3D HEAD TRACKING BY PARTICLE FILTERS

F. Dornaika, F. Davoine, and M. Dang

CNRS HEUDIASYC
Compiègne University of Technology
60205 Compiègne Cedex, FRANCE
{*dornaika, fdavoine, dang*}*@hds.utc.fr*

ABSTRACT

In this paper, we propose a particle filter framework for 3D tracking of heads and faces in monocular video sequences. We propose two different approaches. The first approach utilizes a statistical facial texture model as an observation likelihood. The second approach utilizes a deterministic facial texture which is built on-line. The developed approaches have been successfully tested on several video sequences. The resulting 3D head tracker can find applications in many fields, such as human-computer interaction.

1. INTRODUCTION

Object tracking is required by many vision applications, especially in video technology and visual interface systems. Work on visual tracking can be divided into two groups: deterministic and stochastic tracking. Probabilistic video analysis has recently gained significant attention in the computer vision community. The idea of a particle filter [1] (also known as Sequential Monte Carlo (SMC) algorithm) was independently proposed and used by several vision research groups. These algorithms provide flexible tracking frameworks as they are neither limited to linear systems nor require the noise to be Gaussian and proved to be more robust to distracting clutter as the randomly sampled particles allow to maintain several competing hypotheses of the hidden state. Besides, these frameworks can easily handle uncertainties. These algorithms have gained prevalence in the visual object tracking literature due in part to the CONDENSATION algorithm [2].

Previous work using particle filters has focused on tracking 2D objects or motions such as 2D contours, 2D blobs, 2D affine transforms [3], and 2D ellipses [4, 5]. However, to our knowledge, a particle filter that tracks the six degrees of freedom associated with the motion of complex 3D objects was not reported.

In this paper, we propose a particle filter framework for tracking the 3D motion of heads/faces. The outline of the paper is as follows. In Section 2, we describe the 3D face model, the shape-free texture concept, and the statistical facial texture model. In Section 3, we describe a particle-filter-based 3D tracking algorithm using a statistical facial texture model. In Section 4, we describe a particle-filter-based 3D tracking algorithm using a deterministic facial texture. In Section 5, we present some experimental results.

2. MODELLING FACES

2.1. A 3D face model

In our study, we use the 3D face model *Candide*. The 3D shape of this deformable 3D wireframe model is directly recorded in coordinate form. The 3D face model is given by the 3D coordinates of the vertices $\mathbf{P}_i, i = 1, \ldots, n$ where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector $\mathbf{g}$ – the concatenation of the 3D coordinates of all vertices $\mathbf{P}_i$. The vector $\mathbf{g}$ can be written as:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{S}\,\sigma + \mathbf{A}\,\alpha \tag{1}$$

where $\overline{\mathbf{g}}$ is the standard shape of the model, and the columns of $\mathbf{S}$ and $\mathbf{A}$ are the Shape and Animation Units, respectively. Thus, the term $\mathbf{S}\,\sigma$ accounts for shape variability (inter-person variability) while the term $\mathbf{A}\,\alpha$ accounts for the facial animation (intra-person variability).

Acquiring the user's personalized face model requires the estimation of the vector σ since the static shape only depends on σ. To this end, either feature-based or featureless approaches can be used.

We stress the fact that for the task of head tracking, the parameter vector α is not computed, instead it is set to a fixed value, e.g, the zero vector. Thus, in a local coordinate system the 3D model is described by the static shape $\mathbf{g}_s = \overline{\mathbf{g}} + \mathbf{S}\,\sigma$.

The adopted projection model is the weak perspective projection model. We neglect the perspective effects since the depth variation of the face can be very small compared to its absolute depth. Therefore, the mapping between the 3D

face model and the image is given by a 2×4 matrix encapsulating both the 3D head pose and the camera parameters. More details about the used face model can be found in [6].

