Improving Appearance-Based 3D Face Tracking Using Sparse Stereo Data*

Fadi Dornaika and Angel D. Sappa

Computer Vision Center
Edifici O Campus UAB
08193 Bellaterra, Barcelona, Spain
{dornaika,sappa}@cvc.uab.es

Abstract. Recently, researchers proposed deterministic and statistical appearance-based 3D head tracking methods which can successfully tackle the image variability and drift problems. However, appearance-based methods dedicated to 3D head tracking may suffer from inaccuracies since these methods are not very sensitive to out-of-plane motion variations. On the other hand, the use of dense 3D facial data provided by a stereo rig or a range sensor can provide very accurate 3D head motions/poses. However, this paradigm requires either an accurate facial feature extraction or a computationally expensive registration technique (e.g., the Iterative Closest Point algorithm). In this paper, we improve our appearance-based 3D face tracker by combining an adaptive appearance model with a robust 3D-to-3D registration technique that uses sparse stereo data. The resulting 3D face tracker combines the advantages of both appearance-based trackers and 3D data-based trackers while keeping the CPU time very close to that required by real-time trackers. We provide experiments and performance evaluation which show the feasibility and usefulness of the proposed approach.

Keywords: 3D face tracking, adaptive appearance models, evaluation, stereo, robust 3D registration.

1 Introduction

The ability to detect and track human heads and faces in video sequences is useful in a great number of applications, such as human-computer interaction and gesture recognition. There are several commercial products capable of accurate and reliable 3D head position and orientation estimation (e.g., the acoustic tracker system Mouse [www.vrdepot.com/vrteclg.htm]). These are either based on magnetic sensors or on special markers placed on the face; both practices are encumbering, causing discomfort and limiting natural motion. Vision-based 3D head tracking provides an attractive alternative since vision sensors are not invasive and hence natural motions can be achieved (Moreno et al., 2002). However, detecting and tracking faces in video sequences is a challenging task.

Recently, deterministic and statistical appearance-based 3D head tracking methods have been proposed and used by some researchers (Cascia et al., 2000; Ahlberg, 2002;

* This work was supported by the MEC project TIN2005-09026 and The Ramón y Cajal Program.

J. Braz et al. (Eds.): VISAPP and GRAPP 2006, CCIS 4, pp. 354–366, 2007.

Matthews and Baker, 2004). These methods can successfully tackle the image variability and drift problems by using deterministic or statistical models for the global appearance of a special object class: the face. However, appearance-based methods dedicated to full 3D head tracking may suffer from some inaccuracies since these methods are not very sensitive to out-of-plane motion variations. On the other hand, the use of dense 3D facial data provided by a stereo rig or a range sensor can provide very accurate 3D face motions. However, computing the 3D face motions from the stream of dense 3D facial data is not straightforward. Indeed, inferring the 3D face motion from the dense 3D data needs an additional process. This process can be the detection of some particular facial features in the range data/images from which the 3D head pose can be inferred. For example, in (Malassiotis and Strintzis, 2005), the 3D nose ridge is detected and then used for computing the 3D head pose. Alternatively, one can perform a registration between 3D data obtained at different time instants in order to infer the relative 3D motions. The most common registration technique is the Iterative Closest Point (ICP) (Besl and McKay, 1992). The ICP algorithm and its variants can provide accurate 3D motions but their significant computational cost prohibits real-time performance.

The main contribution of this paper is a robust 3D face tracker that combines the advantages of both appearance-based trackers and 3D data-based trackers while keeping the CPU time very close to that required by real-time trackers. First, the 3D head pose is recovered using an appearance registration technique. Second, the obtained 3D head pose is utilized and refined by robustly registering two 3D point sets where one set is provided by stereo reconstruction.

The remainder of this paper proceeds as follows. Section 2 introduces our deformable 3D facial model. Section 3 states the problem we are focusing on, and describes the online adaptive appearance model. Section 4 summarizes the adaptive appearance-based tracker that tracks in real-time the 3D head pose and some facial actions. Section 5 gives some evaluation results associated with the appearance-based tracker. Section 6 describes the improvement step based on a robust 3D-to-3D registration and the appearance model. Section 7 gives some experimental results.

