx} + \beta_{x1}\left(M_x + Ax - \frac{M_y + Ay - M_{cy}}{\alpha_1}\right),$$
$$S_y = S_{cy} + \beta_{y2}\left(M_y + Ay - \frac{M_x + Ax - M_{cx}}{\alpha_2}\right),$$
$$\alpha_1 = \beta_{x2}\left(\frac{M_{y2} - M_{cy}}{\Re_{x2} - S_{cx}}\right),$$
$$\alpha_2 = \beta_{y1}\left(\frac{M_{x1} - M_{cy}}{\Re_{y2} - S_{cy}}\right).$$

Region II

$$S_x = S_{cx} + \beta_{x3}\left(M_x + Ax - \frac{M_y + Ay - M_{cy}}{\alpha_3}\right),$$
$$S_y = S_{cy} + \beta_{y2}\left(M_y + Ay - \frac{M_x + Ax - M_{cx}}{\alpha_4}\right),$$
$$\alpha_3 = \beta_{x2}\left(\frac{M_{y2} - M_{cy}}{\Re_{x3} - S_{cx}}\right),$$
$$\alpha_4 = \beta_{y3}\left(\frac{M_{x3} - M_{cy}}{\Re_{y2} - S_{cy}}\right).$$

Region III

$$S_x = S_{cx} + \beta_{x3}\left(M_x + Ax - \frac{M_y + Ay - M_{cy}}{\alpha_5}\right),$$
$$S_y = S_{cy} + \beta_{y4}\left(M_y + Ay - \frac{M_x + Ax - M_{cx}}{\alpha_6}\right),$$
$$\alpha_5 = \beta_{x4}\left(\frac{M_{y4} - M_{cy}}{\Re_{x3} - S_{cx}}\right),$$
$$\alpha_6 = \beta_{y3}\left(\frac{M_{x3} - M_{cy}}{\Re_{y4} - S_{cy}}\right).$$

Region IV

$$S_x = S_{cx} + \beta_{x1}\left(M_x + Ax - \frac{M_y + Ay - M_{cy}}{\alpha_7}\right),$$

$$S_y = S_{cy} + \beta_{y2}\left(M_y + Ay - \frac{M_x + Ax - M_{cx}}{\alpha_8}\right),$$

$$\alpha_7 = \beta_{x4}\left(\frac{M_{y4} - M_{cy}}{\Re_{x1} - S_{cx}}\right),$$

$$\alpha_8 = \beta_{y1}\left(\frac{M_{x1} - M_{cy}}{\Re_{y4} - S_{cy}}\right).$$

4. Eye blink and eye close functions

According to the variations in the vertical black segment, y_L, we can obtain the diameter of the pupil, which appears as a black circle. The pupil allows light to enter the eye through neurological input to the iris muscles, which control the pupil diameter. In a bright environment, the pupil may be constricted to an opening of less than 2 mm in diameter. In a dim environment, the pupil may widen to over 8 mm. Such a change represents about a 17-fold increase in the pupil area. The pupil changes size in response to not only ambient light levels but also other things, such as visual fatigue, and exciting conditions, etc. This means that a signal from the scene upon which the eye is focusing can also affect the size of the pupil. When someone is performing a close task, the pupil shrinks to sharpen the image. On the contrary, the pupil becomes larger when one is watching a distant object. As shown in Fig. 6, humans periodically blink to lubricate their eyes. The closing of the eyelids will yield variations in the vertical black segment, y_L. The sampling rate is 12 samples/s. The interval of a blink is about 0.2 s. The scaling factor for the pixel to pupil diameter is 0.1 mm/pixel in our system.

In the eye-game system, the user has a feature that 'fires' or 'emits dead light' with a slow blink (wink). To perform a 'firing' function, the user closes the eye being

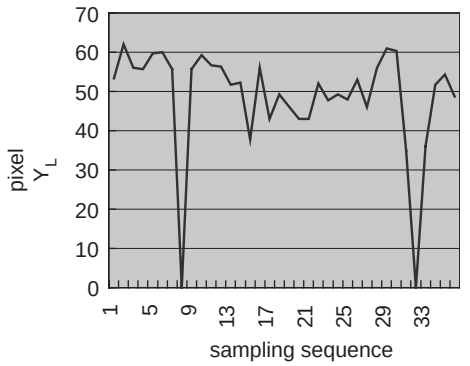


Fig. 6. Variations in the vertical black segment y_L in a periodic blink.

