# A Computer Vision Framework for Eye Gaze Tracking

Marcio R. M. Mimica and Carlos H. Morimoto
Departamento de Ciência da Computação - IME/USP
Rua do Matão 1010, São Paulo, SP
{mimica,hitoshi}@ime.usp.br

## Abstract

*Eye gaze tracking (EGT) allow us to estimate the direction of gaze and the point of regard. This technique has been successfully used as pointing device in computer interfaces for people with disabilities, but significant technological advances are still required to make EGT appropriate to be used in general computer interfaces. In order to improve the EGT technology, a testbed where new devices and algorithms can be evaluated must be defined. This paper presents a survey of the methods used for EGT, and organize them into a computer vision framework, that we use to support the development and evaluation of new techniques. As an example of application of the framework, the calibration function used in the EGT developed in our laboratory is tested, and the results show the precision of the method. Higher accuracy of the testing results is achieved using synthetic images generated by ray tracing, from a physically based model of the eye.*

## 1. Introduction

One of the main research goals of Human Computer Interaction (HCI) is to devise new interfaces to allow more efficient interactions between humans and computers. Traditional graphic interfaces have focused on the communication from the computer to the user, taking advantages of the large bandwidth of the human vision channel. Communication from the user to the computer has been restricted to much slower channels, mostly mouse and keyboard. The use of other means of communication may reduce this gap and make future interfaces more user friendly.

An eye gaze tracker (EGT) is a device that computes the user's direction of gaze. If information about the objects in the scene is available, the point of regard can be computed as the intersection between the direction of gaze and the surface of the objects. For HCI purposes, the monitor screen can be considered the object of interest, and the EGT device could be used, for example, to control the cursor, as a pointing device [8].

The majority of eye gaze research has concentrated on using eye movement data to study perceptual or cognitive processes [13] and the eye movement control mechanism itself. In these situations, the eye movements are just recorded during the experiment, requiring retrospective analysis of the collected data. Likewise, the technology can be applied to testing software usability [2] and the effectiveness of marketing materials.

The use of eye movements in human-computer dialogues has been proven to be useful in interfaces for people with physical disabilities [3, 15, 7], but since the ability of these people to operate other devices is limited, an interface normally rejected as awkward or unnatural might still be effective. There is also research on using EGTs in situations where the hands of the user are busy performing other tasks, such as during airplane piloting [14] or to avoid repetitive strain injury caused by the mouse.

Recent research seeks to use gaze tracking to provide an effective input method to the majority of users [8, 6, 18, 20]. It has been demonstrated that interaction techniques based on gaze tracking are at least as fast as the mouse [9], but EGT technology has not reached the maturity to be applied to general computer interfaces. The main problems with current technology, just to name a few, are its high cost (a commercial system like LC Technologies' Desktop Eye Gaze System costs about US $15,000 [5]), the difficulty to calibrate the device, and its intolerance to head movement.

In [12] we describe a low cost but robust EGT device. A short description of the device is given in Section 2.3. The robustness of the system makes it easier to calibrate, but the system still do not compensate for head motions, requiring the user to keep his/her head still. This paper introduces a computer vision framework that will serve as a testbed for the development and test of new EGT techniques. The framework also defines the method for evaluating the techniques using synthetic images generated by ray tracing, using a physical model of the eye.

To help the performance comparison of other techniques,

Proceedings of the XVI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'03)
1530-1834/03 $17.00 © 2003 **IEEE**

in particular those suitable for HCI purposes, the next section presents the current state of the art in eye gaze tracking technology, discusses their advantages, and points out their problems. The EGT system used in our experiments is described in Section 2.3. Section 3 details the computer vision framework used to continue our research on developing and evaluating EGT devices based on computer vision techniques. Sections 4 and 5 present the experimental tests and the results of applying the framework to evaluate the calibration function used in our EGT system, and Section 6 concludes the paper.

## 2. Eye Gaze Tracking Techniques

Current gaze tracking techniques may be classified as intrusive and non-intrusive. Intrusive techniques require some kind of physical contact with the user, and non-intrusive, or remote techniques, can be done without any physical contact. Since physical contact can be a cause of discomfort, non-intrusive methods are more appropriate for general HCI applications.

