Prototype of a vision-based gaze-driven man-machine interface

C. Colombo S. Andronico P. Dario

ARTS Lab*
Scuola Superiore Sant'Anna
Via Carducci, 40
56127 Pisa, Italy

E-mail: [Columbus, Susy, Dario]@arts.sssup.it

Abstract

This paper describes preliminary work on a non-intrusive gaze-driven interface for man-machine interaction based on vision techniques. The current computer screen location pointed at by the user is evaluated using a linear model of image-to-screen mapping, which can be easily auto-calibrated at run-time on the basis of the current relative position of camera and user and camera settings. A simple active deformable model of the eye is defined, which is used both to track user's movements and estimate the current position of the user's pupil in the image in a decoupled fashion. Experiments show that the proposed approach is fast and accurate enough for the design of low-cost man-machine interfaces, with applications ranging from the assistance to the disabled people to multimedia systems.

1 Introduction

In the last few years, the simultaneous increase of computing power and decrease of hardware costs has allowed the design of more and more complex and flexible interfaces for man-machine interaction. A recent trend in the design of man-machine interfaces (MMI) is the development of *user friendly* interfaces, providing computers with means of communication—such as speech, hearing, touch, vision—which are peculiar to humans, and let the machine be taylored to specific user's needs—e.g. to exploit at best the residual mobility of a disabled user.

Gaze-driven interfaces are based on the principle of "what you look at is what you get," that is, specific commands are executed on the basis of the current user's gaze direction [1, 2]. The application field of such interfaces is remarkably wide, ranging from applications which require a high-precision—registration of eye movements in psychology and for medical diagnoses, eye-driven missile targeting [3, 4]—to general purpose applications—office automation, multimedia systems, market analysis, virtual reality, games [5–7]. Several techniques have been proposed in the past for measuring the direction of gaze. These include evaluating the relative position between pupil's center and reflection from the cornea by means of special contact lenses [1], infrared sensors [8] or data-helmets [3], and placing small electrodes on the skin around the eye—the so called electro-oculographic method [9]. Although often accurate, such techniques have some limitations for generale-purpose applications, which arise from the high cost of the technologies embedded, from being intrusive and from constraining usually the natural mobility of the user [8].

Recently, the use of computer vision techniques[1] has been proposed for MMI applications [10, 11]. This is no doubt a step toward the development of cheap and friendly interfaces for man-machine communication. Being based on a passive sensor (a TV camera), computer vision is intrinsically non intrusive, and does not require an expensive equipment. Among the most recent vision techniques for the recognition and/or tracking of face characteristics and expressions, those based on *active deformable templates* are particularly promising for MMI development [12]. Yet, implementations with such techniques presented so far are either not explicitly designed for gaze-tracking [13] or too slow for real-time MMI applications [14, 15].

In this paper, we describe the prototype of a friendly gaze-driven man-machine interface embedding vision techniques. Thanks to an extremely simple deformable template for eye-iris, eye tracking and pupil localization in the image is addressed in a decoupled fashion at an high cycle rate. Gaze-point localization on the computer screen is based on a linear model of image-to-screen mapping, which is both easy to calibrate and update at run-time on the basis of the current position of camera and user. Preliminary tests with allowed translational user motion show that the approach is suitable for the development of simple interfaces for disabled people and multimedia applications. Active movements of the camera and run-time

*Advanced Robotics Technologies and Systems Laboratory.

[1] *Computer vision* (image analysis) can be considered as the dual of *computer graphics* (image synthesis). Graphics plays a crucial role in MMI development.

0-8186-7108-4/95 $4.00 © 1995 IEEE

Authorized licensed use limited to: University of Pretoria. Downloaded on July 24,2023 at 18:41:57 UTC from IEEE Xplore. Restrictions apply.

adjustment of camera settings can be used in order to extend the proposed framework and improve system performance.

The paper is organized as follows. In sect. 2, we illustrate the interface both in the general aspects and details. Then, in sect. 3 we discuss experimental results, and finally draw the conclusions and outline directions for future research in sect. 4.

