ELSEVIER

Pattern Recognition Letters 20 (1999) 721–730

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Real-time head tracking from the deformation of eye contours using a piecewise affine camera

Carlo Colombo [a,*], Alberto Del Bimbo [a]

[a] *Dipartimento di Sistemi e Informatica, Università di Firenze, Via Santa Marta 3, I-50139 Firenze, Italy*

Received 9 June 1998; received in revised form 6 November 1998

Abstract

A computer vision based approach for human–computer interaction through head movements is presented and evaluated in a non-immersive virtual reality context. Once intercepted and tracked in real-time using a piecewise affine camera model and affine-deformable eye contours, user head displacements are estimated and remapped onto the tridimensional graphic environment according to a natural interface metaphor. Both the real-time performance of the tracker and the improved head parameter estimation accuracy – as compared to the one obtainable using globally affine camera models – encourage the use of this approach to support diverse advanced interaction scenarios and applications. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Computer vision; 3D Human–computer interaction; Contour tracking; Pose estimation; Affine camera models

1. Introduction

In the last two decades, human–computer interfaces have gradually evolved to provide systems with natural interaction modalities inspired from human behavior in the real world. The recent advent of advanced interface paradigms and environments such as virtual and augmented reality has been made possible thanks to new interaction devices, such as head-mounted displays, datagloves, etc., capable of supporting more and more sophisticated dialogue modalities and user needs [surveys can be found e.g. in (Nielsen, 1993; Myers, 1998)]. Despite the recent research and technology efforts, advanced human–computer interaction devices still suffer of a number of problems which, if not solved, are likely to hamper the widespread diffusion of the next generation computer interfaces. Two main problems with current devices are *intrusiveness* (e.g. wearing a virtual reality helmet is not exactly fun) and *expensiveness* (related to the need of special hardware and high computational overload).

Being an intrinsically non-intrusive technology, computer vision is becoming more and more appealing as a human–machine interaction technology [useful reviews can be found in (Pentland, 1996; Crowley, 1997)]. Indeed, in the last few years, a number of computer vision algorithms and techniques have appeared for localizing, tracking and recognizing user body parts such as head, arms, hands and facial features such as lips and eyes even with an inexpensive camera – a

* Corresponding author. Tel.: +39 055 479 6540; fax: +39 055 479 6363; e-mail: columbus@dsi.unifi.it

0167-8655/99/$ – see front matter
PII: S0167-8655(99)00036-7

device which will no doubt be part of the standard equipment of any PC with multimedia facilities in the near future. Computer vision contributions to human–computer interaction are twofold. On the side of *interaction semantics*, recognition of human gestures (see e.g. (Pavlovic et al., 1997)) and expressions (as in (Essa and Pentland, 1997)) can be used to develop natural human–machine interaction languages, while face recognition (for a recent contribution, see (Lam and Yan, 1998)) can be effective for person authentication and surveillance purposes. On the side of *interaction geometry*, localization and tracking of body features can be used to develop special pointers to be used in the place of 3D mice, joysticks, etc. Azarbayejani et al. (1993) proposed a Kalman filter based head tracking technique for virtual holography and teleconferencing; Cipolla and Hollinghurst (1996) developed a system for pointing in a tridimensional (3D) robot workspace using affine stereo vision and hand tracking; in (Colombo and Del Bimbo, 1997), a technique was presented to infer the computer screen location currently observed by the user by the measurement of image eye pupil displacements.

While several computer vision techniques have been proposed with a possible application to human–computer interaction, actually only a few of these techniques were integrated into a complete system (for example, in (Gee and Cipolla, 1996), an effective method is proposed to determine head orientation in real-time with an uncalibrated camera, but no application exploiting this method is presented). In other cases, the computer vision techniques, although often remarkably accurate, are simply too slow for a satisfactory interaction: the overall system would simply have no time but for the required vision computations (this happens e.g. with the eye localization method presented by Yuille and Hallinan (1992) when one attempts to use it for real-time eye tracking with a PC). In this respect, computer vision techniques specifically developed for hard real-time robotic applications are certainly adequate for interaction applications. Such techniques often rely on simplified linear models of camera–world interaction in order to dramatically reduce the burden of visual computations (see e.g. the recent works of (Cipolla and Hollinghurst, 1997) and (Allotta and Colombo, 1999).

In this paper, we present a computer vision based approach to interact with 3D graphic environments through head displacements. The modality of interaction can be effective for both disabled users affected by severe limb motor pathologies and general users requiring to communicate in a non-intrusive way with the computer. The operating context is composed by a camera placed in front of the user and an on-screen environment featuring a 3D realistic graphic scene. A piecewise affine camera model is introduced, allowing the user's head displacements to be estimated in real-time from the comparative analysis of the affine deformations of the left and right eye contours in the image. The estimated head parameters are mapped onto commands for 3D display, and rendered via graphical synthesis (remapping) according to a drag and click interface metaphor. Although integrated in a non-immersive virtual reality context, the proposed framework can be easily adapted to other tasks and scenarios, so as to support interaction in multimedia, videoconferencing, telepresence, usability monitoring, augmented reality, interactive 3D video and similar environments.