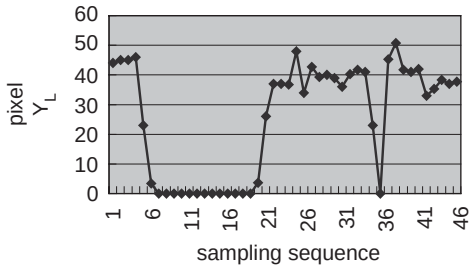


Fig. 7. Variations in the vertical black segment y_L in a slow blink for firing.

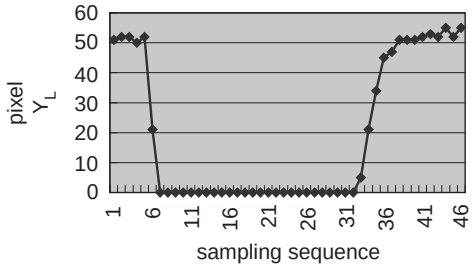


Fig. 8. Variations in the vertical black segment y_L in a slow blink for emitting dead light.

tracked for 1 s until a ring sound appears from the sound blaster card (Fig. 7). As soon as the sound appears, the user opens his eye and completes the 'firing' task. To perform a 'dead light' emission function, the user closes eye being tracked for 2 s until a bell sound appears from the sound blaster card. As soon as the sound appears, the user opens his eye and completes the 'dead light emission' task (Fig. 8).

In the eye-game design, the user must take a rest if he engages in the game too long. An energy bar is used to monitor the amount of time the user is engaged with the game. If the user closes his eye, the energy will increase. If the energy runs out, the game is over. All of these functions can be integrated using the following expression:

$$w_t = \sum_{i=1}^{n} w_{0i}$$

where when $w_t > 0$, increase the energy bar and send out music;when $w_t > w_1$, ready to 'fire' and send a ring signal;when $w_t > w_2$, ready to 'emit dead light' and send a bell signal.

w_t is the summation of the energy increase factor. It will be reset if the user opens his eye. Energy will be lost if the pixel number of the vertical black segment becomes $y_L > 20$ or the energy increase factor becomes $w_{0i} = 0$.

w_{0i} is the energy increased factor in the interval from the time the user closes and opens his eyes (Table 1).

Table 1
Relationship between pixel number of the vertical black segment and energy increase factor w_0

Pixel number of vertical black segment y_L	Energy increasing factor w_0
$y_L > 20$	0
$20 \geqslant y_L > 10$	0.2
$10 \geqslant y_L > 5$	0.3
$5 \geqslant y_L > 0$	1

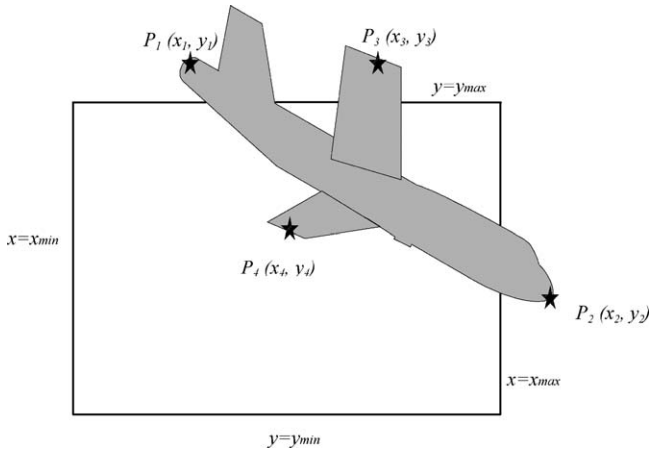


Fig. 9. Destruction window boundaries and the target skeleton.

w_1, w_2 are the threshold values, which are adjustable for controlling the length of time the user performs a 'firing' or 'dead light emitting' task. In this system, the default values for w_1 and w_2 are 12 and 24.

