Keeping an Eye for HCI

CARLOS HITOSHI MORIMOTO[1], DAVID KOONS[2], ARNON AMIR[2], MYRON FLICKNER[2], SHUMIN ZHAI[2]

[1]Departamento de Ciência da Computação do IME-USP - Rua do Matão 1010, São Paulo, SP 05508, Brazil
hitoshi@ime.usp.br
[2]IBM Almaden Research Center - 650 Harry Road, San Jose, CA 95120, USA
{dkoons,arnon,flick,zhai}@almaden.ibm.com

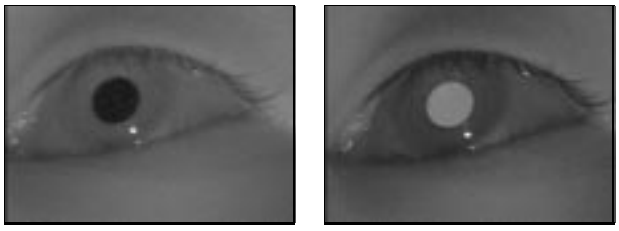
Abstract. Advanced Human Computer Interaction (HCI) techniques are required to enhance current computer interfaces. In this paper we present an eye gaze tracking system based on a robust low-cost real-time pupil detector, and describe some eye-aware applications being developed to enhance HCI. Pupils are segmented using an active lighting scheme that exploits very particular properties of eyes. Once the pupil is detected, its center is tracked along with the corneal reflection (CR) generated by the light sources. Assuming small head motion, the eye gaze direction is computed based on the vector between the centers of the CR and the pupil, after a brief calibration procedure. Other information such as pupil size and blink rate can also be made available. The current prototype runs at frame rate, providing 30 samples of the gaze position per second to all gaze-aware applications, such as advanced pointing and selection mechanisms.

1 Introduction

The idea of using eye gaze tracking for Human Computer Interaction (HCI) is not new. Hutchinson *et al.*[9] describe a computer system to provide nonverbal, motor-disabled individuals with a means to communication and environmental control. Jacob [10] describes several ways of using eye movements as input to HCI, and Glenstrup[8] also argues that it is possible to use the user's eye gaze to aid the control of a computer application, though care should be taken. Recently, Edwards [6] has proposed a development tool that can be used to create eye-aware software applications, which can adapt in real-time to changes in a user's natural eye-movement behaviors and intentions.

The major problems with current eye gaze tracking technology is its high cost and unreliability. Also, some eye tracking systems are cumbersome, requiring the use of helmets or glasses connected through cables to a computer, making it unsuitable for most general purpose HCI. In this paper we describe a robust low-cost real-time remote eye gaze tracking system, which we have been using to develop new eye-aware applications.

Commercial remote eye-tracking systems such as those produced by LC Technologies (LCT)[14], and Applied Science Laboratories (ASL)[11], are able to estimate a person's gaze or point of regard within a very limited area, restricting head motion to a certain small region. They rely on a single light source to facilitate pupil detection and tracking, placing the light on the optical axis of the camera. Illumination from an off-axis source (and normal illumination) generates a dark pupil image, as can be seen in Figure 1a. When the light source is placed on-axis with the camera optical axis, the camera is able to detect the light reflected from the interior of the eye, and the image of the pupil appears bright [9, 17], as shown in Figure 1b. This effect is often seen as the red-eye in flash photographs when the flash is close to the camera lens. These systems require the initial localization of the pupil and the selection of a carefully adjusted threshold in order to begin tracking.



a b

Figure 1: (a) Dark and (b) bright pupil images. Observe the very bright spot on the lower right part of the iris. This glint corresponds to the reflection of the light source from the cornea.

Our eye tracking system is based on a robust pupil detector that uses both dark and bright pupils to obtain a better signal-to-noise ratio. Pupils are detected from the subtraction of the dark from the bright pupil images, as described in Section 2.1. Although still not ideal for general purpose HCI, this system is our first step towards applying computer vision techniques to enhance HCI. The description of other topics of our research are given in [13].

The next section describes the eye gaze tracking system and explains how it is used to compute the user's gaze direction. Experimental results of this system are given in Section 3, Section 4 introduces some eye-aware applica-

tions being developed to enhance HCI, and Section 5 concludes the paper.