The state of the model is given by the six degrees of freedom associated with the 3D head pose. This is given by the vector **b** (3 rotations and 3 translations):

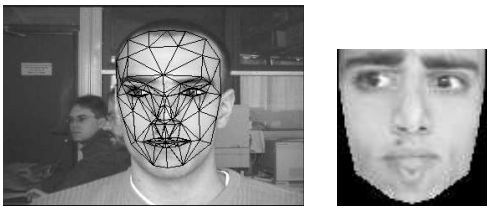
$$\mathbf{b} = [\theta_x, \quad \theta_y, \quad \theta_z, \quad \lambda\, t_x, \quad \lambda\, t_y, \quad \lambda\, t_z]^T \qquad (2)$$

2.2. Geometrically normalized facial images

A face texture is represented as a geometrically normalized image (shape-free texture). The geometry of this image is obtained by projecting the standard shape $\overline{\mathbf{g}}$ (wireframe) using a standard 3D pose (frontal view) onto an image with a given resolution. This geometry is represented by a triangular 2D mesh. The texture of this geometrically normalized image is obtained by texture mapping from the triangular 2D mesh in the input image using a piece-wise affine transform. Mathematically, the warping process applied to an input image **y** is denoted by:

$$\mathbf{x} = \mathcal{W}(\mathbf{y}, \mathbf{b}) \qquad (3)$$

where **x** denotes the shape-free texture and **b** denotes the 3D head pose. $\mathcal{W}$ is the piece-wise affine transform. Figure 1 illustrates the warping process applied to an input image. Two resolution levels have been used for the shape-free textures, 1300 and 5392 pixels.



(a) **(b)**

Fig. 1. **(a)** An input image with correct adaptation. **(b)** The corresponding geometrically normalized image.

2.3. Statistical facial texture model

Using Principal Component Analysis (PCA), the texture of any geometrically normalized image is a linear combination of a set of Texture Units or geometrically normalized eigenfaces. Thus, a texture **x** is given by:

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{X}\,\xi + \gamma \qquad (4)$$

where $\overline{\mathbf{x}}$ is the mean texture, the columns of **X** are the texture modes (the first M eigenvectors), ξ is the vector of texture parameters, and γ accounts for the reconstruction error. Thus, the reconstruction error is given by:

$$e = \|\gamma\|^2$$

3. 3D TRACKING BY A PARTICLE FILTER

Given a video sequence depicting a moving face, the tracking consists in estimating the 3D pose of the face, i.e. the vector $\mathbf{b}_t$ at frame t.

Particle filtering (or Sequential Monte Carlo algorithm) is an inference process which can be considered as a generalization of the Kalman filter. It aims at estimating the unknown state $\mathbf{b}_t$ from a set of noisy observations (images), $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \cdots, \mathbf{y}_t\}$ arriving in a sequential fashion. Two important components of this approach are the state transition and observation models whose most general forms can be given by:

$$\textit{State transition model} \quad \mathbf{b}_t = F_t(\mathbf{b}_{t-1}, \mathbf{U}_t) \qquad (5)$$

$$\textit{Observation model} \quad \mathbf{y}_t = G_t(\mathbf{b}_t, \mathbf{V}_t) \qquad (6)$$

where $\mathbf{U}_t$ is the system noise, F_t is the kinematics, $\mathbf{V}_t$ is the observation noise, and G_t models the observer. The particle filter approximates the posterior distribution $p(\mathbf{b}_t|\mathbf{y}_{1:t})$ by a set of weighted particles $\{\mathbf{b}_t^{(j)}, w_t^{(j)}\}_{j=1}^{J}$. Each element $\mathbf{b}_t^{(j)}$ represents the hypothetical state of the object and $w_t^{(j)}$ is the corresponding discrete probability.

Then, the state estimate can be set to the minimum mean square error (MMSE) estimate or the maximum a posteriori (MAP).

We use the following simple state transition model:

$$\mathbf{b}_t = \mathbf{b}_{t-1} + \mathbf{U}_t \qquad (7)$$

In this model, $\mathbf{U}_t$ is a random vector having a centred normal distribution, $\mathcal{N}(\mathbf{0}, \mathbf{C})$. The covariance matrix **C** is learned off-line from the state vector differences $\mathbf{b}_t - \mathbf{b}_{t-1}$ associated with previously tracked video sequences.

Since image data **y** are represented as shape-free textures **x** (the warped texture), we can set the observation likelihood $p(\mathbf{y}_t|\mathbf{b}_t)$ to $p(\mathbf{x}_t|\mathbf{b}_t)$.

