Proc. of IbPria2003. LNCS 2652, pp. 126-133

Tracking heads using piecewise planar models

José M. Buenaposada, Enrique Muñoz, and Luis Baumela

Universidad Politécnica de Madrid, Departamento de Inteligencia Artificial
Campus de Montegancedo s/n, 28660 Madrid (Spain)
{jmbuena,kike}@dia.fi.upm.es, lbaumela@fi.upm.es

Abstract. We present a procedure for tracking a rigid object based on a piecewise planar model, and show how it can be used to track a human face. The tracking is performed using a single incremental SSD-based tracker. The main feature of the approach presented is that it can track a rigid set of arbitrarily small patches all of which could not be individually tracked.

1 Introduction

Three-dimensional head tracking is a basic component in many applications of computer vision. For instance, the construction of advanced computer interfaces deals with problems such as the identification of head gestures, face expression analysis or lip reading. It is also used in biometric applications, like face or iris-based recognition, for which a stable location of the face is critical. Also, for very low bit-rate communications, the MPEG-4 standard proposes the use of animated artificial face models in a wide range of applications from virtual videoconferencing to virtual actors. All these applications require a robust and efficient (i.e. real-time or near real-time) head tracker with no markers on it.

Various techniques have been proposed in the literature for head tracking. Some of them only track the 2D position of the face on the image plane [2, 6], others model the face as a plane, which can be affinely or projectively [7, 3, 4] tracked in 3D space. Finally, there is a third group of procedures which rely on a 3D model of the face. These are based on individually tracking a set of salient points [11], 2D image patches [8, 9, 12], or 3D surface-based head models [10].

Procedures based on individually tracking a set of features can be quite unstable as each feature, individually, may not provide enough information to be tracked. In order to cope with this problem some higher level process, like a Kalman filter [9, 12] or a set motion restrictions propagated on a network of features [8], are used to accumulate the information provided by the tracker of each feature/patch in order to estimate the motion of the head. This problem does not exist for methods which model the face with a single surface, but, on the other hand, those based on a single-plane are not able to track the head in presence of out-of-the-image plane rotations [7, 3, 4], whereas those which are based on a more complex head model, for example a cylinder [10], need computationally expensive warping algorithms.

In this paper we present a procedure for model-based head tracking. The model is based on a set of image patches located in space with a known 3D structure. Our approach differs from previous feature/patch-based trackers [8, 9, 12] in that we track all features using a single incremental tracker [7, 4]. In this way we integrate in a single tracker the low level information provided by all patches in the image, enabling us to reliably track a set of arbitrarily small patches, all of which could not be individually tracked. In section 2 we briefly introduce the incremental image alignment paradigm. In section 3 we build the tracker. Finally in sections 4 and 5 some experiments are presented and conclusions drawn.

2 Incremental image registration

Let $\mathbf{x}$ represent the location of a point in an image and $I(\mathbf{x}, t)$ represent the brightness value of that location in the image acquired at time t. Let $\mathcal{R} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of N image points of the object to be tracked (*target region*), whose brightness values are known in the first image of a sequence, $I(\mathbf{x}, t_0)$.

Assuming that the brightness constancy assumption holds, then

$$I(\mathbf{x}, t_0) = I(\mathbf{f}(\mathbf{x}, \bar{\mu}_t), t) \; \forall \mathbf{x} \in \mathcal{R}, \tag{1}$$

where $I(\mathbf{f}(\mathbf{x}, \bar{\mu}_t), t)$ is the image acquired at time t rectified with motion model $\mathbf{f}$ and motion parameters $\bar{\mu} = \bar{\mu}_t$.

Tracking the object means recovering the motion parameter vector of the target region for each image in the sequence. This can be achieved by minimising the difference between the template and the rectified pixels of the target region for every image in the sequence

$$\min_{\bar{\mu}} \sum_{\forall \mathbf{x} \in \mathcal{R}} \left[I(\mathbf{f}(\mathbf{x}, \bar{\mu}), t) - I(\mathbf{x}, t_0) \right]^2 \tag{2}$$

This minimisation problem has been traditionally solved linearly by computing $\bar{\mu}$ incrementally while tracking. We can achieve this by making a Taylor series expansion of (2) at ($\bar{\mu}$, t) and computing the increment in the motion parameters between two time instants. Different solutions to this problem have been proposed in the literature, depending on which term of equation (2) the Taylor expansion is made on and how the motion parameters are updated [1].