5. Destruction window

A rectangular area on the display device to a viewport is called the 'destruction window' in the game. If a major part of the target is in the destruction window, the target is exploded. Since the airplane is flying and rotating at all times, it is not suitable to only check if the centroid of the airplane is in the destruction window. Here the airplane is represented by a cross-line shape. We adopted a modified windowing and clipping algorithm to judge if the firing mission is successful. The specified steps for the windowing algorithm are shown as follows (Fig. 9):

(a) set the boundaries of the destruction window: $x = x_{\min}$, $x = x_{\max}$, $y = y_{\min}$, $y = y_{\max}$
set the end point of the target skeleton of the airplane $p_1(x_1, y_1)$, $p_2(x_2, y_2)$, $p_3(x_3, y_3)$, $p_4(x_4, y_4)$

initialize: $r_1 = 0$, $r_2 = 1$, $r_3 = 0$, $r_4 = 1$, (let r_1–r_2 and r_3–r_4 of the lines be in the window) $\Delta_1 = 0$, $\Delta_2 = 0$ (assume Δ_1 and Δ_2 are the percentage)

(b) loop
compute L_1, L_2, L_3, L_4
$L_1 = x_1 - x_{\min}$, $L_2 = x_2 - x_{\min}$, $L_3 = x_3 - x_{\min}$, $L_4 = x_4 - x_{\min}$ for the left boundary of the destruction window
$L_1 = x_{\max} - x_1$, $L_2 = x_{\max} - x_2$, $L_3 = x_{\max} - x_3$, $L_4 = x_{\max} - x_4$ for the right boundary of the destruction window
$L_1 = y_1 - y_{\min}$, $L_2 = y_2 - y_{\min}$, $L_3 = y_3 - y_{\min}$, $L_4 = y_4 - y_{\min}$ for the lower boundary of the destruction window
$L_1 = y_{\max} - y_1$, $L_2 = y_{\max} - y_2$, $L_3 = y_{\max} - y_3$, $L_4 = y_{\max} - y_4$ for the upper boundary of the destruction window

(c) if $L_1 < 0$ & $L_2 < 0$ & $L_3 < 0$ & $L_4 < 0$, then the target is missed

(d) if L_1, L_2 are different sign
$r_a = L_1/(L_1 - L_2)$
if L_3, L_4 are different sign
$r_b = L_3/(L_3 - L_4)$

(e) if $L_1 < 0$, $r_1 = \max(r_a, r_1)$
if $L_2 < 0$, $r_2 = \min(r_a, r_2)$
if $L_3 < 0$, $r_3 = \min(r_b, r_3)$
if $L_4 < 0$, $r_4 = \min(r_b, r_4)$

(f) return to (b)
until all boundaries are processed

(g) if $r_2 \geqslant r_1$, let $\Delta_1 = r_2 - r_1$
if $r_4 \geqslant r_3$, let $\Delta_2 = r_4 - r_3$
if $\Delta_1 + \Delta_2 >$a specified value, then destroy the target

(h) otherwise, the target is missed

6. Experiments

In the experimental setup, the CCD camera has a sensor of a low illumination intensity of 0.01 lx. This sensor can capture an image without any additional light source. In order to connect the CCD to computers or visors, an installation kit and various lenses are needed. One installation kit is an angle adjustment element and the other is a mounting fixture element. On the top or the bottom of the CCD, a connecting element, an L-shaped mounting screw to one end of which is connected to a screw nut is positioned to connect the angle adjustment element. When a user wears the eye-game device, the first thing to do is to adjust a suitable weighting factor α and a thresholding value. The cross-line will shrink if α is too high or the threshold value is too low (Fig. 10). The cross-line will dilate if α is too low or the threshold value is too high (Fig. 11). If the room lighting and display area on the screen are causing a white portion in the pupil image, the user must change the position of the screen, turn off some lighting, or adjust the angle adjustment element. In most cases, this eye-game system cannot work with the user wearing glasses. When the user