2 Modeling Faces

2.1 A Deformable 3D Model

In our study, we use the 3D face model *Candide*. This 3D deformable wireframe model was first developed for the purpose of model-based image coding and computer animation. The 3D shape of this wireframe model is directly recorded in coordinate form. It is given by the coordinates of the 3D 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 $\mathbf{g}$ is written as:

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\,\tau_{\mathbf{a}} \tag{1}$$

where $\mathbf{g}_s$ is the static shape of the model, $\tau_{\mathbf{a}}$ the animation control vector, and the columns of $\mathbf{A}$ are the Animation Units. In this study, we use six modes for the facial Animation Units (AUs) matrix $\mathbf{A}$. Without loss of generality, we have chosen the six

following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer and outer eyebrow raiser. These AUs are enough to cover most common facial animations (mouth and eyebrow movements). Moreover, they are essential for conveying emotions.

In equation (1), the 3D shape is expressed in a local coordinate system. However, one should relate the 3D coordinates to the image coordinate system. To this end, we adopt the weak perspective projection model. We neglect the perspective effects since the depth variation of the face can be considered as small compared to its absolute depth. Thus, the state of the 3D wireframe model is given by the 3D head pose parameters (three rotations and three translations) and the internal face animation control vector $\tau_{\mathbf{a}}$. This is given by the 12-dimensional vector **b**:

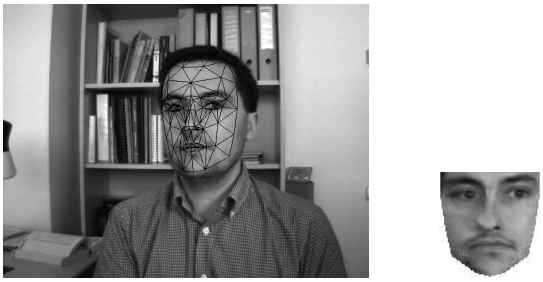
$$\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, {\tau_{\mathbf{a}}}^T]^T \tag{2}$$

2.2 Shape-Free Facial Patches

A face texture is represented as a shape-free texture (geometrically normalized image). The geometry of this image is obtained by projecting the static shape $\mathbf{g}_s$ using a centered frontal 3D pose onto an image with a given resolution. The texture of this geometrically normalized image is obtained by texture mapping from the triangular 2D mesh in the input image (see figure 1) using a piece-wise affine transform, $\mathcal{W}$. The warping process applied to an input image **y** is denoted by:

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

where **x** denotes the shape-free texture patch and **b** denotes the geometrical parameters. Several resolution levels can be chosen for the shape-free textures. The reported results are obtained with a shape-free patch of 5392 pixels. Regarding photometric transformations, a zero-mean unit-variance normalization is used to partially compensate for contrast variations. The complete image transformation is implemented as follows: (i) transfer the texture **y** using the piece-wise affine transform associated with the vector **b**, and (ii) perform the grey-level normalization of the obtained patch.



(a) **(b)**

Fig. 1. (a) an input image with correct adaptation. **(b)** the corresponding shape-free facial image.

3 Problem Formulation

Given a video sequence depicting a moving head/face, we would like to recover, for each frame, the 3D head pose and the facial actions encoded by the control vector $\tau_{\mathbf{a}}$. In other words, we would like to estimate the vector $\mathbf{b}_t$ (equation 2) at time t given all the observed data until time t, denoted $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. In a tracking context, the model parameters associated with the current frame will be handed over to the next frame.

For each input frame $\mathbf{y}_t$, the observation is simply the warped texture patch (the shape-free patch) associated with the geometric parameters $\mathbf{b}_t$. We use the HAT symbol for the tracked parameters and textures. For a given frame t, $\hat{\mathbf{b}}_t$ represents the computed geometric parameters and $\hat{\mathbf{x}}_t$ the corresponding shape-free patch, that is,

$$\hat{\mathbf{x}}_t = \mathbf{x}(\hat{\mathbf{b}}_t) = \mathcal{W}(\mathbf{y}_t, \hat{\mathbf{b}}_t) \tag{4}$$

The estimation of $\hat{\mathbf{b}}_t$ from the sequence of images will be presented in the next Section.