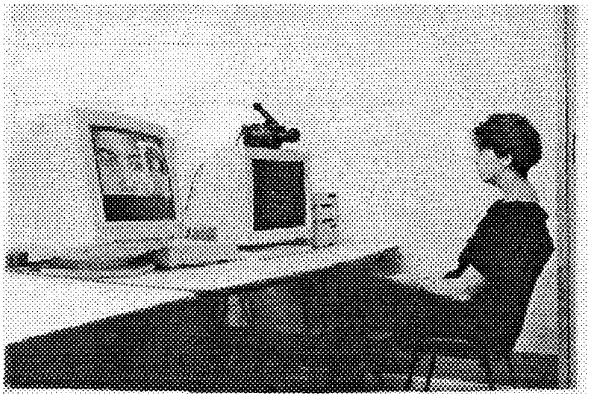

Figure 1: System setup for the gaze-driven interface. Notice the absence of standard interaction devices such as keyboard and mouse.

2 The gaze-driven interface

Fig. 1 shows the setup of the gaze-driven interface. The computer screen is divided into a number of windows (the figure shows a nine-window interface), each corresponding to a specific action selectable by the user. A camera is placed on top of the screen, and the screen location currently pointed at by the user is monitored by evaluating the current position of one of his pupils in the image (in the figure, a second screen shows the image plane's content). The user's eye works as a mouse, and an eye-driven command, the equivalent of a mouse click, is issued on a given window as the persistence of gaze on it exceeds a given threshold[2].

The system works properly then, if the following two conditions are met:

- the mapping between image and screen coordinates is well approximated, and
- the current pupil's position in the image is accurately estimated.

Fulfilling these conditions is not a trivial task, basically for two reasons. First, as the user's head is allowed to move freely, both the image-to-screen mapping and the image of the eye are time-varying. Second, due to the unpredictable nature of user's saccadic eye movements, it is not possible to use standard tracking methods to estimate directly the pupil's position in the image. However, head motions being usually smooth, it is possible to track the image of eye's external contour, which is fixed to the head.

We propose then a *decoupled approach* to ensure the two conditions above:

1. *track at each instant the current position of the external eye contour.* Tracking provides both the necessary information to update properly the image-to-screen mapping, and a *search bound* for the pupil's position in the image;
2. *use a suitable search algorithm to find the pupil's position inside the search bound.*

In subsect. 2.1 a technique for decoupling eye tracking and pupil search in the image is presented. Then, a method for approximating the image-to-screen mapping and detecting gaze direction is proposed in subsect. 2.2.

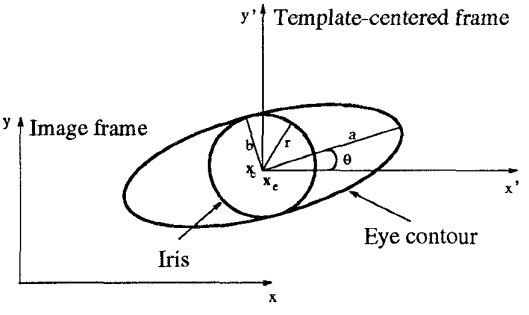

Figure 2: The deformable eye-iris template. The ellipse and circle centers have been taken coincident for clarity of presentation.

2.1 Image measurements

Image measurements for eye tracking and pupil recognition are obtained by coupling a model (template) of eye contour and iris with raw image data through a simple deformable template approach based on least squares fitting.

The template, whose geometry is shown in Fig. 2, assumes an elliptical shape for the eye contour and a circular shape for the iris—the pupil's center coincides of course with the center of the iris. The ellipse has a center $\boldsymbol{x}_e = (x_e, y_e)^T$, a major axis $2a$ and a minor axis $2b$ and an orientation ϑ. The iris has a radius r, and a center $\boldsymbol{x}_c$ which is assumed to lay always inside the ellipse. The eye-iris template is thus specified by $8 = 5 + 3$ independent parameters. Notice that an elliptical eye contour model may appear too simple to be used. Nevertheless, it "captures" important eye features such as eye position, orientation and dimensions encoding them in a small set of parameters, and this suggests its use in many practical applications in which speed is more important than accuracy.