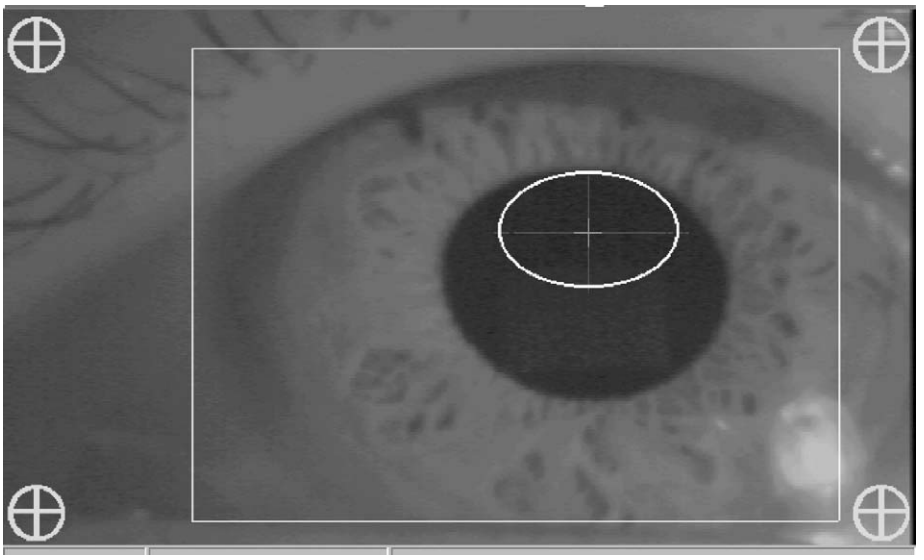


Fig. 10. Weighting factorα result is too high or the threshold value is too low.

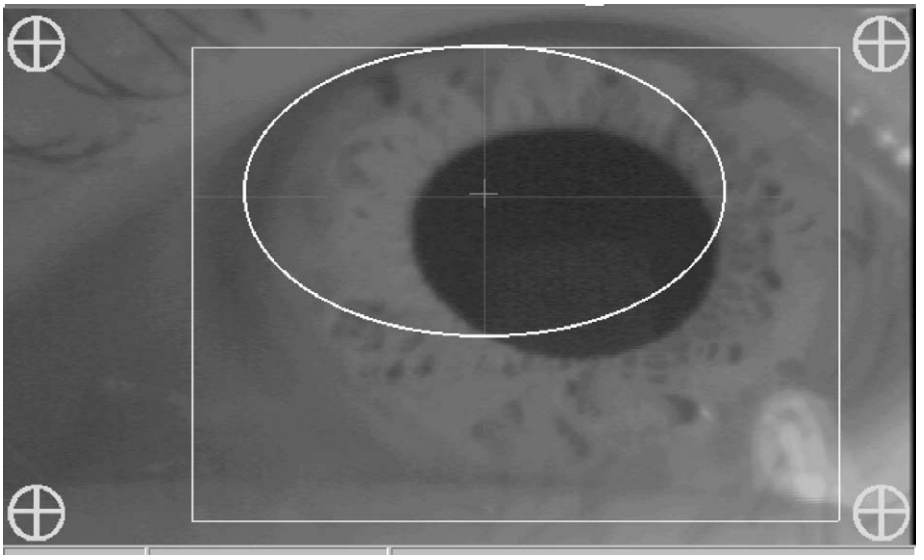


Fig. 11. Weighting factorα result is too low or the threshold value is too high.

wears glasses, the reflection from the surface of the lens may obscure the image of the eye, causing insufficient image information to make an accurate measurement of the cross-line intersection.