The appearance model associated with the shape-free facial patch at time t, A_t, is time-varying on that it models the appearances present in all observations $\hat{\mathbf{x}}$ up to time $(t-1)$. We assume that the appearance model A_t obeys a Gaussian with a center μ and a variance σ. Notice that μ and σ are vectors composed of d components/pixels (d is the size of $\mathbf{x}$) that are assumed to be independent of each other. In summary, the observation likelihood at time t is written as

$$p(\mathbf{y}_t|\mathbf{b}_t) = p(\mathbf{x}_t|\mathbf{b}_t) = \prod_{i=1}^{d} \mathbf{N}(x_i; \mu_i, \sigma_i) \tag{5}$$

where $\mathbf{N}(x; \mu_i, \sigma_i)$ is the normal density:

$$\mathbf{N}(x; \mu_i, \sigma_i) = (2\pi\sigma_i^2)^{-1/2} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2\right] \tag{6}$$

We assume that A_t summarizes the past observations under an exponential envelop, that is, the past observations are exponentially forgotten with respect to the current texture. When the appearance is tracked for the current input image, *i.e.* the texture $\hat{\mathbf{x}}_t$ is available, we can compute the updated appearance and use it to track in the next frame.

It can be shown that the appearance model parameters, *i.e.*, μ and σ can be updated using the following equations (see (Jepson et al., 2003) for more details on Online Appearance Models):

$$\mu_{t+1} = (1-\alpha)\,\mu_t + \alpha\,\hat{\mathbf{x}}_t \tag{7}$$

$$\sigma_{t+1}^2 = (1-\alpha)\,\sigma_t^2 + \alpha\,(\hat{\mathbf{x}}_t - \mu_t)^2 \tag{8}$$

In the above equations, all μ's and σ^2's are vectorized and the operation is element-wise. This technique, also called recursive filtering, is simple, time-efficient and therefore, suitable for real-time applications. The appearance parameters reflect the most recent observations within a roughly $L = 1/\alpha$ window with exponential decay.

Note that μ is initialized with the first patch $\hat{\mathbf{x}}_0$. In order to get stable values for the variances, equation (8) is not used until the number of frames reaches a given value (*e.g.*, the first 40 frames). For these frames, the classical variance is used, that is, equation (8) is used with α being set to $\frac{1}{t}$.

Here we used a single Gaussian to model the appearance of each pixel in the shape-free patch. However, modeling the appearance with Gaussian mixtures can also be used on the expense of some additional computational load (e.g., see (Zhou et al., 2004; Lee, 2005)).

4 Tracking Using Adaptive Appearance Registration

We consider the state vector $\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, {\tau_\mathbf{a}}^T]^T$ encapsulating the 3D head pose and the facial actions. In this section, we will show how this state can be recovered for time t from the previous known state $\hat{\mathbf{b}}_{t-1}$ and the current input image $\mathbf{y}_t$.

The sought geometrical parameters $\mathbf{b}_t$ at time t are related to the previous parameters by the following equation ($\hat{\mathbf{b}}_{t-1}$ is known):

$$\mathbf{b}_t = \hat{\mathbf{b}}_{t-1} + \Delta\mathbf{b}_t \tag{9}$$

where $\Delta\mathbf{b}_t$ is the unknown shift in the geometric parameters. This shift is estimated using a region-based registration technique that does not need any image feature extraction. In other words, $\Delta\mathbf{b}_t$ is estimated such that the warped texture will be as close as possible to the facial appearance A_t. For this purpose, we minimize the *Mahalanobis* distance between the warped texture and the current appearance mean,

$$\min_{\mathbf{b}_t} e(\mathbf{b}_t) = \min_{\mathbf{b}_t} D(\mathbf{x}(\mathbf{b}_t), \mu_t) = \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \tag{10}$$

The above criterion can be minimized using iterative first-order linear approximation which is equivalent to a Gauss-Newton method. It is worthwhile noting that the minimization is equivalent to maximizing the likelihood measure given by (5). Moreover, the above optimization is carried out using Huber function (Dornaika and Davoine, 2004). In the above optimization, the gradient matrix $\frac{\partial \mathcal{W}(\mathbf{y}_t, \mathbf{b}_t)}{\partial \mathbf{b}_t} = \frac{\partial \mathbf{x}_t}{\partial \mathbf{b}_t}$ is computed for each frame and is approximated by numerical differences similarly to the work of Cootes (Cootes et al., 2001).