2. Models

Relative geometry and image projection. Let α, β and γ denote respectively the local coordinate frames associated to screen, user's head and camera. Head movements in space can be described by six independent degrees of freedom (DOF), three for position and three for orientation, measured in some fixed reference frame ρ. The head DOFs are encoded in the linear transformation between the coordinate representations of a generic point in space $\boldsymbol{p}$ in the β-frame (${}^{\beta}\boldsymbol{p}$) and in the ρ-frame (${}^{\rho}\boldsymbol{p}$). Using homogeneous coordinates for $\boldsymbol{p}$, such a transformation can be compactly described by a 4×4 matrix ${}^{\rho}_{\beta}\mathrm{T}$ s.t. $[{}^{\rho}\boldsymbol{p}^{\mathrm{T}}1]^{\mathrm{T}} = {}^{\rho}_{\beta}\mathrm{T}[{}^{\beta}\boldsymbol{p}^{\mathrm{T}}1]^{\mathrm{T}}$. As changes of frame are composed linearly, the $\beta \mapsto \rho$ transformation can be reconstructed from the composition of the $\beta \mapsto \gamma$ and $\gamma \mapsto \rho$ transformations, as ${}^{\rho}_{\beta}\mathrm{T} = {}^{\rho}_{\gamma}\mathrm{T}\,{}^{\gamma}_{\beta}\mathrm{T}$. The $\gamma \mapsto \rho$ transformation is

time-independent, at least as long as the camera is fixed, while the $\beta \mapsto \gamma$ transformation depends on the current position of the user's head.

The six DOF describing user–camera relative geometry are expressed by a translation 3-vector ${}^{\gamma}_{\beta}\boldsymbol{t}$ and a 3×3 rotation matrix ${}^{\gamma}_{\beta}\mathrm{R}(\tau, \sigma, \phi)$, which is completely defined by the three angles τ (tilt), σ (slant) and ϕ (orientation). The slant $\sigma \in [0, \frac{\pi}{2}]$ is the angle between the face and the image planes; this vanishes identically if the two planes are parallel. The tilt angle $\tau \in [0, 2\pi]$ gives instead the image direction of maximum depth decrease (see Fig. 1). Perspective projection of a face point ${}^{\gamma}\boldsymbol{p} = [{}^{\gamma}X \quad {}^{\gamma}Y \quad {}^{\gamma}Z]^{\mathrm{T}}$ onto the camera plane point $\boldsymbol{x} = [x \quad y]^{\mathrm{T}}$ is given by $\boldsymbol{x} = \frac{\lambda}{{}^{\gamma}Z}[{}^{\gamma}X \quad {}^{\gamma}Y]^{\mathrm{T}}$, where λ denotes the focal length of the camera.

Perspective projection can be approximated by an affine map provided that the depth variations for an object of interest in the scene are much smaller than the average depth (see e.g. (Mundy and Zisserman, 1992)). In our specific scenario, we observe that the depth of the points of *a single eye* is virtually constant, and equal to eye centroid depth ${}^{\gamma}Z_{\mathrm{E}}$. Hence, we get $\boldsymbol{x} = \mathrm{H}_{\mathrm{E}}[{}^{\gamma}X {}^{\gamma}Y] + \boldsymbol{h}_{\mathrm{E}}$, where ($c_\alpha \doteq \cos\alpha$, $s_\alpha \doteq \sin\alpha$)

$$\mathrm{H}_{\mathrm{E}} = \kappa_{\mathrm{E}} \begin{bmatrix} c_\tau c_\sigma c_\phi - s_\tau s_\phi & c_\tau c_\sigma s_\phi + s_\tau c_\phi \\ s_\tau c_\sigma c_\phi + c_\tau s_\phi & s_\tau c_\sigma s_\phi - c_\tau c_\phi \end{bmatrix}, \qquad \boldsymbol{h}_{\mathrm{E}} = \kappa_{\mathrm{E}} \begin{bmatrix} {}^{\gamma}X_{\mathrm{E}} \\ {}^{\gamma}Y_{\mathrm{E}} \end{bmatrix}, \tag{1}$$

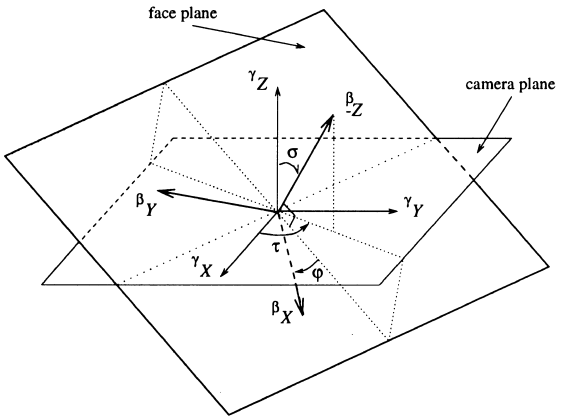

Fig. 1. Definition of head orientation parameters.

being $\kappa_{\mathrm{E}} = \frac{\lambda}{{}^{\gamma}Z_{\mathrm{E}}}$. Notice that linearizing perspective produces a pose ambiguity, i.e. there are two distinct object poses sharing the same visual appearance. In our case, an ambiguity exists whenever $\sigma \neq 0$, as the angle triplets (τ, σ, ϕ) and $(\tau + \pi, \sigma, \phi + \pi)$ yield exactly the same H_{E}. The ambiguity problem must be solved in order to estimate head pose from a single affine projection map (see Appendix A).