[2] The problem of estimating an appropriate threshold has been referred to in [1] as *the Mida's touch* problem: the response time of the interface has to be the smallest possible compatibly with the avoidance of "false alarms."

Authorized licensed use limited to: University of Pretoria. Downloaded on July 24,2023 at 18:41:57 UTC from IEEE Xplore. Restrictions apply.

The template shape is roughly initialized in the image by means of an eye identification technique—[16], but other techniques may work equally well, see e.g. [17]—, and its parameters are then refined and updated at run-time by means of decoupled template-to-image fitting. That is, at each time the current ellipse position in the image is estimated and filtered, thus providing the suitable boundings for the subsequent search of the current iris (and pupil) position. Notice that the decoupled search algorithm reduces the complexity of template-image interaction, in that the fitting of the overall 8-parameter template is obtained by the subsequent fitting of a 5-parameter sub-template and a 3-parameter sub-template.

2.1.1 Eye contour tracking

The fitting of the elliptical template to the current eye contour in the image, is carried out in two phases. In a first phase, a *local search* of new edge points (brightness gradient maxima) takes place in an image neighborhood of the previous template instance. The search is performed at N regularly sampled template points along normal directions to the ellipse, and a set $\{\boldsymbol{x}_g^i = (x_g^i, y_g^i)^{\mathrm{T}},\ i = 1 \ldots N\}$ of edge points is computed by means of a recursive coarse-to-fine algorithm based on finite differences [18]. In the second phase, the new template parameters are extracted by fitting edge points with a generic conic via *least squares*, and then imposing the conic to be an ellipse. That is, we first find the generic conic parameters $\alpha, \beta, \gamma, \delta, \varepsilon$ which minimize the squared error:

$$\sum_{i=1}^{N} \left({x_g^i}^2 + \alpha x_g^i y_g^i + \beta {y_g^i}^2 + \gamma x_g^i + \delta y_g^i + \varepsilon \right)^2, \quad (1)$$

and then express them in terms of ellipse centre, major and minor axes, and orientation:

$$x_e = \frac{2\beta\gamma - \alpha\delta}{\alpha^2 - 4\beta} \quad (2)$$

$$y_e = \frac{2\delta - \alpha\gamma}{\alpha^2 - 4\beta} \quad (3)$$

$$a = \left[\frac{-(\gamma x_e + \delta y_e + 2\varepsilon)}{1 + \beta - \sqrt{(1-\beta)^2 + \alpha^2}} \right]^{1/2} \quad (4)$$

$$b = \left[\frac{-(\gamma x_e + \delta y_e + 2\varepsilon)}{1 + \beta + \sqrt{(1-\beta)^2 + \alpha^2}} \right]^{1/2} \quad (5)$$

$$\tan 2\vartheta = \frac{-\alpha}{\beta - 1}. \quad (6)$$

A smooth tracking of eye contour is obtained by suitably filtering the current estimates of the ellipse parameters using a temporal recursive *mobile mean filter*, which weights each new measurement according to a reliability measure $k_{mm} \in [0, 1]$. For instance, the current major axis of the ellipse is evaluated as:

$$a^{new} := k_{mm} a^{new} + (1 - k_{mm}) a^{old}. \quad (7)$$

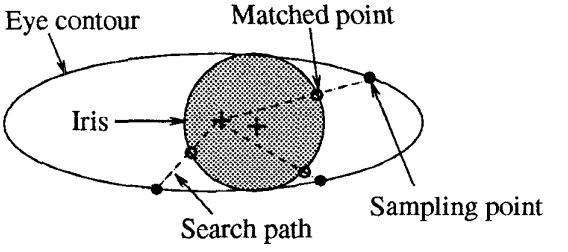

Figure 3: Searching the iris template in the image.

2.1.2 Pupil search

Once that the new ellipse template is found, the current iris position is searched for inside it according to a search-and-fit technique similar to that used for the ellipse. As shown in Fig. 3, the edge search lines are radial paths starting from the center of the new ellipse. Notice that, in order to avoid erroneous results due to possible *partial iris occlusions*, the edge points belonging to the ellipse are automatically excluded from least squares fitting with the circular template. As mentioned above, due to saccadic iris motions no filtering can take place for iris parameters. Therefore, the center of the iris is simply assumed as the new location of the pupil's center, and used for estimating the current direction of gaze.