### 2.1. Intrusive Techniques

The degree of discomfort is also related to the intrusiveness of the method. The most intrusive ones use some device in direct contact with the eye, such as contact lenses. Some methods require direct contact with the skin, while other methods require devices to be in a fixed position relative to the eye or head, such as helmets.

**2.1.1. Contact lenses.** It is possible to track the user's eye and compute its direction of gaze with high accuracy (about $0.08^o$) using special contact lenses [4].

There are at least three known techniques based on contact lenses.In the simplest method, the orientation of the eye can be computed directly from mechanical levers placed on the surface of the lenses.

A second method use very small mirrors placed on the lenses, so that the orientation of the eye and the point of regard can be computed from the reflections of light beams directed to the mirrors.

A more sophisticated method use tiny induction coils placed inside the lenses. High frequency electro magnetic fields around the user's head allow the measurement of the eye orientation with high precision.

Despite the high accuracy, it's the most intrusive technique. Non-slipping contact lenses are grounded to fit precisely over the cornea, and then a slight suction is applied (mechanically or chemically) to hold the lens in place. It is a very uncomfortable method and interferes with blinking, thus its use is restricted to laboratory studies.

**2.1.2. Electro Oculography.** The electro-oculography tracking technique is one of the least expensive. It is based on the existence of an electrostatic field that rotates along with the eye. Electrodes are placed on the skin around the eye socket, and the movements of the eye are detected from small differences in the skin potential captured from the electrodes.

This technique does not require a clear view of the eye, which results in large dynamic range of approximately $70^o$ [4]. But, it also requires direct contact with the user, electrodes in this case, and according to Shaviv *et al.* [16] there are several problems related to head and muscle movement interference, signal drift, and channel crosstalk. It has low accuracy of 1.5 to $2^o$ [4].

**2.1.3. Image Based Methods.** There are several image based techniques, such as limbus tracking, pupil tracking, pupil and cornea reflection tracking, artificial neural networks, Purkinje image tracking, and so on. The camera can be placed on a helmet, therefore in contact with the user, or placed somewhere in the scene, i.e., remotely.

The limbus is the boundary between the white sclera and the iris, which is the colored part around the pupil. Because the high contrast between the iris and the sclera, the limbus is easier to track than the pupil. Occasional covering of the top and bottom of the limbus is a problem solved by the "Longest Line Scanning" algorithm [10], which computes the limbus center by finding the center of the longest horizontal line in the visible part of the iris. Typically, systems based on limbus tracking present better horizontal than vertical accuracy (about 1 to $7^o$ of vertical accuracy and 0.5 to $7^o$ of horizontal accuracy [4]).

Pupil tracking is similar to limbus tracking, but the boundary of the pupil and the iris is used instead. The advantages of this technique over limbus tracking are that the pupil is not covered by the eyelids and the border of the pupil is often sharper than of the limbus, resulting in higher accuracy. But the contrast difference between the pupil and iris is lower than between the iris and sclera, making it harder to detect the edges of the pupils.

Both techniques are based on the position and shape of the limbus or the pupil relative to the head, so either the head must be held still, and in this case a chin-rest or a bite-bar is used, or the equipment must be fixed to the user's head. A small mark attached to a user's glasses can be adopted as a reference point [10].

### 2.2. Non-Intrusive Techniques

The non-intrusive techniques are image based. Using computer vision and geometric properties of the eye, it is possible to compute the gaze direction without the need for any kind of physical contact with the user.

A ray of light entering the eye creates several reflections on the boundaries of the lens and the cornea, called Purkinje images. The first and fourth Purkinje images are used to compute the gaze direction by the Dual-Purkinje technique. Despite its high accuracy of $0.017^o$, it requires a highly controlled light incidence, because the fourth Purkinje image is rather weak. Thus, it is not a technique suitable for applications outside controlled environments such as those in laboratories.

Artificial Neural Networks (ANN) [1] can also be used to estimate the gaze direction. Since it needs to see only a small region around the eye, the entire face can be in the field of view of the camera, increasing the user head mobility. Training the ANN with images of the user's eye and head is necessary, but once it is done no calibration or retraining is necessary before each new session. The low accuracy of 1.3 to $1.8^o$ [17] is its main disadvantage.