In this work, we introduce a *piecewise affine camera model*, in which any view of user's face is not represented by a single affine map, but by two distinct affine maps, namely, $(\mathrm{H}_{\mathrm{L}}, \boldsymbol{h}_{\mathrm{L}})$ and $(\mathrm{H}_{\mathrm{R}}, \boldsymbol{h}_{\mathrm{R}})$, related respectively to the left and right eyes. According to such a model, the imaging projection can be approximated by an affine map for the two eyes taken individually, but not when they are considered together. As it will be clear in the following sections, such a model (1) allows to approximate perspective more accurately than the single affine map model, (2) still permits the use of fast affine template tracking techniques, and (3) does not require any disambiguation strategy to compute the pose of the face.

Head tracking. The user's eyes are good face features to track in the image for the recovery of head displacements from visual appearance changes. Indeed, the *external contour* of each eye, being fixed to the head, can be related to head displacements. Image measurements for head tracking are obtained by coupling an elastic template with raw image data. At startup, the template is automatically initialized in the image, in order to match the location and shape of the user's eyes. For each of the two eyes, a *reference template* is thus produced. The external eye reference template is made of two semi-ellipses sharing their major axis, depending on six parameters (e_1–e_6 in Fig. 2), namely the common major axis (e_1), the two minor

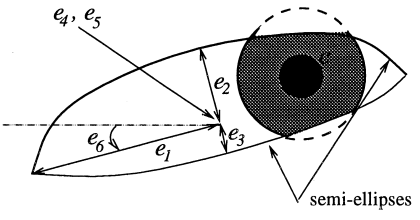

Fig. 2. Reference template for L and R eyes.

axes (e_2, e_3), the ocular center's image coordinates (e_4, e_5), and the common orientation (e_6). At run-time, a visual tracker is started, which keeps itself locked onto the current external eyes' visual appearance. The current tracker's shape and location provide the time-varying information needed to measure user action and use it for human–computer interaction.

Let us assume that the reference frame ρ is the β-frame at initialization time ($t = 0$), i.e. $\rho = \beta(0)$. Assume also that $\sigma(0) = \tau(0) = \phi(0) = 0$, i.e. the reference and camera planes are parallel, and have mutually parallel X-axes. Then, by Eq. (1), the reference template models a frontoparallel view of each eye, translated by $\boldsymbol{h}_{\mathrm{E}}(0)$, and undergoing scaling and specular reflection w.r.t. the face plane's content by $\mathrm{H}_{\mathrm{E}}(0) = \kappa_{\mathrm{E}}(0)\,\mathrm{diag}(1,-1)$. It is not difficult to show that the following affine relationship exists between a generic eye view $\boldsymbol{x}(t)$ and the reference eye view $\boldsymbol{x}(0)$: $\boldsymbol{x}(t) = \boldsymbol{x}_{\mathrm{E}}(t) + \mathrm{L}_{\mathrm{E}}(t)[\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{E}}(0)]$, where

$$\mathrm{L}_{\mathrm{E}}(t) = \mathrm{H}_{\mathrm{E}}(t)\mathrm{H}_{\mathrm{E}}^{-1}(0), \tag{2}$$

and where $\boldsymbol{x}_{\mathrm{E}}(t) \equiv \boldsymbol{h}_{\mathrm{E}}(t)$ is the projection of the ocular center ${}^{\gamma}\boldsymbol{p}_{\mathrm{E}}(t)$, which equals the centroid of the eye projection. Fixing $\beta_0 = \beta(0)$ as the reference frame allows us also to describe the relative position between the current frame $\beta = \beta(t)$ and the reference using the relative rotation/translation pair ${}^{\beta_0}_{\beta}\mathrm{R} = {}^{\beta_0}_{\gamma}\mathrm{R}\,{}^{\gamma}_{\beta}\mathrm{R}$ and ${}^{\beta_0}_{\beta}\boldsymbol{t} = {}^{\gamma}_{\beta_0}\mathrm{R}\left[{}^{\gamma}_{\beta}\boldsymbol{t} - {}^{\gamma}_{\beta_0}\boldsymbol{t}\right]$, where ${}^{\beta_0}_{\gamma}\mathrm{R} = {}^{\gamma}_{\beta_0}\mathrm{R} = \mathrm{diag}(1,-1,-1)$.

3. Measurements

Eye tracking in 2D. At system startup, a raw estimate of left and right eye location and shape in the image is derived, so as to initialize the reference templates (Fig. 3, left). To speed up processing, the two image regions containing the eyes are first roughly located by means of *edge dominance maps* (introduced by (Brunelli and Poggio, 1993)), i.e. maps which take into account the dominance of brightness gradient in a given direction. Once the regions including the eyes are found, the templates are adjusted against image data by an energy-minimization criterion similar to the one introduced in (Yuille and Hallinan, 1992). A quadratic energy term, function of the 6 eye template parameters, is minimized, so that the template is relaxed inside the eye regions by gradient descent. To the energy term contribute both peaks and valleys of image brightness (modeling respectively the sclera and the iris), and brightness discontinuities. After relaxation, each reference template is used to initialize the run-time tracker.

The tracker is a lightweight process allowing a fast eye tracking behavior. It is composed of a discrete set of points, initialized at startup by uniformly sampling the reference template's external eye ellipses (*reference tracker*) for later use. At run-time, the tracker's parameters are refined and updated by means of a simple tracker-to-image fitting approach, based on least squares and the extraction of brightness edge points. To avoid any false edge matches due to the presence of eye iris, an independent tracker is used that monitors the current iris position inside the eye contour (Fig. 3, middle).