The observation likelihood $p(\mathbf{x}_t|\mathbf{b}_t)$ quantifies the consistence of the texture $\mathbf{x}(\mathbf{b}_t)$ with the statistical texture model represented by the texture modes. For this purpose, we use a likelihood measure such the one proposed in [7]:

$$p(\mathbf{x}_t|\mathbf{b}_t) = c\, \exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{\xi_i^2}{\lambda_i^2}\right)\, \exp\left(-\frac{e}{2\rho^\star}\right) \qquad (8)$$

where e is the reconstruction error, λ_is are the eigenvalues associated with the first M eigenvectors, and $\rho^\star$ is the arithmetic average of the remaining eigenvalues.

This likelihood measure takes into account two distances (i) the DFFS (distance-from-feature-space), and (ii) the DIFS (distance-in-feature-space). Maximizing this likelihood is equivalent to minimizing the *Mahalanobis* distance over the original textures.

The particle filter algorithm proceeds as follows:

Initialize *a sample set* $\mathcal{S}_0 = \{\mathbf{b}_0^{(j)}, 1\}_{j=1}^{J}$ *according to some prior distribution* $p(\mathbf{b}_0)$.
For $t = 1, 2, \cdots$

For $j = 1, 2, \cdots, J$

Resample $\mathcal{S}_0$ *to obtain a new sample* $(\mathbf{b}_{t-1}^{\prime(j)}, 1)$.

Predict *the sample* $\mathbf{b}_t^{(j)}$ *from* $\mathbf{b}_{t-1}^{\prime(j)}$ *by drawing* $\mathbf{U}_t^{(j)}$ *and computing according to Eq. (7).*

Compute *the geometrically normalized texture* $\mathbf{x}(\mathbf{b}_t^{(j)})$ *according to Eq. (3).*

Update *the weight using* $w_t^{(j)} = p(\mathbf{x}_t|\mathbf{b}_t^{(j)})$ according to Eq. (8).

End

Normalize *the weights* $w_t^{(j)} = w_t^{(j)} / \sum_{i=1}^{J} w_t^{(i)}$; $j = 1, 2, \ldots, J$
End

During filtering, due to the resampling step, samples with a high weight may be chosen several times while others with relatively low weights may not be chosen at all.

4. TRACKING WITH A DETERMINISTIC TEXTURE MODEL

In the previous Section, the filtering process utilizes a likelihood measure based on a statistical model of facial texture (texture modes) which is built off-line using a training set of facial images. The disadvantage of such models is that whenever the environment changes, one should recompute the facial texture model associated with the new environment.

On the other hand, on-line appearance models release such constraints (e.g. [8]). Also, they can be generalized in the sense that they can handel many classes of objects (not only faces).

In this section, we show that a particle-filter-based tracking is still possible using a deterministic likelihood measure. The components of the tracking algorithm remain the same as in Section 3 except that the likelihood measure becomes a deterministic one. In our work, we use the following likelihood measure:

$$p(\mathbf{x}_t|\mathbf{b}_t) = \frac{1+\rho}{2} \tag{9}$$

where ρ is the normalized cross correlation coefficient between the patch $\mathbf{x}_t(\mathbf{b}_t)$ and the current appearance model $\mathbf{A}_t$. $\mathbf{A}_t$ represents the stable component of the previous warped patches under a sliding temporal window. This component is similar to the stable component introduced in [8]. In addition, we adopt three view-based face patches ($\mathbf{A}_{tl}$: left, $\mathbf{A}_{tc}$: center, $\mathbf{A}_{tr}$: right) since the 3D face orientation may vary significantly around a vertical axis. Texture patches are updated according to the yaw angle (vertical rotation) which is tracked over time. Notice that the appearance models are updated according to:

$$\mathbf{A}_t = (1-\lambda)\,\mathbf{A}_{t-1} + \lambda\,\mathbf{x}(\mathbf{b}_{t-1}^{*}) \tag{10}$$

where $\mathbf{b}_{t-1}^{*}$ is the MAP solution associated with the frame $t-1$. The update rate λ can be chosen experimentally like most incremental algorithms. With this updating scheme, the old information stored in the model decays exponentially over time.