2 Eye Gaze Tracking

The purpose of an eye gaze tracker is to estimate the scene location to where a user is fixating her gaze. This is accomplished by tracking the user's eye movements, and in general require a calibration procedure, that determines the correspondence of the eye movements to scene coordinates. Thus, it is fundamental for an eye tracker to record the movements of the eye. Young and Sheena [17] describe several methods for recording eye movements, that include electro-oculography, limbus tracking, corneal reflection, and contact lens techniques. A comparison of several of these techniques is given in [8].

Both commercial systems mentioned earlier use corneal reflection methods. Such techniques, based on reflected light, seem to be the most appropriate for HCI applications because they are non-invasive, fast, and reasonably accurate.

Traditional corneal reflection techniques require a single light source to generate the corneal reflection (CR). Assuming a static head, an eye can only rotate in its socket, and the surface of the eye can be approximated by a sphere. Since the light source is also fixed, the reflection on the cornea of the eye (glint) can be taken as a reference point, thus the vector from the glint to the center of the pupil will describe the gaze direction. The CR can be easily noticed in Figure 1. Some drawbacks of these methods are the requirement to keep the head still (though some systems allow for small head motion), and the difficulty to obtain and keep a good contrast image to facilitate the segmentation of the CR and the pupil.

To optimize the segmentation process, methods based on multiple light sources to generate dark and bright pupil images are suggested in [15, 5]. The principle is to enhance the signal-to-noise ratio, and detect the pupils by simple image subtraction of the dark from the bright pupil image, as shown in Figure 4c. Tomono *et al.*[15] describe a real-time eye tracking system composed of a 3 CCD camera and 2 near infra red (IR) light sources with different wavelengths. Ebisawa and Satoh [5] also use two light sources, but both with the same wavelength to generate the bright/dark pupil images. Ebisawa [4] also presents a real-time implementation of the system using custom hardware and pupil brightness stabilization for optimum detection of the pupil and the CR. These systems are quite complex and require custom hardware to be implemented.

We have developed an eye gaze tracking system based on a robust pupil detector that also uses the difference between bright and dark pupil images, but it is simpler, inexpensive, and can be built using standard off the shelf components. The pupil detector is also able to process wide field of view images [12], in frame rate, to segment faces. Other eye and face tracking systems such as those described in [1, 7, 16] could also benefit from robust pupil detection techniques. In particular desktop and kiosk [2] applications, which require a detection range of a few meters, are suitable for this technique.

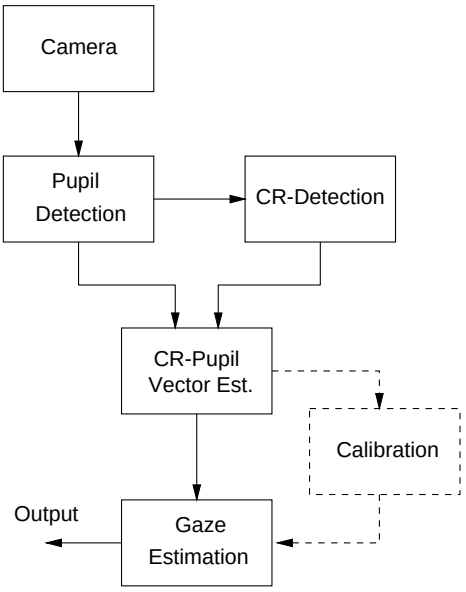


Figure 2: Block diagram of the eye gaze tracking system.

Figure 2 shows a block diagram of the eye tracking system. During regular operation, shown by the solid line, the pupil and CR are detected, and the centers of the pupil and CR are computed to generate the CR-pupil vector. This vector is used to estimate the coordinate on the screen where the user is looking at, after a brief calibration procedure. This procedure is very brief and creates a mapping between the CR-pupil vector space into screen coordinates (see Section 2.3). Next, detailed descriptions of each system component are given.

2.1 Pupil Detector

Our robust low-cost real-time pupil detector uses an active lighting technique to find pupil candidates. We have built functioning imaging prototypes using B/W board cameras, a pan-tilt servo mechanism, and the illuminators for under US$1,500. The system also requires a PC workstation with a digitizer card, which can be acquired for less than US$2,000. Our quotes for a complete remote eye tracking system were around US$20,000 (Oct/97).