If we update the model parameters of the first term in equation (2) using an additive method, then the minimisation can be rewritten as [1, 5]

$$\min_{\delta\bar{\mu}} \sum_{\forall \mathbf{x} \in \mathcal{R}} \left[I(\mathbf{f}(\mathbf{x}, \bar{\mu}_t + \delta\bar{\mu}), t + \delta t) - I(\mathbf{x}, t_0) \right]^2, \tag{3}$$

where $\delta\bar{\mu}$ represents the estimated increment in the motion parameters of the target region between time instants t and $t + \delta t$.

The solution to this linear minimisation problem can be approximated by [5]

$$\delta\bar{\mu} = -\mathbf{H}_0^{-1} \sum_{\forall \mathbf{x} \in \mathcal{R}} \mathbf{M}(\mathbf{x}, \mathbf{0})^\top \mathcal{E}(\mathbf{x}, t + \delta t), \tag{4}$$

where $\mathbf{H}_0$ is

$$\mathbf{H}_0 = \sum_{\forall \mathbf{x} \in \mathcal{R}} \mathbf{M}(\mathbf{x}, \mathbf{0})^\top \mathbf{M}(\mathbf{x}, \mathbf{0}),$$

$\mathcal{E}(\mathbf{x}, t + \delta t)$ is the error in the estimation of the motion of pixel $\mathbf{x}$ of the target region

$$\mathcal{E}(\mathbf{x}, t + \delta t) = I(\mathbf{f}(\mathbf{x}, \bar{\mu}_t), t + \delta t) - I(\mathbf{x}, t_0),$$

and $\mathbf{M}(\mathbf{x}, \mathbf{0})$ is the Jacobian vector of pixel $\mathbf{x}$ with respect to the model parameters $\bar{\mu}$ at time instant t_0 (we will assume $\bar{\mu}_{t_0} = \mathbf{0}$). If $\mathbf{f}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$, then $\mathbf{M}(\mathbf{x}, \mathbf{0})$ can be expressed as

$$\mathbf{M}(\mathbf{x}, \mathbf{0}) = \left. \frac{\partial I(\mathbf{f}(\mathbf{x}, \bar{\mu}), t_0)}{\partial \bar{\mu}} \right|_{\bar{\mu}=\mathbf{0}} = \nabla_{\mathbf{x}} I(\mathbf{x}, t_0)^\top \left[\frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial \bar{\mu}} \right]_{\bar{\mu}=\mathbf{0}},$$

where $\nabla_{\mathbf{x}} I(\mathbf{x}, t_0)$ is the template image gradient and $\frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial \bar{\mu}}$ is the Jacobian vector of the motion model.

The Jacobian of pixel $\mathbf{x}$ with respect to the model parameters in the reference template, $\mathbf{M}(\mathbf{x}, \mathbf{0})$, is a vector whose values are our *a priori* knowledge about target structure, i.e. how the brightness value of each pixel in the reference template changes as the object moves infinitesimally. It represents the information provided by each template pixel to the tracking process. Note that when $\mathbf{H}_0 = \sum_{\forall \mathbf{x} \in \mathcal{R}} \mathbf{M}(\mathbf{x}, \mathbf{0})^\top \mathbf{M}(\mathbf{x}, \mathbf{0})$ is singular the motion parameters cannot be recovered, this would be a generalisation of the so called *aperture problem* in the estimation of optical flow.