On a 3.2 GHz PC, a non-optimized C code of the approach computes the 3D head pose and the six facial actions in 50 ms. About half that time is required if one is only interested in computing the 3D head pose parameters.

5 Accuracy Evaluation

The monocular tracker described above provides the time-varying 3D head pose (especially the out-of-plane parameters) with some inaccuracies whose magnitude depends on several factors such as the absolute depth of the head, the head orientation, and the

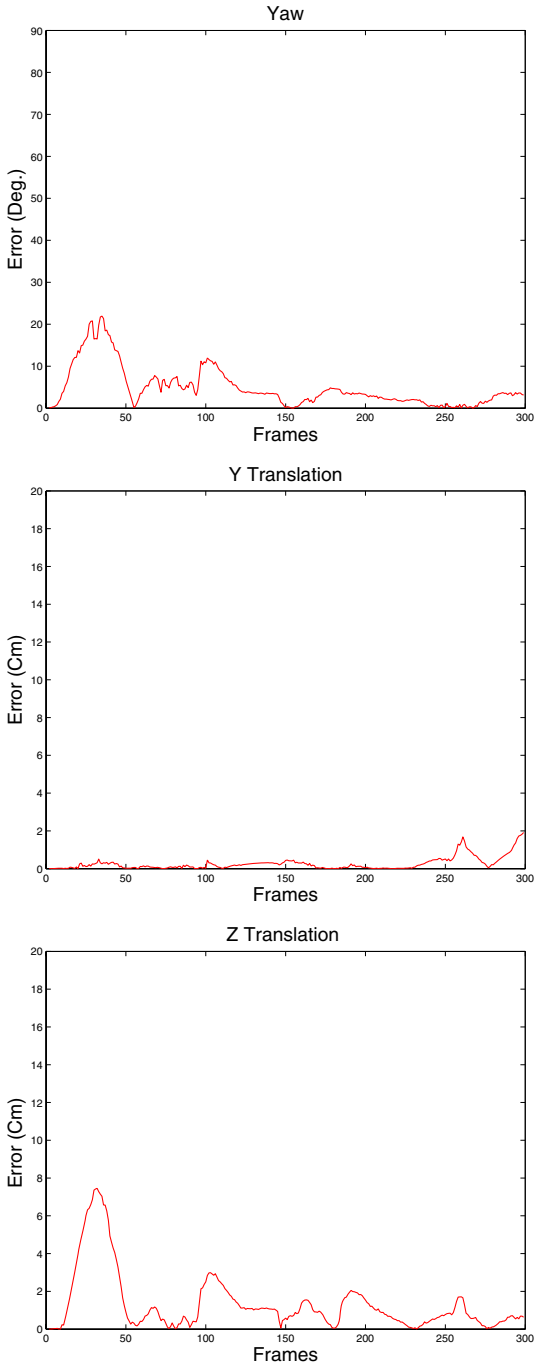


Fig. 2. 3D head pose errors computed by the ICP algorithm associated with a 300-frame long sequence

camera parameters. We have evaluated the accuracy of the above proposed monocular tracker using ground truth data that were recovered by the Iterative Closest Point algorithm (Besl and McKay, 1992) and dense 3D facial data.

Figure 2 depicts the monocular tracker errors associated with a 300-frame long sequence which contains rotational and translational out-of-plane head motions. The nominal absolute depth of the head was about 65 cm, and the focal length of the camera was 824 pixels. As can be seen, the out-of-plane motion errors can be large for some frames for which there is a room for improvement. Moreover, this evaluation has confirmed the general trend of appearance-based trackers, that is, the out-of-plane motion parameters (pitch angle, yaw angle, and depth) are more affected by errors than the other parameters. More details about accuracy evaluation can be found in (Dornaika and Sappa, 2005).

One expects that the monocular tracker accuracy can be improved if an additional cue is used. In our case, the additional cue will be the 3D data associated with the mesh vertices provided by stereo reconstruction. Although the use of stereo data may seem as an excess requirement, recall that cheap and compact stereo systems are now widely available (e.g., `[www.ptgrey.com]`).