The eye-game system calibration process is shown in Fig. 12. The user must focus his gaze on the top, left, middle, right, bottom points in the screen. The

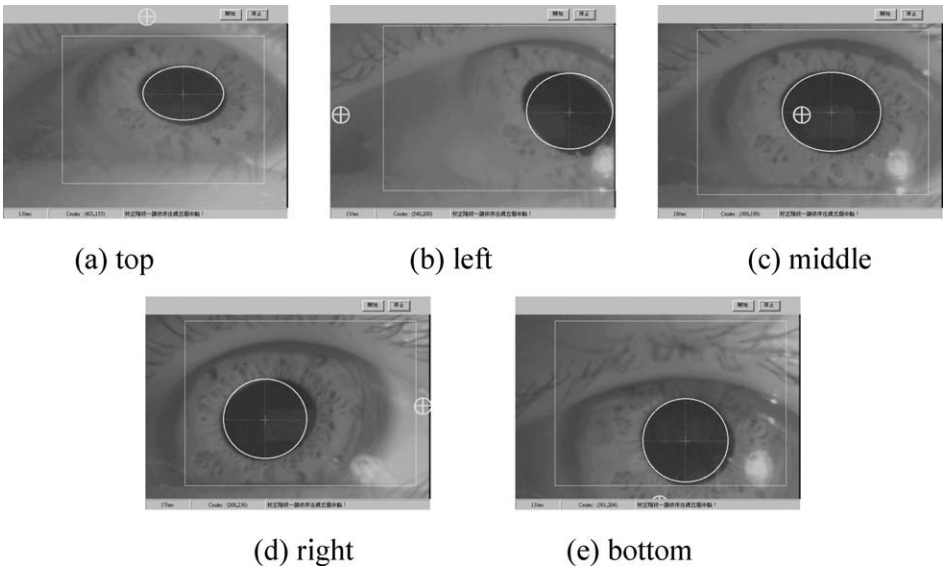


Fig. 12. Eye-game system calibration process.

user can see the movement of his eye in the screen. The system notes when the user gazes at the right point, the pupil center moves to the left in the image of the user's eye. The coordinate transformation coefficients (β_{xn}, β_{yn}) are useful for evaluating the gaze ability of the user. Three examples are shown in Fig. 13. Case (A) is an ideal case, Case (B) presents a user with poor gazing ability when he gazes at the top point. Case (C) is that of a user with poor ability gazing at the right point. The coordinate transformation coefficients for these three cases are shown in Table 2.

From 1998 to 2000, a prototype system was developed for three broadcasting exhibitions in Taipei, Taiwan. In the public test bed, over 5000 people used this device. We found that nearly 10% of the users had problems gazing at the four corners of the screen (Table 3).

One background the game story example is described as follows. A superman can emit dead light with his eyes to fight against evil. First, he attacks a spaceport with his eyes. When he occupies the spaceport, he must emit laser light from his eyes to weld the cracks in the spaceport. He then uses the missile in the spaceport to destroy the enemy spaceship by eye gaze or blink function (Figs. 14(A) and (B)). Anytime the user closes his eye, the energy will increase and different music will play to tell him if the enemy will come from the right or left side. Moreover, the user will enter a 'blind mode' to play the game with the keyboard per 3 min. In the 'blind mode' user must close his eye to relax and hear the music to attack enemy. The energy status bar is on the right side of Fig. 14(C). All of the subroutines are packaged with Direct3D, dynamic link library (DLL) and component object model (COM).