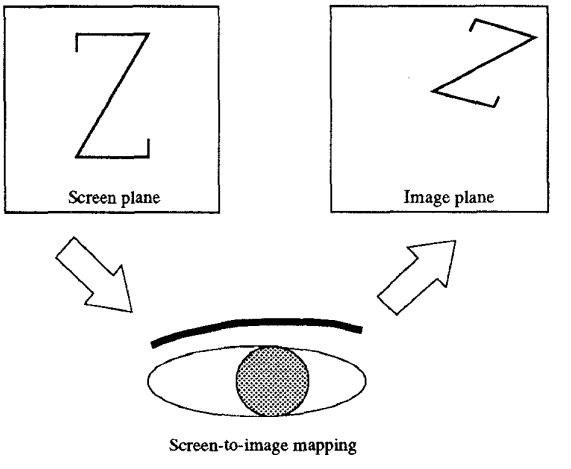

Figure 4: Linear relationship between screen and image shape.

2.2 Estimating gaze direction

The user's direction of gaze, that is the computer screen location $\boldsymbol{x}_s$ currently pointed by the user, is put in a one-to-one correspondence with the estimated image location $\widehat{\boldsymbol{x}}_p$ of the pupil's center—obtained as above—by means of a simple *linear model of image-to-screen mapping*:

$$\boldsymbol{x}_s = \mathcal{A}\, \boldsymbol{x}_p + \boldsymbol{t}, \quad (8)$$

Authorized licensed use limited to: University of Pretoria. Downloaded on July 24,2023 at 18:41:57 UTC from IEEE Xplore. Restrictions apply.

where t is a rigid coordinate translation and $\mathcal{A}$ accounts for affine image shape transformations: rigid rotation, dilation (isotropic scaling), and shear (anisotropic scaling)—see Fig. 4. The affine model $\{\mathcal{A}, t\}$ is estimated at system startup by means of a "calibration" procedure, and then updated, or "auto-calibrated," at run-time on the basis of current visual data and camera parameters.

For affine model calibration, a set of M image observations $\{\widehat{\boldsymbol{x}}_p^j,\ j = 1 \ldots M\}$ is collected by letting the user execute a *scanpath*—i.e. a sequence of gaze shifts—onto a number of desired locations $\{\widetilde{\boldsymbol{x}}_s^j\}$ on the screen, and by estimating and recording the corresponding image pupil positions. The affine model is then obtained using data and observations, as a least squares solution to an overdetermined system obtained from eq. (8). The least squares computation can be carried out in two steps, by first estimating the translation t simply as the centroid of the observations, and then using the estimated translation to solve for $\mathcal{A}$. Notice that during the calibration phase, head motions are not allowed, in principle, as they determine a calibration error which depends on the extent of user displacements. Inaccuracy in estimating image pupil positions is another important source of error. Calibration errors can be reduced by suitably choosing the distribution and the number of observation points, and zooming with the camera on the user's eye once it has been approximately located in the image—refer again to Fig. 1.

Auto-calibration at run-time is easily accomplished in the case of user's head translations parallel to the camera plane. Since these affect only the translation component t of the mapping, auto-calibration is obtained in this case by updating t on the basis of the estimated displacement of the ellipse's centroid between two consecutive images. The case above is met very often in practice, if the camera lens is approximately parallel to the screen. In fact, when using a computer, the user has a natural tendency to keep his face parallel to the screen. The problem of auto-calibration in the general case of t *and* $\mathcal{A}$ both time-varying can be addressed by studying how the affine mapping model is affected by changes of camera parameters and relative head-camera motions, as shown in [19].

3 Preliminary results

In this section, we present experimental results obtained with an off-the-shelf TV camera and frame grabber, and the 768 × 768 pixels square XWindows interface of Fig. 1. Such an interface runs on a SUN SPARCStation 2 and is composed of nine identical windows 192 × 192 pixels large, each separated from the other by screen stripes 96 pixels wide. The stripes correspond to regions of the screen in which no action takes place, and are introduced to reduce the risk of "false alarms" in the case of slightly incorrect pointing.