Thanks to the affine projection model, each eye/iris tracker can be made quite robust by allowing it to deform only in an affine fashion, thus constraining the possible contour deformations to a six-dimensional space (Fig. 3, right). Tracking is

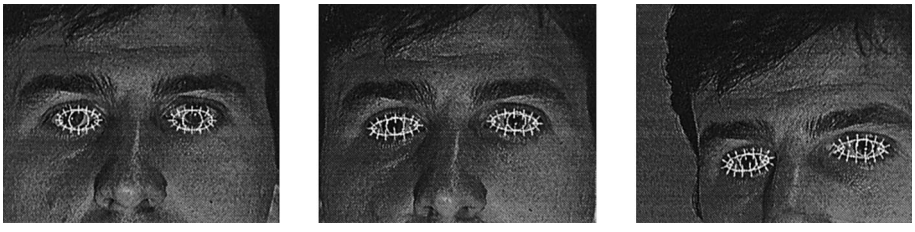

Fig. 3. Visual tracking phases. Left to right: raw tracker placement, tracker adjustment, tracker in action during head panning. Normal segments indicate the local search range for the tracker (see text).

equivalent to estimating a *dynamic visual state* composed, at a generic time $t \geqslant 0$, by the tracker points $\{\boldsymbol{x}_i(t)\}$, $i = 1, \ldots, n$, and their image velocities. Let the tracker's centroid be $\boldsymbol{x}_{\mathrm{E}}(t)$ and the reference tracker be $\{\boldsymbol{x}_i(0)\}$ with centroid $\boldsymbol{x}_{\mathrm{E}}(0)$ and zero initial velocity. Then external eye tracking proceeds according to the following algorithm.

Eye tracking

1. *State prediction*. A new (time $t+1$) visual state is predicted based on the current state and a constant velocity model.
2. *Image search*. In a neighborhood of each predicted tracker's point, a local search of edge points (brightness gradient maxima) takes place at each of the n tracker's points and along normal directions to the tracker itself. The set $\{\tilde{\boldsymbol{x}}_i(t+1)\}$ of edge points (measurement set) is computed by means of a recursive coarse-to-fine algorithm based on finite differences.
3. *Least squares fit*. The LS approximation of the new template centroid $\boldsymbol{x}_{\mathrm{E}}(t+1)$ is simply the centroid of the measurement set: $\tilde{\boldsymbol{x}}_{\mathrm{E}}(t+1) = (1/n)\sum_i \tilde{\boldsymbol{x}}_i(t+1)$. The 2×2 matrix $\mathrm{L}_{\mathrm{E}}(t+1)$ is also estimated via LS as the best approximation $\tilde{\mathrm{L}}_{\mathrm{E}}(t+1)$ of the affine transformation about the origin between the measurement set and the reference template.
4. *Filtering*. The six parameters of the affine image transformation $[\tilde{\mathrm{L}}_{\mathrm{E}}(t+1), \tilde{\boldsymbol{x}}_{\mathrm{E}}(t+1)]$ are smoothed using a mobile mean filter. To achieve a better control of the tracking process, a different filter gain is assigned to each parameter.
5. *Affine state projection*. Finally, once the affine transformation $[\hat{\mathrm{L}}_{\mathrm{E}}(t+1), \hat{\boldsymbol{x}}_{\mathrm{E}}(t+1)]$ is obtained, the new tracker is computed as $\boldsymbol{x}_i(t+1) = \boldsymbol{x}_{\mathrm{E}}(t+1) + \hat{\mathrm{L}}_{\mathrm{E}}(t+1)[\boldsymbol{x}_i(0) - \boldsymbol{x}_{\mathrm{E}}(0)]$, $i = 1, \ldots, n$, with $\boldsymbol{x}_{\mathrm{E}}(t+1) = \hat{\boldsymbol{x}}_{\mathrm{E}}(t+1)$. The new tracker point velocities can now be estimated from the LS comparison between the new (time $t+1$) and old (time t) tracker instances. The affine projection ensures that at each time t the tracker is an affine-transformed instance of the reference tracker obtained at $t = 0$.

Experimental evidence has demonstrated that this tracking algorithm has a good performance in terms of tracking efficiency (it executes in real-time even with a PC based system), adaptability to environment (especially lighting) conditions and to human subjects variability, accuracy and reliability. Specifically, the recovery time from tracking lags is inversely proportional to the template-to-eye mismatch, while the average time before tracking loss is about 20 minutes.

3D parameters estimation. In order to remap user movements into 3D pointer commands for the system, the 2D information extracted as above is used to derive 3D estimates related to head movements. An estimate of relative translation $({}^{\beta_0}_{\beta}\boldsymbol{t})$ and orientation (τ, σ, ϕ) can be obtained, in principle, from a single affine transformation $(\mathrm{L}_{\mathrm{E}}, \boldsymbol{x}_{\mathrm{E}})$, based on $\boldsymbol{x}_{\mathrm{E}}(t)$, $\boldsymbol{x}_{\mathrm{E}}(0)$, and the manipulation of $\mathrm{L}_{\mathrm{E}}(t)$; this method (see Appendix A) requires a disambiguation strategy for the τ and ϕ angles, which proves to be not robust enough for this operational context. Therefore, we introduce here a more direct approach, exploiting the piecewise affine camera model expounded in Section 2 to obtain a robust and unambiguous estimate of both head pose and relative translation from the comparison of the current (time t) and reference (time 0) trackers for the left and right eyes. Such a comparison is expressed through the transformations $(\mathrm{L}_{\mathrm{L}}, \boldsymbol{x}_{\mathrm{L}})$ and $(\mathrm{L}_{\mathrm{R}}, \boldsymbol{x}_{\mathrm{R}})$. The quantities of interest are the ocular centers $\boldsymbol{x}_{\mathrm{L}}$ and $\boldsymbol{x}_{\mathrm{R}}$, the *depth ratio* $\eta = {}^{\gamma}Z_{\mathrm{R}}/{}^{\gamma}Z_{\mathrm{L}}$ and the *weighted interocular difference* $\delta\boldsymbol{x} = \boldsymbol{x}_{\mathrm{L}} - \eta\boldsymbol{x}_{\mathrm{R}}$. We see from Eqs. (1) and (2) that η can be estimated as