Figure 3 shows one possible configuration of the pupil detection system. The system uses one camera and two light sources, LIGHT1 and LIGHT2. For convenience, near infra-red (IR) LED's with wavelength 875nm (invisible to

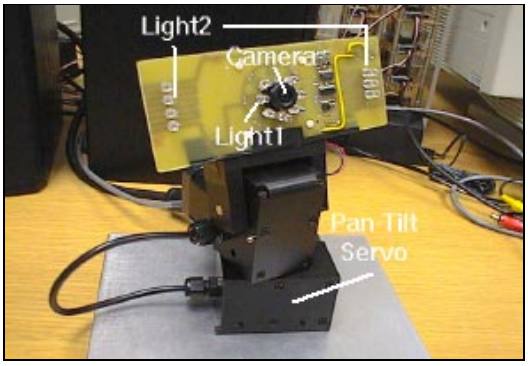


Figure 3: Camera and IR illumination setting. LIGHT1 is placed around the optical axis of the camera to generate the bright pupil image, and LIGHT2 is placed off-axis to provide about the same illumination, but generating a dark pupil image.

the human eye) are used. Figure 3 also shows an inexpensive black and white 1/3" CCD board camera with a visible light blocking filter. The camera is about 40×40×15mm in size, and the lens is 12mm in diameter. Several focal lengths were used.

LIGHT1 is placed very close to the optical axis of the camera to generate the bright pupil image, as seen in Figure 4a, and LIGHT2 is placed off-axis, farther from the optical axis, to provide about the same scene illumination, but generating a dark pupil image (Figure 4b).

The video signal from standard NTSC cameras is composed of interlaced frames, where one frame is composed by an even and an odd field. Thus, a field has half the vertical resolution of a frame. Let F_t be an image frame taken at time instant t, with resolution c columns (width) by r rows (height), or $c \times r$. F_t can be de-interlaced into E_t and O_t, where E_t is the even field composed by the even rows of F_t and O_t is the odd field composed by the odd rows of F_t.

We have developed a simple synchronization device that keeps LIGHT1 on and LIGHT2 off when the camera is scanning E_t, and LIGHT1 off and LIGHT2 on when the camera is scanning O_t. The digitizer card grabs 30 interlaced frames per second, which are de-interlaced by software. For the computation of the set of pupil candidates at time t, it is considered that a dark pupil image is always subtracted from the bright pupil image, i.e., the difference image D_t is computed as $D_t = E_t - O_t$ [1]. It follows that the regions correspoding to pupils will be always positive. A thresholding operation is performed, and the resulting binary image is processed by a connected component labeling algorithm. Geometrical constraints based on the shape and size of each connected component are then applied to eliminate false positives.

We have also developed a frame-based pupil detection technique, which is limited to 30 frames per second, but which allows the full frame resolution to be used. This technique was not pursued further because it requires a messaging mechanism to allow the computer to determine when a bright or dark pupil image was grabbed. Synchronism is also harder to keep in this case, particularly when the system drops frames due to other system constraints. The faster field rate also helps reducing motion artifacts.

2.2 CR-Pupil Vector

The pupil detection system, as described in Section 2.1, can output several pupil candidates, filtered by the high contrast and geometrical constraints. Since the field of view of the camera is very narrow, the pupil is selected as the biggest pupil candidate, and the center of mass of the segmented region is used as the center of the pupil.

A search for bright pixels around the center of the pupil, using the dark pupil image, finds the glint and its center of mass. Figure 5 shows the bright, dark, and the dark pupil image with two crosses superimposed, that correspond to the computed centers of the pupil and glint.

2.3 Calibration Procedure

To estimate the screen coordinates to where the user is looking at, we use a simple second order polinomial transformation computed from an initial calibration procedure. Once the system is calibrated, a simple possible application is to control the screen cursor using eye gaze (eye mouse), which provides an estimate about the accuracy of the system. We have obtained an accuracy of about 1 degree of resolution, that corresponds to about 1cm on the screen looking from a distance of 50cm.