- `Offline computations:`
  1. `Compute and store` $\mathbf{M}(\mathbf{x}, \mathbf{0})$.
  2. `Compute and store` $\mathbf{H}_0$.
- `On line computations:`
  1. `Warp` $I(\mathbf{z}, t + \delta t)$ `to compute` $I(\mathbf{f}(\mathbf{x}, \bar{\mu}_t), t + \delta t)$.
  2. `Compute` $\mathcal{E}(\mathbf{x}, t + \delta t)$.
  3. `From` (4) `compute` $\delta\bar{\mu}$.
  4. `Update` $\bar{\mu}_{t+\delta t} = \bar{\mu}_t + \delta\bar{\mu}$.

Fig. 1. Outline of the incremental tracking algorithm

The on-line computation performed by this tracking procedure is quite small (see Fig. 1) and consists of a warping of N pixels, which can be made very fast by conventional software o even by specialised hardware, a subtraction of N pixels to compute $\mathcal{E}(\mathbf{x}, t + \delta t)$, the addition of N vectors multiplied by one constant, and the multiplication of this result by the $n \times n$ matrix ${\mathbf{H}_\mathbf{0}}^{-1}$, where $n = \dim(\bar{\mu})$.

3 The tracker

In this section we will introduce the target region motion model, $\mathbf{f}$, and show how to compute the image Jacobian $\mathbf{M}(\mathbf{x}, \mathbf{0})$ with respect to the parameters of the model.

3.1 Motion model

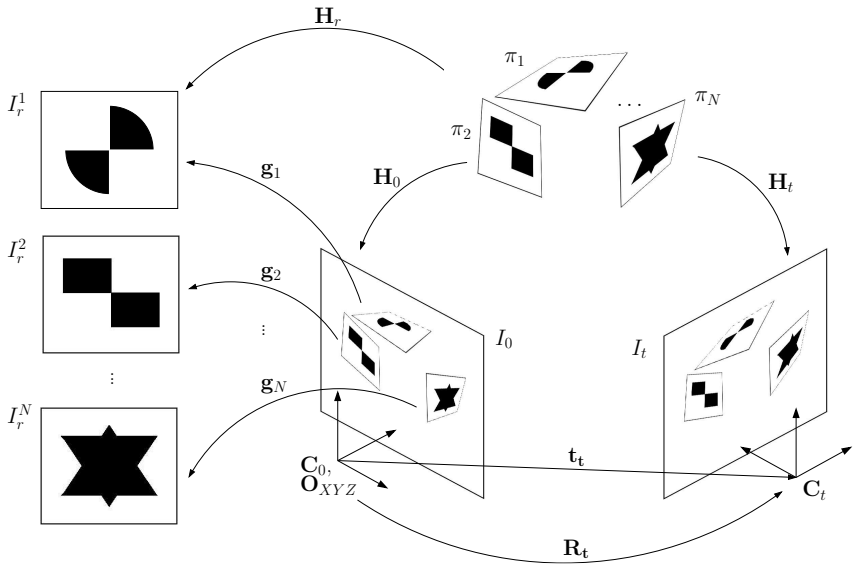

Fig. 2. Geometrical set up of the tracking process.

Let $\{\pi_i\}$ be a set of N planar patches in 3D space, each one containing a target region. Each patch, π_i, of this set can be described by equation $\pi_i \equiv \mathbf{n}_i^\top \mathbf{P} = 1$, where $\mathbf{n}_i = [a, b, c]^\top$ is a three-element vector containing the normal direction to the plane π_i, and $\mathbf{P} = [X, Y, Z]^\top \in \pi_i$ are the coordinates of a 3D point on that plane expressed in the reference system of the scene, O_{XYZ}. Each plane, π_i, will have a *reference template* or high-resolution image of the target region, I_r^i, associated to it. At the initial time instant, we will assume that the reference systems attached to the camera and scene are perfectly aligned.

The projection of a point on a planar patch $\mathbf{P}_{\pi_j}$ onto image I_i of the sequence is given by

$$\mathbf{x}_i^{\pi_j} = \underbrace{\mathbf{K}\,\mathbf{R}_i[\,\mathbf{I} - \mathbf{t}_i\mathbf{n}_j^\top\,]}_{\mathbf{H}_i}\,\mathbf{P}_{\pi_j}, \tag{5}$$

where $\mathbf{K}$ is the camera intrinsics matrix, which is assumed to be known, $\mathbf{I}$ is the 3×3 identity matrix, $\mathbf{R}_i$, $\mathbf{t}_i$ represent the pose of the camera and $\mathbf{x}_i^{\pi_j}$ represents

the homogeneous coordinates of the pixel projection. As we are dealing with 3D points that are located on planes, their projection model is a 2D linear projective transformation or *homography*, $\mathbf{H}_i$.