Experiments of pupil location and eye contour tracking have been performed on an image sequence in which the user's head moved vertically at approximately 3 pixels/frame in the image and least squares fitting was carried out using $N = 64$ points. Fig. 5 shows two non consecutive frames of the sequence, explicitly the first frame after eye recognition and image zooming (*left*) and a frame taken some instants later (*right*). Notice from a comparison of the two frames that eye tracking exhibits a lag—the template does not change much its shape, position and orientation—in the initial phase, due to a heavy filtering ($k_{mm} = 0.01$) of ellipse parameters. Nevertheless, the pupil location in the image is computed quite correctly, with an average error of 3 pixels with respect to the "ground truth." Moreover, the overall cycle rate is quite high—in the order of a few milliseconds.

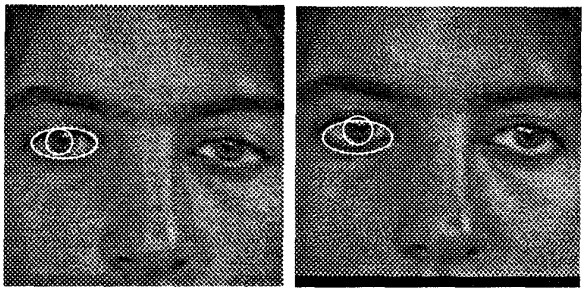

Figure 5: Results for pupil localization and eye tracking (see text). The black stripe at the bottom of the right image gives the amount of user's vertical translation with respect to the left frame.

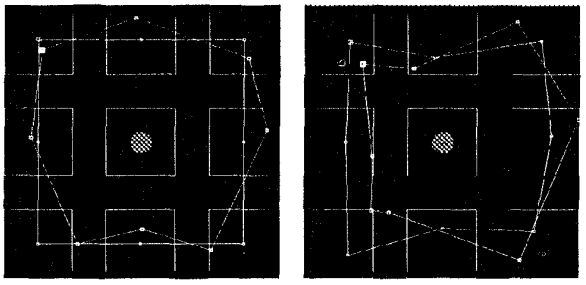

Figure 6: Results of calibration. Results of 9 out of 36 test points are shown. For ease of interpretation, both data and observations are connected through polygonal lines. *Left*: Test on the calibration point set. *Right*: Test on another point set.

Tests of gaze direction estimation were carried out using as ground truth both the set of $M = 36$ uniformly sampled screen locations used for calibration, and another set of M locations chosen at random in the screen. The user was asked to successively point at these locations ($\{\boldsymbol{x}_t^k,\ k = 1 \ldots M\}$), while the corresponding computed pupil positions in the image, $\{\widehat{\boldsymbol{x}}_p^k\}$, were remapped onto the screen as $\{\widehat{\boldsymbol{x}}_s^k\}$ according to the linear model of eq. (8). The mean error $\mu = (1/M)\sum_{k=1}^{M} \|\widehat{\boldsymbol{x}}_s^k - \boldsymbol{x}_t^k\|$ was of 86.2 pixels (that

Authorized licensed use limited to: University of Pretoria. Downloaded on July 24,2023 at 18:41:57 UTC from IEEE Xplore. Restrictions apply.

makes approximately 3.25 cm, given a monitor resolution of 0.38 mm/pixel) in the case of the test on calibration data (Fig. 6, *left*), and increased to 96.3 pixels (3.63 cm) in the test with randomly chosen screen points (Fig. 6, *right*). Although the error is relatively high, due to both inaccuracies in the previous phase of pupil localization and to slight head movements during calibration, the commands to the interface were correctly interpreted by the system, except when user's fixation was computed in a no-action stripe[3], in which case no command was issued.

4 Conclusions and Future Work

In this paper we have described preliminary work on a friendly (non intrusive, user translational motions allowed) vision-based gaze-driven MMI. We have shown how to compute the gaze direction by means of a linear model of image-to-screen mapping, and defined a simple deformable contour model of eye-iris for decoupling eye contour tracking and estimation of pupil's position in the image. Experiments show that the proposed method is accurate enough for the design of low-cost man-machine interfaces, with applications ranging from the assistance to the disabled to multimedia systems.