The calibration procedure is simple and brief. Nine points in a 3×3 grid on the screen are displayed, in sequence, and the user is asked to fixate her gaze on the target point, press a key, and move to a next displayed target. On each fixation, the vector from the center of the CR to the center of the pupil is saved, so that 9 corresponding points are obtained. The transformation from a CR-pupil vector $\mathbf{E} = (x_e, y_e)^t$, to a screen coordinate $\mathbf{S} = (x_s, y_s)^t$ is given by:

$$\begin{aligned} x_e = &\ a_0 + a_1 x_s + a_2 y_s + a_3 x_s y_s + a_4 x_s^2 + a_5 y_s^2 \\ y_e = &\ a_6 + a_7 x_s + a_8 y_s + a_9 x_s y_s + a_{10} x_s^2 + a_{11} y_s^2 \end{aligned} \quad (1)$$

where a_i are the coeficientes of this second order polynomial. Each corresponding point gives us 2 equations from (1), so 9 points produce 18 equations, and an overdetermined linear system is obtained. The coefficients of the

[1] Observe that pupils could be detected at 60Hz by also considering a second difference image $D'_t = E_t - O_{t-1}$, i.e., the difference between fields from consecutive frames.

(a) (b) (c)

Figure 4: (a) Bright and (b) dark pupil images. (c) Difference of the dark from the bright pupil after thresholding.

polynomial can be obtained independently, thus 2 overdetermined linear systems with 6 unknowns and 9 equations each are solved using a least squares method.

2.4 Pan-Tilt Servo Mechanism

In order to allow some head motion, it is required to keep the pupil centered in the image. The magnitude of the rotation angle of the camera which brings the pupil to the center of the image (assuming the rotation is around its principal point), will only depend of the image size and the field of view (FOV) of the camera. If the center of the pupil is at the pixel (x, y), and given the FOV $= (\phi_x, \phi_y)$, and image size $W \times H$, the pan and tilt are given by:

$$\text{pan} = \phi_x x/W; \qquad \text{tilt} = \phi_y y/H. \tag{2}$$

3 Experimental Results

The current eye gaze tracker prototype was implemented in a dual Pentium II 400MHz machine running Windows NT4, using a commercial PCI frame grabber compatible with Video for Windows. The eye tracker runs at 30 frames per second, grabbing interlaced frames of resolution 640 $\times$ 480 $\times$ 8 bits.

Figure 4 shows a bright and dark pupil images, and the difference of the dark from the bright pupil image after thresholding. The differencial technique using active lighting detects the pupil candidates from this binary image. Contact lens do not interfere with the detection process, but eyeglasses can generate specular reflections that might result in false positives (Figure 6). Observe in Figure 6c that the pupil still corresponds to the biggest blob after thresholding, but these spurious reflections can also block the dark pupil response under very particular head orientations. In most cases, since head motion must be restricted, a slight change in the orientation of the glasses is enough to restablish detection and gaze estimation. Pupil detection using only the dark or only the bright pupil images, as it is done by most commercial eye trackers for gaze estimation, would have a lot more spurious responses, which can be expected from images of the same kind shown in Figure 6.

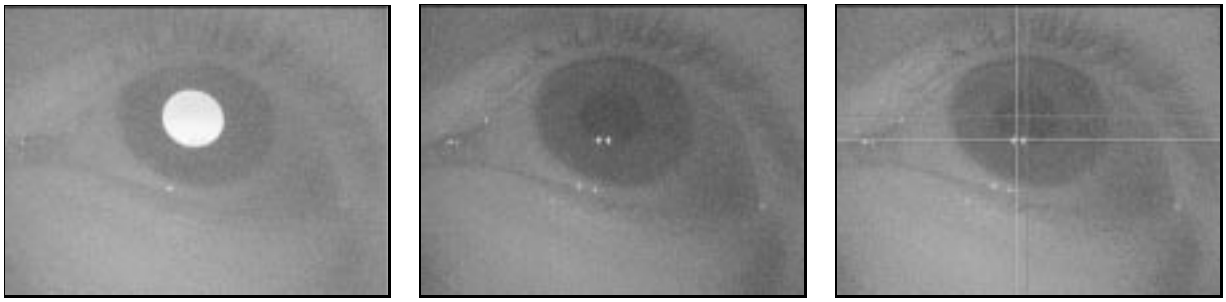
Figure 5 shows bright and dark pupil images, and the dark pupil images with two crosses that correspond to the centers of the pupil and CR, using the maximum magnification of the lenses to obtain the best accuracy. Observe that the glint cannot be detected using the bright pupil image (Figure 5a) due to the saturation of the bright pupil, and that two distinct glints are actually present in the dark pupil image (Figure 5b). The two glints are generated by LIGHT2, the off-axis light source, which is composed of seven IR LED's placed symmetrically on the sides of the lenses (Figure 3).