$$\eta = \sqrt{\det \mathrm{L}_{\mathrm{L}} / \det \mathrm{L}_{\mathrm{R}}}, \tag{3}$$

where the determinant is a measure related to area changes of the imaged pattern. Besides, from the face planarity assumption and the projection model of Eq. (1) we easily get that the σ and τ angles are related to each other by

$$\tan\sigma = \frac{\lambda(\eta - 1)}{\begin{bmatrix} \cos\tau & \sin\tau \end{bmatrix}^{\mathrm{T}} \cdot \delta\boldsymbol{x}} \geqslant 0. \tag{4}$$

Fig. 4 shows the relationship between tracker's shape and head position when the head is in an arbitrary (*left*) and in the reference (*right*) position. We assume that, independently from head cyclotorsions (rotations about the normal to the face plane), *the slant angle σ always coincides with the*

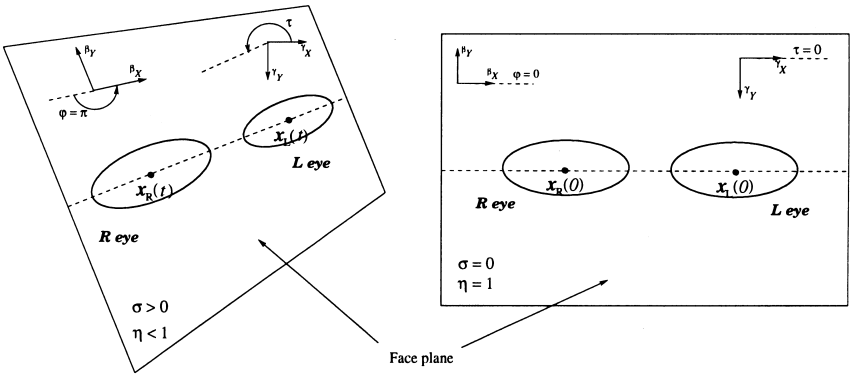

Fig. 4. The 2D parameters used for 3D head displacement estimation.

head pan angle (rotation about the neck axis). A consequence of equating pan and slant angles is that the ϕ angle equals 0 for $\eta \geqslant 1$, and π otherwise. Another consequence is that the direction of the line passing through the left and right ocular centers coincides with the direction of maximum depth variation, i.e. the absolute value of the denominator in Eq. (4) is maximum. This, and the fact that the left-hand side of Eq. (4) must be positive or zero, yields the following equality which can be used to compute τ:

$$[\cos\tau \quad \sin\tau]^{\mathrm{T}} = \mathrm{sgn}(\eta - 1)\,\delta\boldsymbol{x}/\|\delta\boldsymbol{x}\|, \tag{5}$$

where $\mathrm{sgn}(0) \doteq 1$. Using the computed value of τ in Eq. (4) provides us finally with an estimate of the pan angle σ.

So far with the orientation estimation. Concerning translations we notice that, since $\|{}^{\gamma}\boldsymbol{p}_{\mathrm{R}}\|$ is much larger than ${}^{\beta}Z$, we can neglect this unknown value and estimate ${}^{\beta_o}_{\beta(t)}\boldsymbol{t}$ simply as ${}^{\gamma}\boldsymbol{p}_{\mathrm{R}}(t) - {}^{\gamma}\boldsymbol{p}_{\mathrm{R}}(0)$. Hence we have

$${}^{\beta_o}_{\beta(t)}\boldsymbol{t} \propto {}^{\gamma}_{\beta_0}\mathrm{R}\left\{\zeta_{\mathrm{R}}(t)\begin{bmatrix}\boldsymbol{x}_{\mathrm{R}}(t)\\ \lambda\end{bmatrix} - \begin{bmatrix}\boldsymbol{x}_{\mathrm{R}}(0)\\ \lambda\end{bmatrix}\right\}, \tag{6}$$

where the relative depth for the right eye $\zeta_{\mathrm{R}}(t) = {}^{\gamma}Z_{\mathrm{R}}(t)/{}^{\gamma}Z_{\mathrm{R}}(0)$ is computed using the weighted interocular differences, the depth ratio and the slant, as $\zeta_{\mathrm{R}}(t) = \eta(t)\cos\sigma(t)\,\|\delta\boldsymbol{x}(0)\|/\|\delta\boldsymbol{x}(t)\|$. The translation parameters can be recovered using Eq. (6) up to the unknown scale factor $\kappa_{\mathrm{R}}(0)$. This indetermination – which has no practical drawbacks in our operating context, see next section – originates from modeling the eyes not from how they look like in the face plane but from their image appearance.