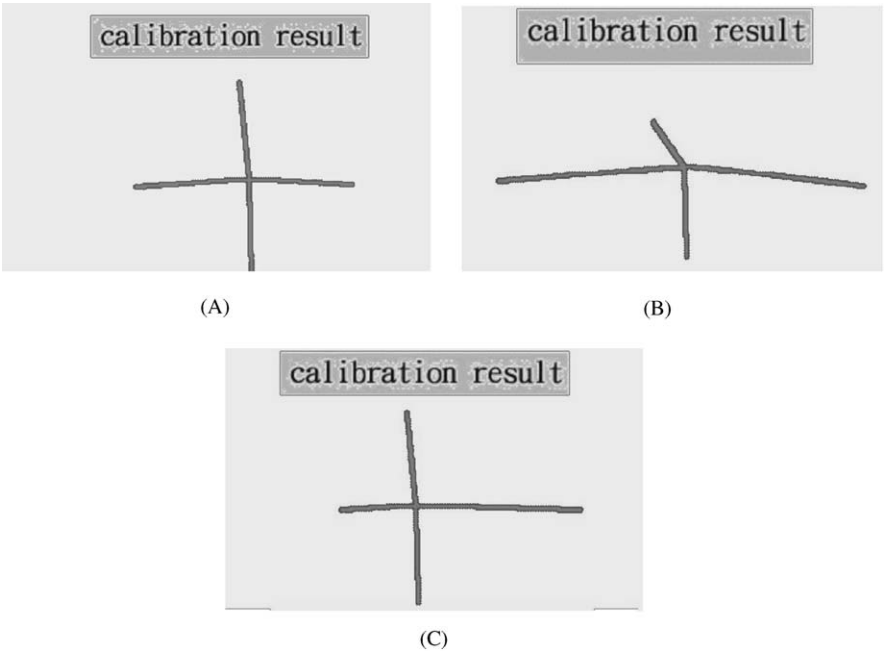


Fig. 13. (A) Ideal case, (B) poor gazing ability at the top point and (C) poor gazing ability at the right point.

Table 2
Coordinate transformation coefficients for three cases

Coefficients	Case (A)	Case (B)	Case (C)
β_{x1}	6	6.2	6.2
β_{x3}	6	6.3	19.3
β_{y2}	6	17.2	8.5
β_{y4}	6	9.3	8.9

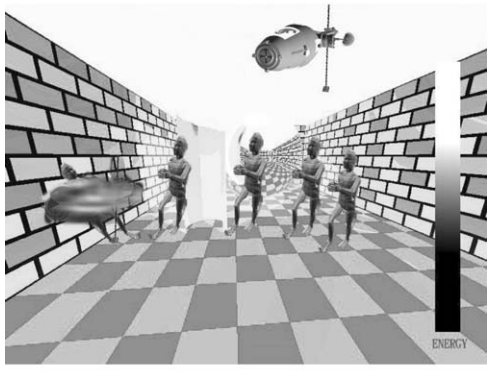
Table 3
Percentage of users with poor gazing ability at the four corners of the screen

Coefficients	Percentage
$\beta_{x1} > 15$	1.97
$\beta_{x3} > 15$	3.05
$\beta_{y2} > 15$	3.26
$\beta_{y4} > 15$	2.15

Because the eye-game system is very interesting, this computer game played using the eye-tracking device is suitable for rehabilitating eye movement dysfunction of children. We selected many users who had poor eye-tracking ability for preliminary

(A)

(B)



(C)

Fig. 14. Multimedia game screen with eye-tracking device.

testing. Fig. 15 shows the case study in which improvement in the eye-tracking ability was shown after using this computer game. We defined improvement degree using

$$I_{\mathrm{d}} = (\beta_{\mathrm{a}}/\beta_{\mathrm{b}} - 1) \times 100\%,$$

where β_{a} is the coordinate transformation coefficient for first-time use and β_{b} the coordinate transformation coefficient after the user has practiced for 2 weeks.

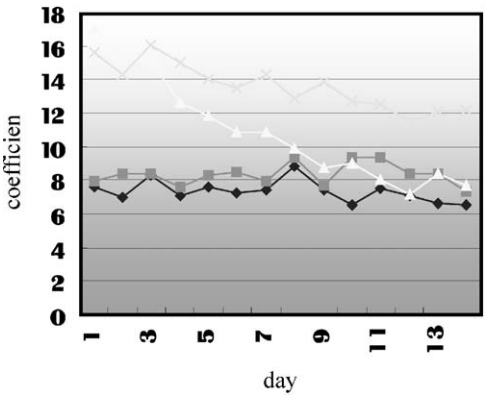


Fig. 15. Five studies showing eye improvement after using this computer game: ■, β_{x1}; ◆, β_{x3}; ▲, β_{y2}; ×, β_{y4}.