We point out that there is no need to refine the facial feature motions obtained by the above appearance-based tracker since their independent motion can be accurately recovered. Indeed, these features (the lips and the eyebrows) have different textures, so their independent motion can be accurately recovered by the appearance-based tracker.

6 Improving the 3D Head Pose

The improved 3D face tracker is outlined in Figure 3. The remainder of this section describes the improvement steps based on sparse stereo-based 3D data. Since the monocular tracker provides the 3D head pose by matching the input texture with the adaptive facial texture model (both textures correspond to a 2D mesh), it follows that the out-of-plane motion parameters can be inaccurate even when most of the facial features project onto their true location in the image. We use this fact to argue that the appearance-based tracker will greatly help in the sense that it will provide the putative set of 3D-to-3D correspondences through the 2D projections. Our basic idea is to start from the 3D head pose provided by the monocular tracker and then improve it by using some sparse 3D data provided by stereo reconstruction. Here we use the estimated six degrees of freedom as well as the corresponding projection of all vertices. The estimated 3D head pose will be used for mapping the 3D mesh in 3D space while the 2D projections of the vertices will be processed by the stereo system in order to get their 3D coordinates.

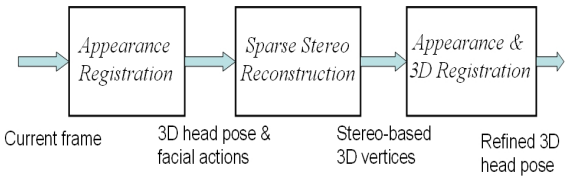


Fig. 3. The main steps of the developed robust 3D face tracker

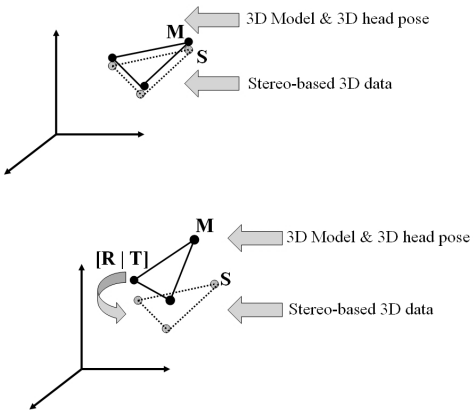


Fig. 4. (**M**) A 3D facial patch model positioned using the estimated 3D head pose. (**S**) the same 3D patch (three vertices) provided by stereo reconstruction. **Top:** An ideal case where the appearance-based 3D head pose corresponds to the true 3D head pose. **Bottom:** A real case where the appearance-based 3D head pose does not exactly correspond to the true 3D head pose. It follows that the improvement is simply the rigid 3D displacement $[\mathbf{R}|\mathbf{T}]$ that aligns the two sets of vertices.

Improving the 3D head pose is then carried out by combining a robust 3D-to-3D registration and the appearance model. The robust 3D registration uses the 3D mesh vertices (the 3D model is mapped with the estimated 3D head pose) and the corresponding 3D coordinates provided by the stereo rig while the appearance model is always given by (10). Recall that the stereo reconstruction only concerns the image points resulting from projecting the 3D mesh vertices onto the image. Since our 3D mesh contains about one hundred vertices the whole refinement step will be fast.

Figure 4 illustrates the basic idea that is behind the improvement step, namely the robust 3D registration. Figure 4 (Top) illustrates an ideal case where the estimated appearance-based 3D head pose corresponds to the true 3D pose. In this case, the vertices of the 3D mesh after motion compensation coincide with their corresponding 3D points provided by the stereo rig. Figure 4 (Bottom) illustrates a real case where the estimated appearance-based 3D head pose does not correspond exactly to the true one. In this case, the improvement can be estimated by recovering the 3D rigid displacement $[\mathbf{R}|\mathbf{T}]$ between the two sets of vertices.

We point out that the set of vertex pairs may contain some outliers caused for instance by occluded vertices. Thus, the 3D registration process must be robust. Robust 3D registration methods have been proposed in recent literature (e.g., see (Chetverikov et al., 2005; Fitzgibbon, 2003)). In our work, we use a RANSAC-like technique that computes an adaptive threshold for outlier detection. The whole improvement algorithm is outlined in Figure 5. As can be seen, the final solution (see the second paragraph in Figure 5) takes into account two criteria: i) the 3D-to-3D registration, and ii) the adaptive appearance model.