Based on this short survey, it is not surprising that most commercial systems are based on the pupil-cornea reflection technique, since it is non-intrusive and it does not need a controlled ambient light, such as the dual-Purkinje image technique, and has better accuracy than the technique based on ANN. Next we describe the EGT device developed in our laboratory, also based on this method, and that relies on dual near infrared light sources to enhance the performance of the system.

### 2.3. Robust Eye Gaze Tracker

Most commercial remote eye gaze tracking devices use a single near infrared light source to generate a reflection (glint) on the surface of the cornea (first Purkinje image). This reflection is generally used as a reference point. Consider a spherical cornea, so the position of the glint does not change with cornea rotations, although the pupil position does. The 2D vector defined by the glint and the pupil center can be used to compute the point of regard using a direct mapping from the pupil-glint vector to computer screen coordinates. The mapping is computed during the calibration procedure, where typically the user has to fixate his/her gaze at a few known scene points in a particular order. Although the calibration procedure might take only a few seconds, it has to be made before each user session, and once the calibration is made, the head must remain still.

Figure 1 shows a picture of our eye gaze tracker, which was developed in collaboration with the IBM Almaden Research Center. It consists of two near infra-red light sources, which are synchronized with the even and odd fields of the interlaced video signal from the black and white camera, as described in [12]. The use of two sets of near infra-red lights instead of one increases the robustness of the EGT because the pupils can be more easily detected. The first light source is positioned near the camera's optical axis. When this source is on, the camera sees a bright pupil, as in Figure 2a. This phenomenon is similar to the one that creates the red-eye effect on flash photographs. The second light source is positioned farther from the optical axis, and generates a dark pupil image, as can be seen in Figure 2b.

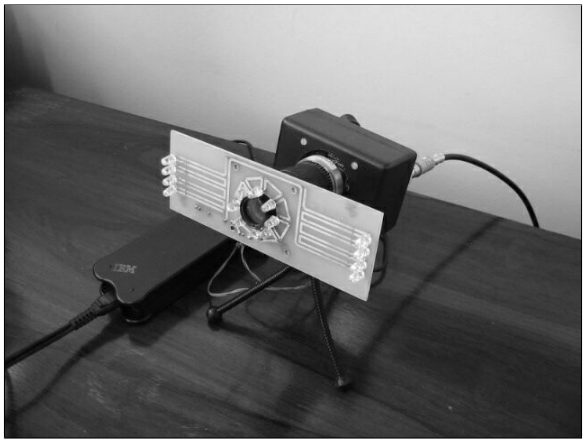


**Figure 1. Eye gaze tracker**

By subtracting the dark pupil image from the corresponding bright pupil image, and binarizing the difference image with a threshold, pupils can be easily detected (Figure 2c). First, a connected component labeling algorithm detects all the blobs in the binary image. Then, geometric properties of each blob, such as size and shape, are tested to filter blobs with unexpected shapes, that cannot correspond to pupils. Finally, the largest pupil candidate is selected.

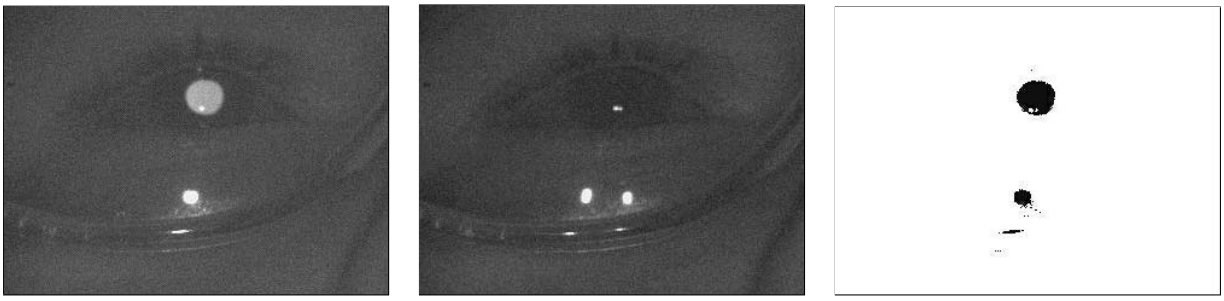
### 2.4. Calibration Procedure

Assuming a static head, an eye can only rotate in its socket, and the surface of the cornea of the eye can be approximated by a sphere. Since the light sources are also fixed, the glint on the cornea of the eye is also stationary and is taken as a reference point. The vector from the glint to the center of the pupil represents the gaze direction in image coordinates.