Fig. 5 presents some examples of real-time estimation of 3D head parameters. In each of the examples, the computed pose parameters are graphically represented by the perspective view of a planar disc and its associated normal vector (upper left corner of each image). Translation parameters – shown just below the pose parameters – are synthetically encoded through a 2D oriented segment (representing the translation components parallel to the image plane) and a circle, whose radius is inversely proportional to the head translation component perpendicular to the camera. The first row of Fig. 5 shows (at left) the situation soon after the reference tracker acquisition, and (at right) the situation after a head translation in the direction of increasing depth (notice that the translation circle is smaller than that of the reference position). In the second row of Fig. 5, head pose estimates are shown after a rightwards head pan and upwards translation (left), and (right) after a rightwards head pan of double magnitude accompanied by a rotation around the optical axis. Notice, especially in the last case, a shape change w.r.t. the reference very similar to that illustrated in Fig. 4: one eye is becoming bigger and the other is becoming smaller.

Comparative experiments conducted at fixed (ground truth) head positions have shown the superiority of the piecewise affine representation for

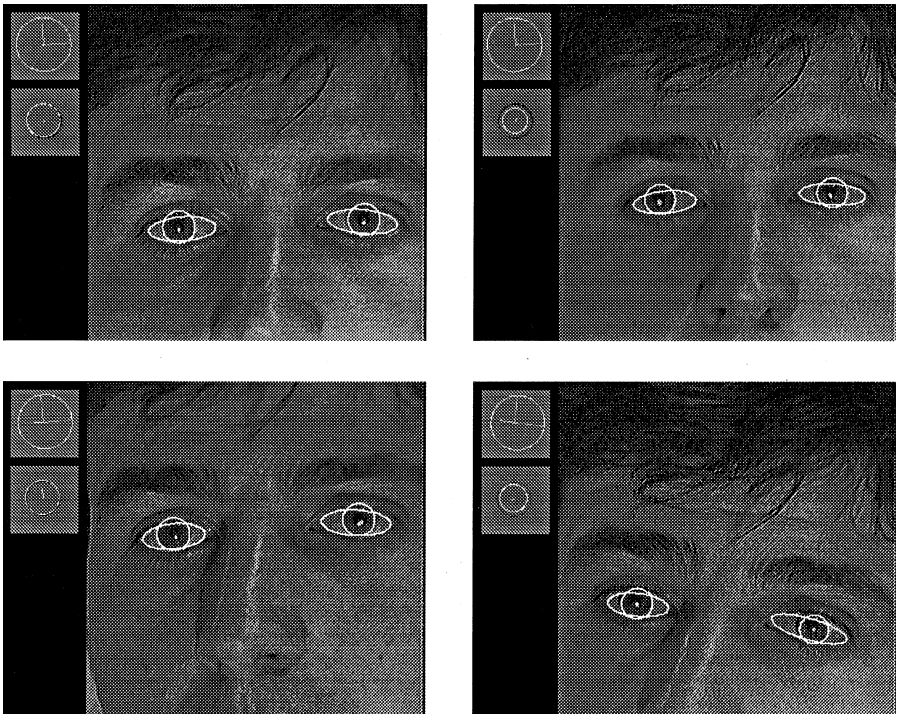

Fig. 5. Head pose estimation examples. In raster order: reference, translation away from the camera, $\sigma = \pi/8$ rotation plus upwards translation, $\sigma = \pi/4$ rotation plus counterclockwise cyclotorsion.

3D pose estimation expounded above over the single affine map strategy sketched in Appendix A. In particular, the average orientation error in the first case is always below 5°, and up to 15° in the second case.

4. Interactive graphics application

The 2D tracking and 3D pose estimation system described above was used for the design and implementation of a 3D interactive graphics environment, visualizing a virtual museum featuring canvases by famous 20th century artists. The environment allows to navigate around in the museum so as to inspect each single canvas.

Graphical remapping and interface semantics. User head displacements are remapped into the interactive environment by means of a *virtual camera*, representing the imaginary device used to obtain the 2D on-screen view of the virtual scene from a given point of the 3D graphic environment (see (Foley and van Dam, 1982)). Specifically, any 3D user head motion is replicated in the environment, as if the virtual camera was moving in his place. The remapping is one–one for orientation angles, and proportional for translations, with a constant of proportionality controlling interface sensitivity to the amount of user translation:

$$ {}^{\chi}\boldsymbol{p} = {}^{\beta_0}_{\beta}\mathrm{R}^{\mathrm{T}}\left[{}^{\chi_0}\boldsymbol{p} - \mathrm{diag}(k_x, k_y, k_z)^{\beta_0}_{\beta}\boldsymbol{t}\right], \tag{7} $$

where $\mathrm{diag}(k_x, k_y, k_z)$ is the translation scaling diagonal matrix. In Eq. (7) we assume that the 3D virtual scene is defined in terms of χ_0 coordinates, and that the virtual camera frame at time t is $\chi = \chi(t)$, with $\chi(0) = \chi_0$. Using this method, as the user approaches the screen without rotating the head (translation in the positive ${}^{\beta_0}Z$ direction), the objects in the observed scene come closer, with an amount of scale change depending directly on the remapping constant k_z. Proportional viewpoint

control is basically a *qualitative* way of performing remapping which has the advantage of providing the user with a natural feedback as the result of his action. Non-proportional viewpoint control or the presence of large 3D parameter estimation errors would produce an unexpected interface behavior, with the result of confusing the user.