Inlier detection. The question now is: Given a subsample k and its associated solution $\mathbf{D}_k$, How do we decide whether or not an arbitrary vertex is an inlier? In techniques

Random sampling: Repeat the following three steps K times

1. Draw a random subsample of 3 different pairs of vertices. We have three pairs of 3D points $\{\mathbf{M}_i \leftrightarrow \mathbf{S}_i\}, \quad i = 1, 2, 3$.
2. For this subsample, indexed by k ($k = 1, \ldots, K$), compute the 3D rigid displacement $\mathbf{D}_k = [\mathbf{R}_k|\mathbf{T}_k]$, where $\mathbf{R}_k$ is a 3D rotation and $\mathbf{T}_k$ a 3D translation, that brings these three pairs into alignment. $\mathbf{R}_k$ and $\mathbf{T}_k$ are computed by minimizing the residual error $\sum_{i=1}^{3} |\mathbf{S}_i - \mathbf{R}_k \mathbf{M}_i - \mathbf{T}_k|^2$. This is carried out using the quaternion method (Horn, 1987).
3. For this solution $\mathbf{D}_k$, compute the median M_k of the squared residual errors with respect to the whole set of N vertices. Note that we have N residuals corresponding to all vertices $\{\mathbf{M}_j \leftrightarrow \mathbf{S}_j\}, \quad j = 1, \ldots, N$. The squared residual associated with an arbitrary vertex $\mathbf{M}_j$ is $|\mathbf{S}_j - \mathbf{R}_k \mathbf{M}_j - \mathbf{T}_k|^2$.

Solution:

1. For each solution $\mathbf{D}_k = [\mathbf{R}_k|\mathbf{T}_k], k = 1, \ldots, K$, compute the number of inliers among the entire set of vertices (see text). Let n_k be this number.
2. Select the 10 best solutions, i.e. the solutions that have the highest number of inliers.
3. Refine each such solution using all its inlier pairs.
4. For these 10 solutions, compute the corresponding observation likelihood (5).
5. Choose the solution that has the largest observation likelihood.

Fig. 5. Recovering the 3D rigid displacement using robust statistics and the appearance

dealing with 2D geometrical features (points and lines) (Fischler and Bolles, 1981), this is achieved using the distance in the image plane between the actual location of the feature and its mapped location. If this distance is below a given threshold then this feature is considered as an inlier; otherwise, it is considered as an outlier. Here we can do the same by manually defining a distance in 3D space. However, this fixed selected threshold cannot accommodate all cases and all noises. Therefore, we use an adaptive threshold distance that is computed from the residual errors associated with all subsamples. Our idea is to compute a robust estimation of standard deviation of the residual errors. In the exploration step, for each subsample k, the median of residuals was computed. If we denote by $\overline{M}$ the least median among all K medians, then a robust estimation of the standard deviation of the residuals is given by (Rousseeuw and Leroy, 1987):

$$\hat{\sigma} = 1.4826 \left[1 + \frac{5}{N-3}\right] \sqrt{\overline{M}} \tag{11}$$

where N is the number of vertices. Once $\hat{\sigma}$ is known, any vertex j can be considered as an inlier if its residual error satisfies $|r_j| < 3\,\hat{\sigma}$.

Computational cost. On a 3.2 GHz PC, a non-optimized C code of the robust 3D-to-3D registration takes on average 15 ms assuming that the number of random samples K is set to 20 and the total number of the 3D mesh vertices, N, is 113. This computational time includes both the stereo reconstruction and the robust technique outlined in Figure 5. Thus, by appending the robust 3D-to-3D registration to the appearance-based tracker (described before) a video frame can be processed in about 70 ms.

7 Experimental Results

Figure 6 displays the head and facial action tracking results associated with a 300-frame-long sequence (only four frames are shown). The tracking results were obtained using the adaptive appearance described in Sections 4. The upper left corner of each image shows the current appearance (μ_t) and the current shape-free texture $(\hat{\mathbf{x}}_t)$. In this sequence, the nominal absolute depth of the head was about 80 cm.