The performance of the system is similar to the commercial ones, providing about one degree accuracy and restricted head motion with the pan-tilt servo mechanism, but it is a more affordable and robust alternative. The simple model used is not strong enough to deal with large head motion though, because the calibration should also be a function of head position. Thus, more complex models will be required to handle free head motion.

Our eye tracker has been successfully tested for a very large number of people, and it has proven to be very robust indoors, although it has not been tested outdoors, where natural lighting with high intensity IR illumination might introduce difficulties.

4 Eye-Aware Interfaces

User interfaces based on eye tracking systems have the potential for faster and more natural interaction than current standard interfaces. Jacob [10] describes ways of using eye tracking devices for pointing and selection, and the challenge of building a useful interface without the problem of activating everything that the user looks at (known as the Midas Touch problem). His initial experiments indicate a 30% improvement on selection time using selection by dwell time over standard mouse input. Hutchingson *et al.*[9], Glenstrup [8], and Colombo and Del Bimbo [3] also describe applications using selection by dwell time.



(a) (b) (c)

Figure 5: (a) Bright and (b) dark pupil images. (c) Dark pupil image showing the centers of the pupil and corneal reflection.

When dwell time is adopted for selection, the user has to adjust her behavior to avoid looking at objects for long periods, otherwise that object can be activated when the user had no intention of doing so. Alternatives to this method is the use of clutch mechnisms such as buttons, footpedals, or other mechanical devices, to engage and disengage control activities such as selection and dragging.

A more fundamental question that we pose regards the adequacy of the use of eye-gaze for pointing and selection in general computer interfaces, i.e., would an eye-mouse become a popular device, once the cost issues have been solved? People are accostumed to use their eyes for exploration (sensory input) and not for manipulation (output), thus further studies have to be conducted to verify if most users would eventually adapt or simply reject such interfaces. Also, given the current state of eye tracking technology, fine pointing in small high resolution displays is also not possible, what restricts the size of the displayed objects that can be selected.

Zhai et al. [18] introduces an elegant way of combining eye-gaze rapid movements with the high accuracy of current manual pointing devices, e.g., a regular mouse. The basic idea is to move the cursor to where the user has fixated her gaze only when the user demonstrate intent to do so, i.e., touches the mouse. If the cursor was originally far from the target point, it is immediatelly warped to near the target position, according to the precision of the eye tracker, and then fine position adjustments are made manually.

Even if eye-gaze prooves to be inadequate for pointing and selection in general computer interfaces, we have proposed alternative ways of using eye-gaze information for HCI. For example, eye-gaze information can be used to determine eye contact, thus helping disanbiguate speech commands in a environment populated with speech-aware devices, or setting the context for some applications, such as pre-fetching hyperlinks near the position being read by the user, and counting the number of times the user looks at certain regions of the screen, which might be advertisements, or the rear mirror or obstacles of a simulated driving test. These and other research work is described at [13], and will the the subject of future publications.

5 Conclusion

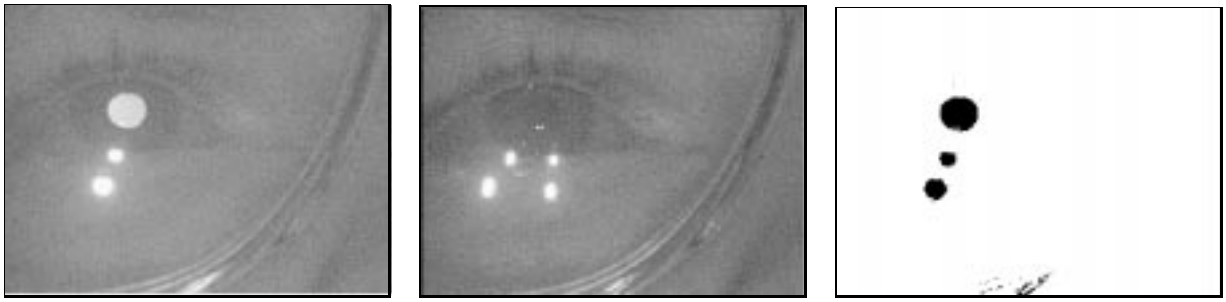
Efficient and robust techniques for eye detection in images are of particular importance to HCI. Information about the eye behavior can be used as indicators of the user's internal state, or simply used to determine the position and orientation of the face, for face authentication purposes, monitoring human activity, multi-modal interfaces, etc.