The dialogue semantics currently implemented in the interface is based on a *drag and click metaphor*, which lets the user interact with the environment with his head to perform navigational and selection actions as with conventional pointing devices. Explicitly, head displacements w.r.t. a fixed reference are interpreted as pointer drags, or *navigational actions*; the pointer can instead trigger a *selection action* (click) as it persists in the neighborhood of a geometric configuration for a convenient time interval. Hence, the head can be used to navigate or to displace objects (drag) and to select, or "freeze", a 3D scene of interest in it (click). An issue of relevant practical importance is the time responsiveness of the interface; this is directly related to the time threshold used to measure persistence and assign the proper semantics to user action. Choosing the right threshold value is of key importance to have a good balance between speed of operation and naturality of interaction. A good time threshold taking into account the relative slow mobility of the head, is 2 s, which on the one hand guarantees a fast response, and on the other limits the occurrence of false alarms.

Equipment and interaction examples. The interface uses the OpenGL graphic libraries and runs on a Silicon Graphics Indy workstation. The vision subsystem software also runs on the Indy, and gets raw image data through a VINO frame grabber board from an inexpensive B/W camera. The overall interaction loop time for our system is the sum of the time spent doing visual computations and graphic environment manipulation. Initializating visual algorithms involves automatic eye extraction and template initialization and takes around 450 ms to complete. At run-time visual tracking runs at video rate (25 Hz) instead, using $n = 64$ sampling points for external eye search. Without special hardware for graphics acceleration, most of the loop time is taken by 3D graphic remapping (some hundreds of ms at an intermediate picture quality).

Experiments have been performed with several users and different interaction conditions. Examples of typical interaction sessions are illustrated in Fig. 6 (left, right), in which image frames are presented in raster order. Fig. 6 (left) shows a zoom-in sequence. Zooming is obtained by approaching the screen with the head; this causes the painting in the middle of the wall to be displayed at full resolution. Fig. 6 (right) illustrates the generation of a viewpoint change determined by a head rotation: a leftward head pan causes the graphic environment to move rightwards, and display a previously invisible museum wall. Notice the simultaneous presence of a slight leftward head translation. Once a specific viewpoint has been selected by head rotation, the user can produce a "click" (scene freeze), and go back to the reference position, so as to inspect the on-screen scene and

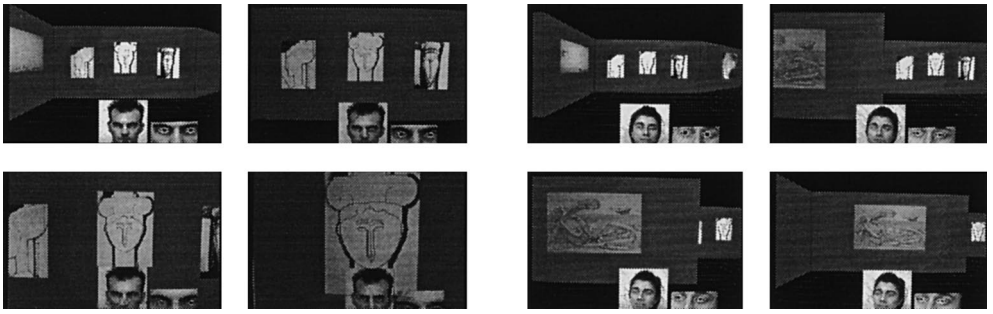

Fig. 6. Zoom (left) and Pan (right) sequences.

learn more about painter/period by clicking with the mouse on any canvas.

5. Conclusions and future work

In this paper, a computer vision based approach for advance human–computer interaction through head displacements has been presented, and integrated in a non-immersive virtual reality application. Such an approach exploits a piecewise affine camera model to ensure a robust and fast tracking of eye contours, whose geometrical deformations are then used to estimate head pose and translation parameters and drive the interaction according to a simple interface metaphor.

The approach can be easily modified so as to support different advanced interaction scenarios and applications. Another direction of future research is to fully couple the head driven 3D approach presented here with the gaze driven approach presented by the authors in (Colombo and Del Bimbo, 1997), featuring exactly the same eye tracking engine but with the different scope of enabling interaction in 2D contexts based on eye pupil shifts.

Appendix A. Pose and depth from a single affine projection map

The scope of this appendix is to show how the 3D quantities that enter in the affine projection map of a planar object – namely, pose (τ, σ, ϕ) and centroid depth ${}^{\gamma}Z_{\mathrm{O}}$ – can be recovered in principle from the knowledge of $\mathrm{H} = \mathrm{LH}(0)$. In our case, the planar object of interest can be either a single eye or the left and right eyes taken together, and considered as a subset of the face plane. First of all, let us express H in terms of its (unique) decomposition

$$\mathrm{H} = \frac{1}{2}\left\{ \begin{bmatrix} \delta & -\gamma \\ \gamma & \delta \end{bmatrix} + \begin{bmatrix} \alpha & \beta \\ \beta & -\alpha \end{bmatrix} \right\}. \tag{A.1}$$

Since the numbers δ (divergence) and γ (curl) are invariant w.r.t. coordinate system rotations, referring to Eq. (1) it is easy to show that the unknown 3D parameters are related to the entries of H by

$$\begin{aligned} \delta &= \kappa_{\mathrm{O}} c_{\tau-\phi}(c_\sigma - 1), \\ \gamma &= \kappa_{\mathrm{O}} s_{\tau-\phi}(c_\sigma - 1), \\ \beta &= \kappa_{\mathrm{O}} s_{\tau+\phi}(c_\sigma + 1), \\ \alpha &= \kappa_{\mathrm{O}} c_{\tau+\phi}(c_\sigma + 1). \end{aligned} \tag{A.2}$$