The motion model, $\mathbf{f}(\mathbf{x}, \bar{\mu})$, can be derived from (5) by considering the projection of 3D point $\mathbf{P}_{\pi_j}$ onto $I_0 \equiv I(\mathbf{x}_0, t_0)$ and onto $I_t \equiv I(\mathbf{x}_t, t)$

$$\mathbf{x}_t^{\pi_j} = \mathbf{K}\,\mathbf{R}_t[\,\mathbf{I} - \mathbf{t}_t\mathbf{n}_j^\top\,]\,\mathbf{K}^{-1}\mathbf{x}_0^{\pi_j},$$

where, $\mathbf{R}_t(\alpha, \beta, \gamma)$ and $\mathbf{t}_t(t_x, t_y, t_z)$ are the six parameters, $\bar{\mu} = (\alpha, \beta, \gamma, t_x, t_y, t_z)^\top$, of the motion model, which represent the pose of the camera with respect the first image in the sequence. Note that, since our scene is rigid, these motion parameters are common to all patches π_j in the model.

3.2 The image Jacobian

In this subsection we will show how to compute the second element of our algorithm, $\mathbf{M}(\mathbf{x}, \mathbf{0})$.

Due to partial occlusions, perspective effects or low resolution, the projection of a target region onto I_0 may not provide enough information to accurately compute $\nabla_{\mathbf{x}} I(\mathbf{x}, t_0)$. In this case we use the reference template to compute it, through the following relation

$$\nabla_{\mathbf{x}} I(\mathbf{x}, t_0)|_{\forall \mathbf{x} \in \pi_i} = \left[\frac{\partial I_r^i(\mathbf{g}_i(\mathbf{x}, \bar{\mu}))}{\partial \mathbf{g}_i}\right]^\top \left[\frac{\partial \mathbf{g}_i(\mathbf{x}, \bar{\mu})}{\partial \mathbf{x}}\right],$$

where $\mathbf{g}_i$ is the warping function that transforms the projection of planar patch π_i in image I_0 onto reference template I_r^i, that is, $I_0(\mathbf{x}) = I_r^i(\mathbf{g}_i(\mathbf{x}, \bar{\mu}))\ \forall \mathbf{x} \in \pi_i$.

Finally, the Jacobian of the motion model with respect to the motion parameters is given by

$$\left.\frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial \bar{\mu}}\right|_{\bar{\mu}=\mathbf{0}} = \left[\frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial \alpha}, \ldots, \frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial t_z}\right]_{\bar{\mu}=\mathbf{0}}, \tag{6}$$

where, for example

$$\frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial \alpha} = \mathbf{K}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}\mathbf{K}^{-1}\mathbf{x}_0; \quad \text{and} \quad \frac{\partial \mathbf{f}(\mathbf{x}, \bar{\mu})}{\partial t_x} = -\mathbf{K}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\mathbf{K}^{-1}\mathbf{x}_0.$$

4 Experiments

We have carried out three experiments to test the tracking algorithm here presented, for each of which we have generated an image sequence (See videos at http://www.dia.fi.upm.es/~lbaumela/FaceExpressionRecognition/research.html). Sequences A and B were generated using pov-ray[1] (see Fig 3 and 4), in order to

[1] A free ray tracer software, `http://www.povray.org`

have ground truth data of the motion of our target. Sequence C (see Fig. 5) was captured with a Sony VL-500 CCD colour camera with no gain and no gamma correction.

In the first experiment we test the accuracy of our tracker. For this test we have used sequence A (see Fig. 3), in which a cube located 4 meters away from the camera translates along the X axis (t_x varies) and rotates around the Z axis (γ varies). As can be seen in Fig. 3 the parameters estimated with our tracker coincide with the ground truth data. Note that as we are generating the sequences with synthetic ligths and we are warping the textures over the planar patches (with aliasing effects involved), the sequences are not noise free.