In order to transform the pupil-glint vector in image coordinates to screen coordinates, we need to compute the following mapping:

$$\mathbf{x_s} = \mathbf{A} * \mathbf{x_e} \tag{1}$$

where $\mathbf{x_s} = (x_s, y_s)^T$ is the column vector that represents the screen coordinates of the point of regard, and $\mathbf{x_e} = (x_e, y_e)^T$ is the pupil-glint column vector that represents the direction of gaze. $\mathbf{A}$ is a matrix that contains the transformation between the two groups of points.



(a) (b) (c)

**Figure 2. Bright (a) and dark (b) pupil images resulting from on and off axis illumination. (c) shows the difference image of the dark image subtracted from the bright.**

We use a second order polynomial transformation, defined as:

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

where $a_i$ are the coefficients of this second order polynomial, $(x_e, y_e)^t$ is the pupil-glint vector and $(x_s, y_s)^t$ are the screen coordinates. A set of corresponding points can be obtained by a calibration procedure, where the user is asked to fixate her/his gaze at certain known targets on the screen. Each corresponding point defines 2 equations from (2). For 9 different known targets, 18 equations are produced and an over determined linear system is obtained. The polynomial coefficients for $x_s$ and $y_s$ can be obtained independently, so that 2 simpler overdetermined systems can be solved using least squares. Notice that only 6 point correspondences would be enough to compute the mapping.

Besides the increased robustness due to the pupil detection scheme, another great advantage of this system is its low cost, since it was built using off the shelf components.

## 3. Computer Vision Framework

Testing the performance of a computer vision system is a challenging problem because it is very hard to control the input signal due to changes in camera parameters, noise, objects in the scene, illumination conditions, and so on. Besides, most vision systems have a large number of parameters, such as thresholds or window sizes, that have a great influence on its performance, and finding their optimal settings is not quite straight forward. Therefore, in order to further develop the system in a systematic way, the following general framework is proposed. It is based on a pipeline model that represents the modules required to build a remote EGT device based on computer vision techniques, and the flow and transformation of the information through the pipeline. A block diagram of the model is shown in Figure 3. Most current commercial remote EGT have a similar architecture.

The Image Acquisition (IA) module is responsible for converting the input signal from the sensor, or multiple sensors, to a suitable format. For example, the sensor could be a stereo camera that outputs a 3D image to the Image Processing (IP) module, and in the case of the system described in Section 2.3, the module is responsible to deinterlace the video signal and generate the bright and dark images.

The IP module processes the input data to filter and transform the image information before features and other characteristics can be processed by the Feature Estimation (FE) module. The image difference and binarization processes are done in IP module, and the blob segmentation and ellipse fitting are examples of the functions performed by the FE module.

These features are then used, maybe in combination with a calibration information, to estimate the direction of gaze by the Gaze Estimation (GE) module. The calibration includes any information from the user that is required to

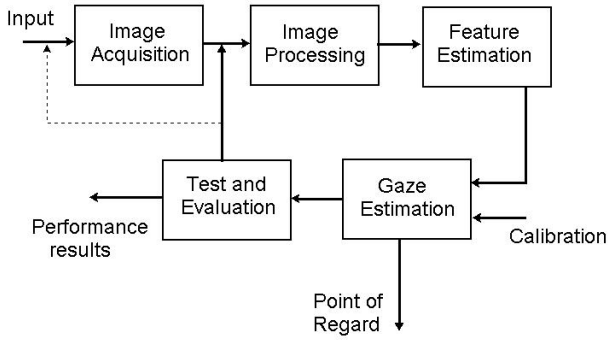



**Figure 3. Block Diagram of an EGT Device**

compute the direction of gaze, and cannot be assumed by the model or easily measured by the system. Finally, the point of regard of the user can be sent to gaze aware applications.

The framework includes a test and evaluation module, that can feed appropriate information for the IP module (or another module such as the IA or FE modules). The performance evaluation can be done more precisely using synthetic images, but real data can be used as well.