As can be seen, the tracking results indicate good alignment between the mesh model and the images. However, it is very difficult to evaluate the accuracy of the out-of-plane motions by only inspecting the projection of the 3D wireframe onto these 2D images.

Therefore, we have used ground truth data for the 3D head pose parameters associated with a video sequence similar to the one shown Figure 6. The ground truth data are recovered by means of 3D registration between dense 3D facial clouds using the Iterative Closest Point algorithm. Figure 7 displays an accuracy comparison between the appearance-based tacker and the improved tracker using ground-truth data. The solid curves correspond to the errors obtained with the appearance-based tracker, and the dashed ones correspond to those obtained with the developed approach including the robust 3D-to-3D registration technique. The top plot corresponds to the pitch angle, the middle plot to the vertical translation, and the bottom plot to the in-depth translation. As can be seen, the most significant improvement affects the in-depth translation. The noisy value of the pitch angle error could be explained by the fact the 3D rotation (improvement) is estimated from a small set of 3D points. However, on average the value of the obtained error is equal to or less than the error obtained with the appearance-based tracker.

8 Conclusion

In this paper, we have proposed a robust 3D face tracker that combines the advantages of both appearance-based trackers and 3D data-based trackers while keeping the CPU time very close to that required by real-time trackers. Experiments on real video sequences indicate that the estimates of the out-of-plane motions of the head can be considerably improved by combining a robust 3D-to-3D registration with the appearance model.

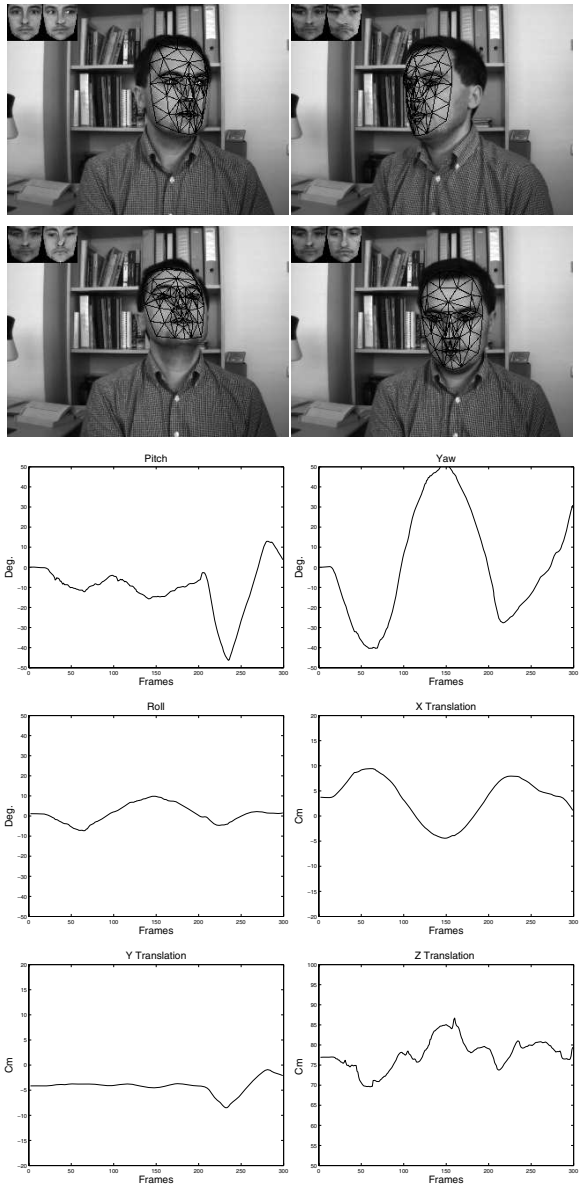


Fig. 6. Tracking the 3D head pose with appearance-based tracker. The sequence length is 300 frames. Only frames 38, 167, 247, and 283 are shown. The six plots display the six degrees of freedom of the 3D head pose as a function of time.