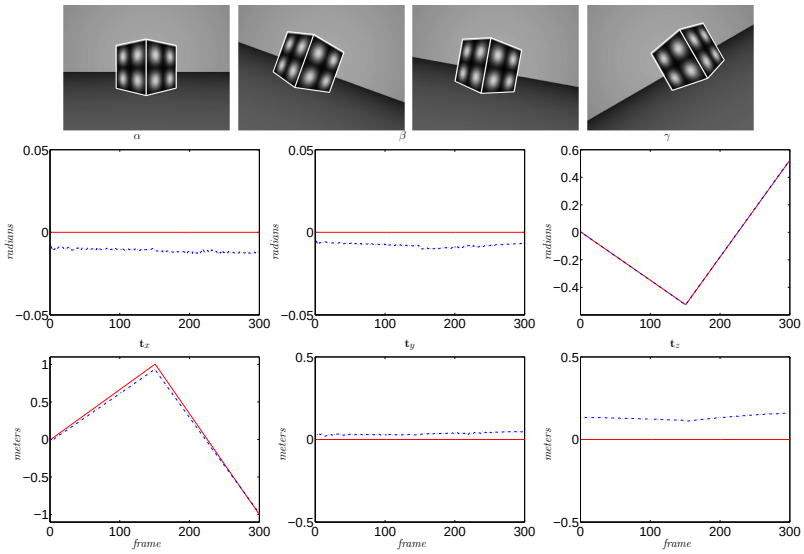

Fig. 3. Sequence A. First row: images 1, 100, 200 and 300 of the sequence. In white thick lines is shown the motion estimated by our tracker. Second and third rows: tracking parameters for sequence A. In solid line is shown the ground truth data and in dash-dot line is shown the motion estimated by the tracker.

The second experiment compares the tracking procedure presented in this paper with a traditional patch-based tracker in which each of the patches is tracked individually. For this test we have generated sequence B (see Fig. 4) which is identical to sequence A except that now the moving object is composed of two planar patches with textures which individually do not provide enough information for tracking. As shown in Fig. 4 the individual tracker diverges after a few frames. This is caused by the ambiguity of the textures in the patches.

In the last experiment we test the performance of our tracker when following a human face. For this test we use sequence C. As shown in Fig. 5, the tracker accurately tracks the face even for moderate out-of-the-image plane rotations.

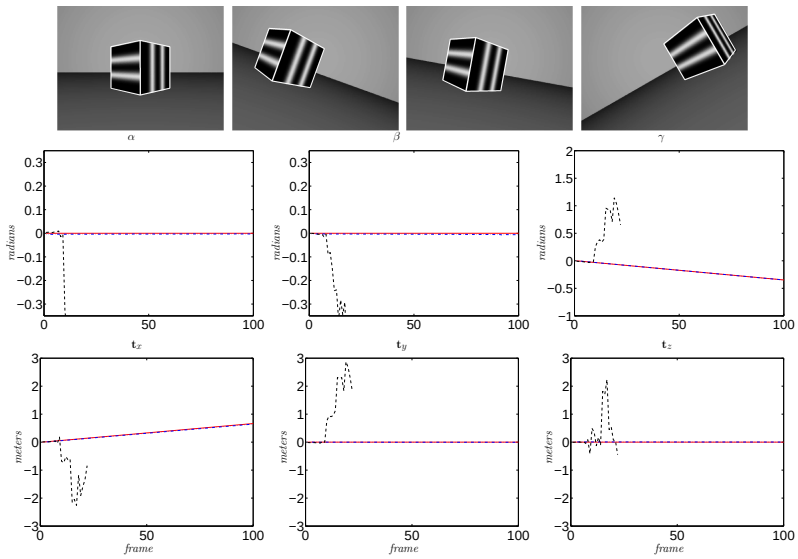

Fig. 4. Sequence B. First row: images 1, 100, 200 and 300 of the sequence. In white thick lines is shown the motion estimated by our tracker. Second and third rows: tracking parameters for the first 100 frames in sequence B. In solid line is shown the ground truth data, with dashed line is shown the estimation of the individual tracker, finally with dash-dot line is shown the motion estimated by our tracker.