We have presented an inexpensive and yet robust and reliable real-time eye gaze tracker, with very high potential to be used in HCI. The even and odd fields of a video camera are synchronized with two IR light sources. A pupil is alternately illuminated with an on-axis IR source when even fields are being captured, and with an off-axis IR source for odd fields. The on camera axis illumination generates a bright pupil, and the off axis illumination keeps the scene at about the same illumination, but the pupil remains dark. Detection follows from thresholding the difference between even and odd fields. Once the pupil is detected, its center is tracked along with the corneal reflection (CR) generated by the light sources. Assuming small head motion, the eye gaze direction is computed based on the vector between the centers of the CR and the pupil, after a brief calibration procedure. A real-time prototype is current running at 30 frames per second (frame-rate) using interlaced images of resolution $640 \times 320 \times 8$ bits, on a dual PII 400MHz platform.

Future extensions include the generalization of the problem to more complex models in order to allow for free head motion, and the development and performance studies of new eye-aware computer interfaces.

Acknowledgements

We would like to thank Chris Dryer, Dragutin Petkovic, Steve Ihde, Wayne Niblack, Wendy Ark, Xiaoming Zhu, and the other people involved in the BlueEyes project for



(a) (b) (c)

Figure 6: (a) Bright and (b) dark pupil images for a person with glasses. (c) Difference image.

their valuable discussions and contributions during the development of this project.

References

[1] S. Birchfield. An elliptical head tracker. In *Proceeding of the 31st Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, November 1997.

[2] A.C. Christian and B.L. Avery. Digital smart kiosk project. In *Proc. ACM SIGCHI - Human Factors in Computing Systems Conference*, pages 155–162, Los Angeles, CA, April 1998.

[3] C. Colombo and A. Del Bimbo. Interacting through eyes. *Robotics and Autonomous Systems*, 19:359–367, 1997.

[4] Y. Ebisawa. Unconstrained pupil detection technique using two light sources and the image difference method. *Visualization and Intelligent Design in engineering and architecture*, pages 79–89, 1995.

[5] Y. Ebisawa and S. Satoh. Effectiveness of pupil area detection technique using two light sources and image difference method. In A.Y.J. Szeto and R.M. Rangayan, editors, *Proceedings of the 15th Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society*, pages 1268–1269, San Diego, CA, 1993.

[6] Gregory Edwards. A tool for creating eye-aware applications that adapt to changes in user behavior. In *Proc. of ASSETS 98*, Marina del Rey, CA, April 1998.

[7] A. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, February 1996.

[8] A. Glenstrup and T. Engell-Nielsen. Eye controlled media: Present and future state. Master's thesis, University of Copenhagen DIKU (Institute of Computer Science), Universitetsparken 1 DK-2100 Denmark, June 1995.

[9] T.E. Hutchinson, K.P. White Jr., K.C. Reichert, and L.A. Frey. Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics*, 19:1527–1533, Nov/Dec 1989.

[10] R.J.K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems*, 9(3):152–169, April 1991.

[11] Applied Science Laboratories. URL: http:// www.a-s-l.com.

[12] C. Morimoto, D. Koons, A. Amir, and M. Flickner. "real-time detection of eyes and faces". In *Proceedings of 1998 Workshop on Perceptual User Interfaces*, pages 117–120, San Francisco, CA, November 1998.

[13] IBM Almaden Research Center: BlueEyes Project. URL: http:// www.almaden.ibm.com/ cs/ blueeyes.

[14] LC Tecnologies. URL: http:// www.lctinc.com.

[15] A. Tomono, M. Iida, and Y. Kobayashi. A tv camera system which extracts feature points for non-contact eye movement detection. In *Proceedings of the SPIE Optics, Illumination, and Image Sensing for Machine Vision IV*, volume 1194, pages 2–12, 1989.

[16] J. Yang and A. Waibel. A real-time face tracker. In *Proceedings of the Third IEEE Workshop on Applications of Computer Vision*, pages 142–147, Sarasota, FL, 1996.

[17] L. Young and D. Sheena. Methods & designs: Survey of eye movement recording methods. *Behavioral Research Methods & Instrumentation*, 7(5):397–429, 1975.

[18] S. Zhai, C.H. Morimoto, and S. Ihde. Manual and gaze input cascaded (magic) pointing. In *Proc. ACM SIGCHI - Human Factors in Computing Systems Conference*, pages 246–253, Pittsburgh, PA, May 1999.