As shown in Fig. 15, we can see that the largest coordinate transformation coefficient is improved (decrease) by 50–110% after the user has practiced with this system 30 min/day for 2 weeks. Certainly, it is partially because the experience of the user is accumulated. However, we can also see that the subject's eye has become more agile. It is helpful for children who have problems in misalignment of the eyes or inefficiency in using both eyes together.

7. Conclusion

This eye-game system is composed of cross-line tracking, coordinate mapping and blink weighting methods. These methods are suitable for real-time extraction of the position of the pupil in the eye image because they are computationally inexpensive. These methods were applied to a man–machine interface and implemented on a prototype hardware system. Preliminary evaluation experiments using this prototype system were introduced to demonstrate the effectiveness of the eye-game system rehabilitative function. This system is worthy of further research.

Acknowledgements

This work was sponsored by the National Science Council, Chung-Shan Institute of Science and Technology, Taiwan, Republic of China under grant numbers NSC90-2614-H-035-001-F20 and BV91U14P.

References

[1] Chern-Sheng Lin, Chih-Chung Chien, Nanjou Lin, Chiao-Hsiang Chen. The method of diagonal-box checker search for measuring one's blink in eyeball tracking device. Opt Lasers Technol 1998;30(5):295–301.

[2] Wagner R, Galiana HL. Evaluation of three template matching algorithms for registering images of the eye. IEEE Trans Biol Eng 1992;39(12):1313–9.
[3] Spindler F, Chaumette F. Gaze control using human eye movement. IEEE International Conference on Robotics and Automation, 1997 Albuquerque, New Mexico. p. 2258–63.
[4] Grattan KT, Palmer AW. Interrupted reflection fiber optic communication device for the severely disabled. J Biomed Eng 1984;6:321–2.
[5] Grattan KT, Palmer AW, Sorrell SR. Communication by eye closure—a microcomputer-based system for the disabled. IEEE Trans Biomed Eng 1986;BME33:977–82.
[6] Xie X, Sudhakar R, Zhuang H. A cascaded scheme for eye tracking and head movement compensation. IEEE Trans Syst Man Cybern 1998;28(4):487–90.
[7] Martin CF, Schovance L. A control model of eye movement. Conference on Decision and Control, San Diego, CA, USA, 1997. p. 1135–9.
[8] Allison RA, Eizenman M, Cheung BSK. Combined head and eye tracking system for dynamic testing of the vestibular system. IEEE Trans Biol Eng 1996;43(11):1073–82.
[9] Talmi K, Liu J. Eye and gaze tracking for visually controlled interactive stereoscopic displays. Signal Process: Image Commun 1999;14:799–810.
[10] Pastoor S, Liu J, Renault S. An experimental multimedia system allowing 3-D visualization and eye-controlled interaction without user-worn devices. IEEE Trans Multimedia 1999;1(1):41–52.
[11] Myers GA, Sherman KR, Stark L. Eye monitor—microcomputer-based instrument uses an internal model to track the eye. IEEE Comput 1991;24(3):14–21.
[12] Kirby M, Sirovich L. Applications of the Karhunen–Loeve procedure for the characterization of human faces. IEEE Trans Patt Anal Mach Intell 1990;12(1):103–8.
[13] Yankang Wsng, Hideo Kuroda, Makoto Fujimura, Nakamura. Automatic extraction of eye and mouth fields from monochrome face image using fuzzy technique. Universal Personal Communications, 1995. p. 778–82.
[14] Xie X, Sudhaker R, Zhuang H. On improving eye feature extraction using deformable templates. Pattern Recognition 1994;27(6):791–9.
[15] Preston White K, Hutchinson TE, Carley JM. Spatially dynamic calibration of an eye-tracking system. IEEE Trans Syst Man Cybern 1993;23(4):1162–8.
[16] Chern Sheng Lin. An eye behavior measuring device for v.r. system. Opt Lasers Eng 2002;38(6): 13–39.
[17] Chern Sheng Lin, Kai-Chieh Chang, Young-Jou Jain. A new data processing and calibration method for an eye-tracking device pronunciation system. Opt Lasers Technol 2002;34(5):405–13.