Much work has still to be done, both in the theoretical developments and in the experimental aspects, towards the specification of a flexible general-purpose MMI based on the proposed approach. As an example, the system could be made more precise and robust by adapting at run-time camera parameters (focus, aperture, etc.) to current visual requirements. Flexibility could be added to the system also by using an active camera head instead of a fixed camera, thus allowing to track the user in a wider workspace and even to shift among different users.

Acknowledgements

The authors would like to thank Francesca Di Donato for lending her face for the experimental tests. This work has been funded, in part, by Centro Protesi INAIL of Budrio (BO), Italy. Carlo Colombo was a visiting scientist at LIFIA-IMAG Grenoble, France, supported by a fellowship from the EC Human Capital and Mobility network SMART.

References

1. R.J.K. Jacob. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings ACM CHI'90 Human Factors in Computing Systems Conference, 1990*, pages 11–18, 1990.
2. R. Razdan. Eye tracking for man/machine interfaces. *Sensors*, pages 39–42, 1988.
3. SMI: SensoMotoric Instruments GmbH, Potsdamer Str. 18a, D-14513 Teltow, Germany. *ISCAN: Eye Movement Monitoring Systems*, August 1993.
4. E. Oshuga, H. Tsuji, and H. Ide. Eyeball movement analysis and communication system for the blind. *Journal of Robotics and Mechatronics*, 4(4):354–356, 1992.
5. I. Starker and R.A. Bolt. A gaze-responsive self-disclosing display. In *Proceedings ACM CHI'90 Human Factors in Computing Systems Conference, 1990*, pages 3–9, 1990.
6. S.K. Singh, S.D. Pieper, D.O. Popa, and J. Guinness. Control and coordination of head, eyes and facial expressions of virtual actors in virtual environments. In *Proceedings of the 2nd IEEE International Workshop on Robot and Human Communication (RO-MAN'93), Tokyo, Japan, 1993*, pages 335–339, 1993.
7. J.C. Schryver and J.H. Goldberg. Eye-gaze and intent: Application in 3D interface control. In *Proceedings of the 4th International Conference on Human-Computer Interaction, Japan, 1992*, pages 573–578, 1992.
8. D. Cleveland and N. Cleveland. Eyegaze eyetracking system. In *IMAGINA 92*, pages 17–23, Monte Carlo, January 1992.
9. J. Gips, P. Olivieri, and J. Tecce. Direct control of the computer through electrodes placed around the eyes. In *Proceedings of the 5th International Conference on Human-Computer Interaction, Japan, 1993*, pages 630–635, 1993.
10. S. Baluja and D.A. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, chapter 6. Morgan Kaufmann, 1994.
11. A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):602–605, 1993.
12. A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, 1992.
13. C.L. Huang and C.W. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12):1435–1444, 1992.
14. A. Yuille and P. Hallinan. Deformable templates. In A. Blake and A. Yuille, editors, *Active Vision*, chapter 2, pages 21–38. MIT Press, 1992.
15. A.L. Yuille, P.W. Hallinan, and D.S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.
16. R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
17. I. Craw, D. Tock, and A. Bennett. Finding face features. In *Proceedings of the 2nd European Conference on Computer Vision, S. Margherita Ligure, Italy, 1992*, pages 92–96, 1992.
18. R. Curwen and A. Blake. Dynamic contours: Real-time active splines. In A. Blake and A. Yuille, editors, *Active Vision*, chapter 3, pages 39–57. MIT Press, 1992.
19. C. Colombo and P. Dario. Visual understanding of gaze pointing actions. In *Proceedings of the 3rd International Symposium on Intelligent Robotic Systems SIRS'95, Pisa, Italy, July 1995*, 1995.

[3] The mean error performance itself can be used in order to suitably size up the width of the stripes.

Authorized licensed use limited to: University of Pretoria. Downloaded on July 24,2023 at 18:41:57 UTC from IEEE Xplore. Restrictions apply.