These rotations could be even larger just by including patches taken from the sides of the head.

5 Conclusions

We have presented a procedure for tracking a rigid object based on a set of image patches. By integrating low level information in a single tracker we have been able to reliably track in 3D a set of patches which individually could not provide enough information. With this algorithm we could also track a face with out-of-the-image plane rotations, even with a poor face model.

Another issue that should be addressed in the future is the speed of convergence of the tracker. This is related to the approximation made to solve (3) and to the dependencies (correlations) in the columns of the $\mathbf{H_0}$ matrix, which are, in turn, directly related to the ambiguities in the estimation of the tracking parameters and which may result in slow convergence, and eventually divergence, of the tracker. Other open issues are the invariance to illumination changes and to variation in the texture of the patches (e.g. variations in face appearance).

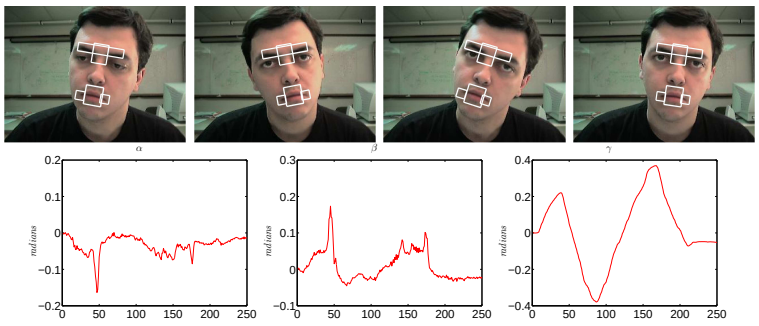

Fig. 5. Sequence C. Upper row: four images of the sequence. In white thick lines is shown the location of each feature estimated by the tracker. Bottom row: Estimated rotation parameters.

References

1. Simon Baker and Ian Matthews. Equivalence and efficiency of image alignment algorithms. In *Proc. of CVPR*, volume 1, pages I–1090–I–1097. IEEE, 2001.
2. Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. of CVPR*, pages 232–237. IEEE, 1998.
3. M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *Int. Journal of Computer Vision*, 25(1):23–48, 1997.
4. José M. Buenaposada and Luis Baumela. Real-time tracking and estimation of plane pose. In *Proc. of International Conference on Pattern Recognition*, volume II, pages 697–700, Quebec, Canada, August 2002. IEEE.
5. José M. Buenaposada, Enrique Muñoz, and Luis Baumela. Incremental image alignment. Technical Report DIA-CV-2003-01, Computer Vision Lab, Faculty of Computer Science, UPM, January 2003.
6. José M. Buenaposada, David Sopeña, and Luis Baumela. Face tracking using the dynamic grey world algorithm. In *Proc. of Computer Analysis of Images and Patterns*, volume LNCS 2124, pages 341–348, Warsaw, Poland, September 2001. Springer.
7. Gregory Hager and Peter Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998.
8. Gregory D. Hager and Kentaro Toyama. X vision: Combining image warping and geometric constraints for fast visual tracking. In *Proc. European Conference on Computer Vision (2)*, volume 1065 of *Lecture Notes on Computer Science*, pages 507–517. Springer, 1996.
9. Tony S. Jebara and Alex Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *Proc. of CVPR*, pages 144–150. IEEE Comput. Soc. Press, 1997.
10. S. Sclaroff M. La Cascia and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *PAMI*, 22(4), April 2000.
11. Rainer Stiefelhagen, Yie Yang, and Alex Waibel. A model-based gaze tracking system. *Int. Journal of Artificial Intelligence Tools*, 6(2):193–209, 1997.

12. Stephan Valente and Jean-Luc Dugelay. A visual analysis/synthesis feedback loop for accurate face tracking. *Signal Processing: Image Communications*, 16:585–608, 2001.