Due to the pose ambiguity (see Section 2), the nonlinear system of Eq. (A.2) admits the dual solutions $(\overline{\tau}, \overline{\sigma}, \overline{\phi}; {}^{\gamma}\overline{Z}_{\mathrm{O}})$ and $(\overline{\tau}+\pi, \overline{\sigma}, \overline{\phi}+\pi; {}^{\gamma}\overline{Z}_{\mathrm{O}})$. Explicitly, it holds

$$\overline{\sigma} = \arccos\left(\frac{\sqrt{\alpha^2+\beta^2} - \sqrt{\gamma^2+\delta^2}}{\sqrt{\alpha^2+\beta^2} + \sqrt{\gamma^2+\delta^2}} \right) \tag{A.3}$$

(with the constraint $\sqrt{\alpha^2+\beta^2} \geqslant \sqrt{\gamma^2+\delta^2}$),

$${}^{\gamma}\overline{Z}_{\mathrm{O}} = \frac{2\lambda}{\sqrt{\alpha^2+\beta^2} + \sqrt{\gamma^2+\delta^2}}, \tag{A.4}$$

where the term $\sqrt{\alpha^2+\beta^2}$ (deformation) is also invariant w.r.t. the local coordinate system, and

$$\begin{aligned} 2\overline{\tau} &= \arctan\left(\frac{\beta}{\alpha}\right) + \arctan\left(\frac{\gamma}{\delta}\right), \\ 2\overline{\phi} &= \arctan\left(\frac{\beta}{\alpha}\right) - \arctan\left(\frac{\gamma}{\delta}\right). \end{aligned} \tag{A.5}$$

Following the strategy proposed in (Horaud et al., 1995), the two solutions can be disambiguated by choosing, between the two candidate solutions, the one with smallest LS error with respect to raw tracking data, after template data resynthesis using the computed 3D parameters and the full perspective model. This method works well, of course, if the differences between the full pespective view and the affine view are large, i.e., in our case, if the left and right eyes are considered as a single planar object. However, if the departure from the affine model is significant, the tracking performance using affine templates gets worse, and an incorrect estimate of the candidate solutions is obtained. Besides, even if a single eye map is considered, estimating pose and depth through Eqs. (A.3)–(A.5) can lead to gross errors due to ill-conditioning, since the matrix entries are quite small if the object extension is small. That is why, in this paper, 3D parameters are estimated by

decomposing the overall projection map into two different affine maps (see Section 3).

References

Allotta, B., Colombo, C., 1999. On the use of linear camera–object interaction models in visual servoing. IEEE Trans. Robotics Automation 15 (2), 350–357.

Azarbayejani, A., Starner, T., Horowitz, B., Pentland, A., 1993. Visually controlled graphics. IEEE Trans. PAMI 15 (6), 602–605.

Brunelli, R., Poggio, T., 1993. Face recognition: Features versus templates. IEEE Trans. PAMI 15 (10), 1042–1052.

Cipolla, R., Hollinghurst, N., 1996. Human–robot interface by pointing with uncalibrated stereo vision. Image and Vision Computing 14, 171–178.

Cipolla, R., Hollinghurst, N., 1997. Visually guided grasping in unstructured environments. Robotics and Autonomous Systems 19 (3–4), 337–346.

Colombo, C., Del Bimbo, A., 1997. Interacting through eyes. Robotics and Autonomous Systems 19 (3–4), 359–368.

Crowley, J.L., 1997. Vision for man–machine interaction. Robotics and Autonomous Systems 19 (3–4), 347–358.

Essa, I.A., Pentland, A.P., 1997. Coding analysis, interpretation, and recognition of facial expressions. IEEE Trans. PAMI 19 (7), 757–763.

Foley, J., van Dam, A., 1982. Fundamentals of Interactive Computer Graphics. Addison–Wesley, Reading, MA.

Gee, A.H., Cipolla, R., 1996. Fast visual tracking by temporal consensus. Image and Vision Computing 14 (2), 105–114.

Horaud, R., Christy, S., Dornaika, F., Lamiroy, B., 1995. Object pose: Links between paraperspective and perspective. In: Proc. Fifth ICCV, Cambridge, MA, pp. 426–433.

Lam, K.-M., Yan, H., 1998. An analytic-to-holystic approach for face recognition based on a single frontal view. IEEE Trans. PAMI 20 (7), 673–686.

Mundy, J.L., Zisserman, A., 1992. Projective geometry for machine vision. In: Geometric Invariance in Computer Vision. MIT Press, Cambridge, MA, pp. 463–519.

Myers, B.A., 1998. A brief history of human–computer interaction technology. Interactions 5 (2), 44–54.

Nielsen, J., 1993. Noncommand user interfaces. Commun. ACM 36 (4), 83–99.

Pavlovic, V.I., Sharma, R., Huang, T.S., 1997. Visual interpretation of hand gestures for human–computer interaction: A review. IEEE Trans. PAMI 19 (7), 677–695.

Pentland, A.P., 1996. Smart rooms. Scientific American 274 (4), 54–62.

Yuille, A., Hallinan, P., 1992. Deformable templates. In: Active Vision. MIT Press, Cambridge, MA, pp. 21–38.