Despite the fact that most computer vision systems for EGT follow somehow a similar structure, new research could benefit from a framework. With this, it's easier to compare the steps of different algorithms. It also has a structure that allows one to combine the most effective modules of different algorithms and evaluate these new combinations. For example, in [12], the center of the pupil is computed as the center of mass of the blob. With this framework, we could try an algorithm that computes the center of the pupil by fitting an ellipse using the edges of the blob. As a result of this experiment, we observed a more accurate tracking of the pupil center, because in the previous method the glint could interfere in the computation of the center of mass. The glint is defined as a small region of bright pixels near the center of the pupil (often it appears within the pupil region) and is detected by a simple search algorithm using the dark pupil image. The ellipse fitting turn out to be robust even when the glint appears on the edge of the pupil, since most of its edges are not affected.

## 4. Evaluation and Tests

This section shows how the framework can be used to test and evaluate EGT devices. The following experiment shows the contribution of the calibration function as a source of intrinsic errors of the system.

To eliminate other possible sources of errors, in particular due to imprecision and noise of the IA and IP modules, we will use synthetic images generated by ray tracing of a physical model of the eye based on the Gullstrand's eye model [11].

### 4.1. Eye Model for Ray Tracing

The human eye has an approximately spherical shape with a radius of about 12 mm [19]. The cornea is a transparent membrane, void of blood vessels, that protrudes toward the front of the eye, covering the iris. The iris has a circular aperture in the center, called pupil, which regulates the amount of light coming into the eye. Behind the iris there is the lens, a biconvex multilayered structure. The shape of the lens changes during accommodation, a process that allows to bring the image of an object to a sharp focus in the retina. Between the cornea and the lens lies the anterior chamber which is filled with the watery aqueous humor and in the space between the lens and the retina is the transparent gelatinous vitreous body. The light that penetrates the retina, has traversed the whole eye optic media, suffering reflection and refraction at each media boundary.

Different eye models describe the human eyes optical characteristics under different complexity levels. Gullstrand's schematic eye was adopted. Table 1 shows the light path through the cornea until the retina. In this model, eye structures are set as spherical surfaces.

| | Position (mm) | Radius (mm) | Refraction index after surface |
|---|---|---|---|
| Cornea | 0 | 7.7 | 1.376 |
| | 0.5 | 6.8 | 1.336 |
| Eyelens | 3.2 | 5.33 | 1.385 |
| | 3.8 | 2.65 | 1.406 |
| | 6.6 | -2.65 | 1.385 |
| | 7.2 | -5.33 | 1.336 |
| Retina | 24.0 | -11.5 | |

**Table 1. Path of a light ray using the Gullstrand's eye model.**

### 4.2. Camera, Monitor and Eye Setup

The optical center of the camera is assumed to be the center of coordinates. The $x$ coordinate points to the right, the $y$ up, and the viewing angle is $z$ (a camera centered at left-handed coordinate system). The camera's vertical field of view is $3.5^o$. The eye position is at $P_0$=(0, 270, 600), and the screen area is defined by the rectangle (about 18" in diameter) with lower left coordinate (-183, 0, 0) upper right coordinate (183, 274, 0). All coordinates are in millimeters.

Nine synthetic images of the eye at $P_0$ looking at the center points of a $3\times3$ grid on the monitor were generated for calibration. Each grid was further divided into another $3\times3$ grid totalling 81 images for that position, that are used for testing the calibration on other screen points, other those 9 used for calibration.

## 5. Experimental Results

The eye was positioned at thirteen different points and 81 images were generated at each of these points. For each eye position, the eye is oriented to look at the 81 different monitor locations. This will allows us to observe how the accuracy of the calibration function decays with eye (or head) motion. So the eye position is move in the $x$, $y$ and

$z$ axis, and a total of 1134 ray traced images were generated for this experiment. For each position the eye gaze was estimated and compared to the real point of regard.

Figure 4 shows the computed errors along the screen for the eye positioned at $P_0$, the average error is 8.07 mm for this position, or about $0.8^o$.