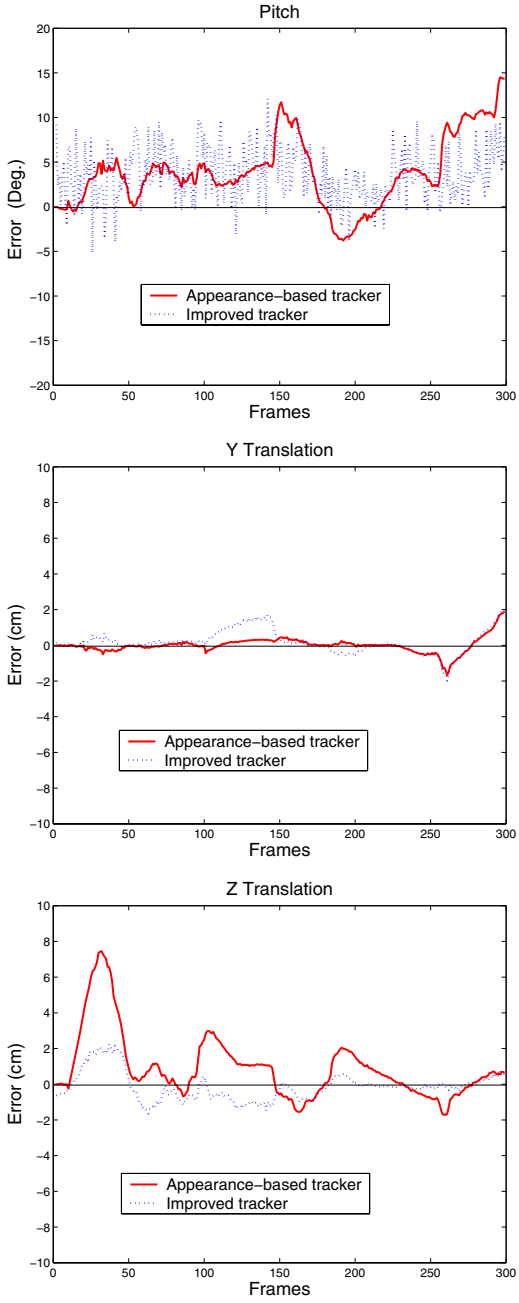


Fig. 7. 3D head pose errors associated with the sequence as a function of the frames. From top to bottom: pitch angle error, vertical translation error, and in-depth translation error. The solid curves display the errors obtained with the appearance-based tracker, and the dashed ones display those obtained with the improved tracker.

References

Ahlberg, J.: An active model for facial feature tracking. EURASIP Journal on Applied Signal Processing 2002(6), 566–571 (2002)

Besl, P., McKay, N.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)

Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(4), 322–336 (2000)

Chetverikov, D., Stepanov, D., Kresk, P.: Robust Euclidean alignment of 3D point sets: the trimmed iterative closet point algorithm. Image and Vision Computing 23, 299–309 (2005)

Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 681–684 (2001)

Dornaika, F., Davoine, F.: Head and facial animation tracking using appearance-adaptive models and particle filters. In: IEEE Workshop on Real-Time Vision for Human-Computer Interaction, Washington DC, IEEE, Los Alamitos (2004)

Dornaika, F., Sappa, A.: Appearance-based tracker: An evaluation study. In: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE, Los Alamitos (2005)

Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communication ACM 24(6), 381–395 (1981)

Fitzgibbon, A.: Robust registration of 2D and 3D point sets. Image and Vision Computing 21, 1145–1153 (2003)

Horn, B.: Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Amer. A. 4(4), 629–642 (1987)

Jepson, A., Fleet, D., El-Maraghi, T.: Robust online appearance models for visual tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10), 1296–1311 (2003)

Lee, D.: Effective Gaussian mixture learning for video background subtraction. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(5), 827–832 (2005)

Malassiotis, S., Strintzis, M.G.: Robust real-time 3D head pose estimation from range data. Pattern Recognition 38(8), 1153–1165 (2005)

Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision 60(2), 135–164 (2004)

Moreno, F., Tarrida, A., Andrade-Cetto, J., Sanfeliu, A.: 3D real-time tracking fusing color histograms and stereovision. In: IEEE International Conference on Pattern Recognition, IEEE, Los Alamitos (2002)

Rousseeuw, P., Leroy, A.: Robust Regression and Outlier Detection. John Wiley & Sons, New York (1987)

Zhou, S., Chellappa, R., Mogghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Transactions on Image Processing 13(11), 1473–1490 (2004)