5. TRACKING RESULTS

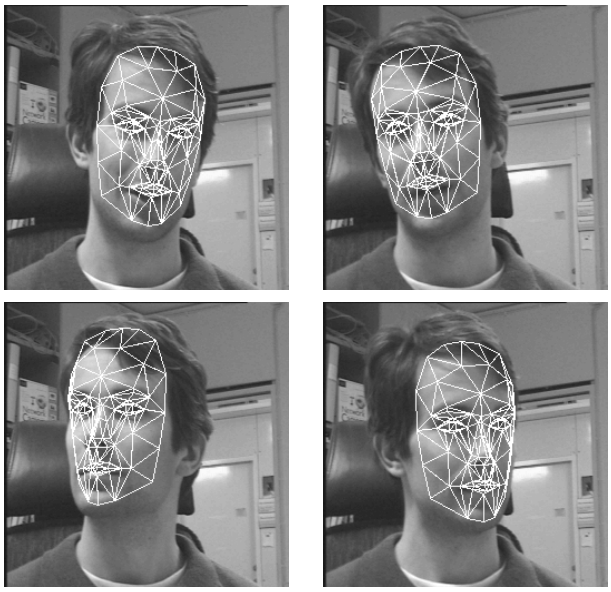


Fig. 2. Particle-filter-based 3D head tracking with a statistical facial texture model.

Figure 2 displays the tracking results associated with several frames of a long test sequence. It corresponds to the particle-filter-based tracking algorithm with a statistical texture model. The number of particles is set to 300. The statistical facial texture is built with 330 training face images [6].

Figure 3 displays the tracking results associated with several frames of another test sequence. It corresponds to the particle-filter-based tracking algorithm with an on-line appearance (a deterministic likelihood measure). The number of particles is set to 800.

Figure 4 shows other tracking results based on another test sequence of 140 frames. The top of this figure displays the tracking results when the 3D head pose is computed with a conventional feature-based RANSAC (RANdom SAmpling Consensus) technique [9] using two consec-

Fig. 3. Particle-filter-based 3D head tracking with an online appearance model.

utive images together with the 3D model of the face. Note that the features in the old image are the projection of the model vertices and the set of 3D-to-2D correspondences is recovered by searching for the best matches in the current image. The bottom of the figure displays the results of applying our particle-filter-based tracking method (see Section 3) to the same sequence. For both methods, the adaptation is displayed for frames 10 and 139. As can be seen, the RANSAC-based tracking suffers from some drifting due to the 3D model inaccuracies which has not occurred with our particle-filter-based tracking method.

Frame 10 Frame 139

Fig. 4. Top: applying the RANSAC technique to a video sequence of 140 frames. Bottom: applying the particle-filter-based method with an active appearance model.

6. CONCLUSION

We have developed two approaches for 3D head tracking based on particle filters. The first one utilizes a statistical texture model and the second one utilizes a deterministic texture model that is built on-line. Preliminary 3D head tracking results are successful despite the presence of facial actions. Currently, we are investigating appearance-adaptive models as well as tracking both 3D face pose and facial actions.

Acknowledgment

The authors thank Jörgen Ahlberg for providing the facial texture model.

7. REFERENCES

[1] A. Doucet, N. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, 2001.

[2] M. Isard and A. Black, "Contour tracking by stochastic propogation of conditional density," in *Proc. European Conference on Computer Vision*, 1996.

[3] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Computer Vision and Image Understinding*, vol. 91, no. 1-2, pp. 214–245, 2003.

[4] H. Nait-Cherif and S.J. McKenna, "Head tracking and action recognition in a smart meeting room," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, March 2003.

[5] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Object tracking with an adaptive color-based particle filter," in *Symposium for Pattern Recognition of the DAGM*, September 2002.

[6] F. Dornaika and J. Ahlberg, "Face and facial feature tracking using deformable models," *International Journal of Image and Graphics*, July 2004.

[7] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.

[8] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[9] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communication ACM*, vol. 24, no. 6, pp. 381–395, 1981.