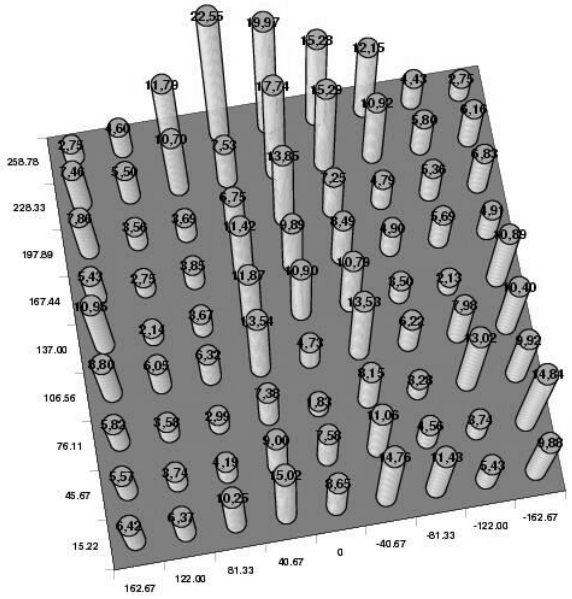



**Figure 4. Errors (in mm) along the screen for eye at $P_0$.**

Translating the eye along the $x$ axis resulted in small variations of the average errors. Figure 5 compares the average errors at each of the nine screen grid coordinates for the eye at $P_0$ and the eye positioned at (-100, 270, 600). Note that when the eye is moved to a new position, the camera direction was changed to keep the eye in its view, but the same calibration parameters are used. The average error is 9.92 mm for this second point.

Figure 6 shows the comparison for the eye at $P_0$ and at (0, 270, 700), a translation in $z$. The average errors increases to 40.56 mm in this position, showing that this calibration function is more sensitive to eye movements along the $z$ axis (scale).

The average error for the eye positioned at (0, 170, 600) is 21.76 mm, what shows that the technique is not very robust for movements along the $y$ axis as it is for lateral movement. From these experiments it is clear that this model is less robust for movements along the $z$ axis than it is for the others.

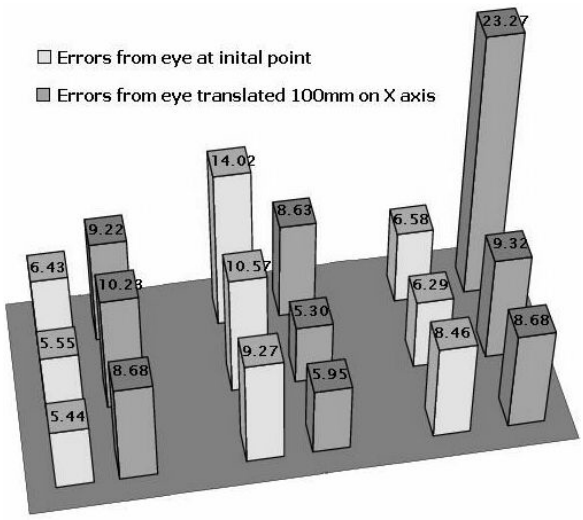



**Figure 5. Average screen grid errors (in mm) to eye at $P_0$ and at (-100, 270, 600).**

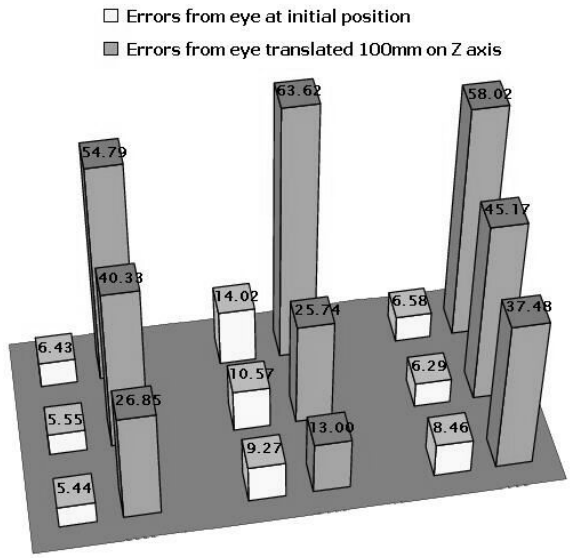



**Figure 6. Average screen grid errors (in mm) to eye at $P_0$ and at (0, 270, 700).**

## 6. Conclusion

We have introduced a computer vision framework for remote eye gaze tracking (EGT) that can serve as a testbed for the development and performance evaluation of new EGT techniques. The framework was successfully applied to tuning our current EGT system, and also used to evaluate the accuracy of the calibration function. Based on the experimental results obtained using synthetic images generated by ray tracing and using a physical model of the eye, we observed that the simple second order polynomial mapping function is enough to generate a better than $1^o$ accuracy. Future work includes testing and comparing other functions for the mapping between the pupil-glint vectors and screen coordinates, and develop and test new models that can eliminate the requirement of keeping the head still during the operation of the system.

## Acknowledgements

We thank the Fundação ao Amparo à Pesquisa do Estado de São Paulo (FAPESP), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the Conselho Nacional de Pesquisa e Desenvolvimento (CNPq) for their financial support.

## References

[1] S. Bajula and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102, Carnegie Mellon University, 1994.

[2] E. Crowe and N. Narayanan. Comparing interfaces based on what users watch and do. In *Proceedings of the Eye Tracking Research & Applications Sysmposium*, pages 29–36, 2000.

[3] J. Gips and P. Olivieri. EagleEyes: An eye control system for persons with disabilities. In *presentation at The Eleventh International Conference on Technology and Persons with Disabilities*, Los Angeles, California, March 1996.

[4] A. Glenstrup and T. Angell-Nielsen. Eye controlled media: Present and future state. Thesis (BS), Laboratory of Psychology, University of Copenhagen, June 1995.

[5] L. T. Inc. website: http://www.ltinc.com/doc.eds.htm#price, 2003.

[6] H. Istance and P. Howarth. Keeping an eye on your interface: The potential for eye-based control of graphical user interfaces (GUI's). In *Proceedings of HCI'94*. Cambrifge University Press, August 1994.

[7] H. Istance, C. Spinner, and Howarth. Providing motor-impaired users with access to standad graphical user interface (GUI) software via eye-based interaction. In *Proceedings of the ECDVRAT: 1st European Conference on Disability, Virtual Reality and Associated Technologies*, 1996.

[8] R. Jacob. *Virtual Environments and Advanced Interface Design*, chapter Eye Tracking in Advanced interface Design, pages 289–301. Oxford University Press, New York, 1995.

[9] R. Jacob and L. Sibert. Evaluation of eye gaze interaction. In *Proc. ACM CHI 2000 Human Factors in Computer Systems Conference*, pages 281–288. Addison-Wesley/ACM Press, 2000.

[10] K. Kim and R. Ramakrishna. Vision-based eye-gaze tracking for human computer interface. In *IEEE International Conf. on Systems, Man and Cybernetics*, Tokyo, Japan, October 1999.

[11] R. S. Longhurst. *Geometrical and Physical Optics*. Longmans, 2nd edition, 1967.

[12] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Computing, special issue on Advances in Facial Image Analysis and Recognition Technology IVC(18)*, (4):331–335, March 2000.

[13] R. Rao, G. Zelinsky, M. Hayhoe, and D. Ballard. Eye movements in visual cognition: A computational study. Technical report, National Resource Laboratory for the Study of Brain and Behavior, University of Rochester, March 1997.

[14] T. Schnell. Applying eye tracking as an alternative approach for activation of controls and functions in aircraft. Technical report, Iowa Space Grant Consortium's, 2000.

[15] B. Shaviv. The design and improvement of an eye-controlled interface. Technical report, Visualization Laboratory of the Department od Computer Science at the State University of New York at Stony Brook, 1993.

[16] B. Shaviv, A. Kaufman, and A. Bandopadhay. An eye tracking computer user interface. In *Proceedings IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, page 120, Los Alamitos, CA, USA, 1993. IEEE Comput. Soc. Press.

[17] R. Stiefelhagen and J. Yang. Gaze tracking for multimodal human-computer interaction. In *Proceedings of the International Conf. on Acoustics, Speech and Signal Processing*, Munich, Germany, April 1997.

[18] C. Ware and H. Mikaelian. An evaluation of an eye tracker as a device for computer input. In *Proceedings of SIGCHI+GI'87, Human Factors in Computing Systems*, 1987.

[19] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley-Interscience, 2nd edition, 1982.

[20] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (MAGIC) pointing. In *Proc. CHI'99*, pages 246–253, Pittsburgh, PA, USA, May 1999. ACM, Addison-Wesley.