Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 2116–2126
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Rigid and non-rigid face motion tracking by aligning texture maps and stereo 3D models

Fadi Dornaika [a,*], Angel D. Sappa [b]

[a] Institut Géographique National, 2 avenue Pasteur, 94165 Saint-Mandé, France
[b] Computer Vision Center, Edifici O, Campus UAB, 08193 Bellaterra, Barcelona, Spain

Received 6 July 2006; received in revised form 18 April 2007
Available online 4 July 2007

Communicated by P. Bhattacharya

Abstract

Accurate rigid and non-rigid tracking of faces is a challenging task in computer vision. Recently, appearance-based 3D face tracking methods have been proposed. These methods can successfully tackle the image variability and drift problems. However, they may fail to provide accurate out-of-plane face motions since they are not very sensitive to out-of-plane motion variations. In this paper, we present a framework for fast and accurate 3D face and facial action tracking. Our proposed framework retains the strengths of both appearance and 3D data-based trackers. We combine an adaptive appearance model with an online stereo-based 3D model. We provide experiments and performance evaluation which show the feasibility and usefulness of the proposed approach.

Keywords: Face tracking; Facial action tracking; Adaptive appeareance models; Image registration; 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 due to the image variability caused by pose, expression, and illumination changes.

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

* Corresponding author. Tel.: +33 1 43988429; fax: +33 1 43988581.
E-mail addresses: fadi.dornaika@ign.fr (F. Dornaika), asappa@cvc.uab.es (A.D. Sappa).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.06.011

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) algorithm. This algorithm and its variants can provide accurate 3D motions but their significant computational cost prohibits real-time performance. Moreover, this algorithm is intended for registering rigid objects.

Classical 3D face tracking algorithms that are based on 3D facial features are subject to drift problems. Moreover, these algorithms cannot compute the facial actions due, for instance, to facial expressions.

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. In our work, we use the *Candide* deformable 3D model (Ahlberg, 2001) which is a simple model embedding non-rigid facial motion using the concept of facial actions. Our proposed framework for tracking faces in videos can be summarized as follows. First, the 3D head pose and some facial actions are estimated from the monocular image by registering the warped input texture with a shape-free facial texture map. Second, based on these current parameters the 2D locations of the mesh vertices are inferred by projecting the current mesh onto the current video frame. Then the 3D coordinates of these vertices are computed by stereo reconstruction. Third, the relative 3D face motion is then obtained using a robust 3D-to-3D registration technique between two meshes corresponding to the first video frame and the current video frame, respectively. Our framework attempts to reduce the number of outlier vertices by deforming the meshes according to the same current facial actions and by exploiting the symmetrical shape of the 3D mesh.

The resulting 3D face and facial action tracker is accurate, fast, and drift insensitive. Moreover, unlike many proposed frameworks (e.g., Xiao et al., 2004), it does not require any learning stage since it is based on online facial appearances and online stereo 3D data.

The remainder of the paper proceeds as follows. Section 2 introduces our deformable 3D facial model. Section 3 states the problem we are focusing on, and summarizes the appearance-based monocular tracker that tracks in real-time the 3D head pose and some facial actions. It gives some evaluation results. Section 4 describes a robust 3D-to-3D registration that combines the monocular tracker's results and the stereo-based reconstructed vertices. Section 5 gives some experimental results.

2. Modeling faces

In this section, we briefly describe our deformable face model and explain how to produce a shape-free facial texture map.

2.1. A deformable 3D model

As mentioned before, we use the *Candide* 3D face model. 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 $\boldsymbol{P}_i$, $i = 1, \ldots, n$ where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector $\boldsymbol{g}$; the concatenation of the 3D coordinates of all vertices $\boldsymbol{P}_i$. The vector $\boldsymbol{g}$ is written as

$$\boldsymbol{g} = \boldsymbol{g}_s + \boldsymbol{A}\boldsymbol{\tau}_a \tag{1}$$

where $\boldsymbol{g}_s$ is the static shape of the model, $\boldsymbol{\tau}_a$ the animation control vector, and the columns of $\boldsymbol{A}$ are the animation units. In this study, we use six modes for the facial animation units (AUs) matrix $\boldsymbol{A}$. Without loss of generality, we have chosen the six following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lower 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 Eq. (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 $\boldsymbol{\tau}_a$. This is given by the 12-dimensional vector $\boldsymbol{b}$

$$\boldsymbol{b} = [\theta_x, \quad \theta_y, \quad \theta_z, \quad t_x, \quad t_y, \quad t_z, \quad \boldsymbol{\tau}_a^{\mathrm{T}}]^{\mathrm{T}}. \tag{2}$$

2.2. Shape-free facial texture maps

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 $\boldsymbol{g}_s$ 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 Fig. 1) using a piece-wise affine transform, $\mathscr{W}$. The warping process applied to an input image $\boldsymbol{y}$ is denoted by

$$\boldsymbol{x}(\boldsymbol{b}) = \mathscr{W}(\boldsymbol{y}, \boldsymbol{b}) \tag{3}$$

where $\boldsymbol{x}$ denotes the shape-free texture map and $\boldsymbol{b}$ 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

Fig. 1. (a) An input image with correct adaptation and (b) the corresponding shape-free facial texture map.

implemented as follows: (i) transfer the texture $\boldsymbol{y}$ using the piece-wise affine transform associated with the vector $\boldsymbol{b}$, and (ii) perform the gray-level normalization of the obtained patch.

3. Tracking by aligning facial texture maps

3.1. Problem formulation

Given a monocular 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 $\boldsymbol{\tau_a}$. In other words, we would like to estimate the vector $\boldsymbol{b}_t$ (Eq. (2)) at time t given all the observed data until time t, denoted $\boldsymbol{y}_{1:t} \equiv \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$. In a tracking context, the model parameters associated with the current frame will be handed over to the next frame.

In (Dornaika and Davoine, 2006), we have developed a fast method to compute this state from the previous known state $\hat{\boldsymbol{b}}_{t-1}$ and the current input image $\boldsymbol{y}_t$. An overview of this method is presented in this section.

3.2. Facial texture model

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

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

The estimation of $\hat{\boldsymbol{b}}_t$ from the sequence of images will be presented in Section 3.3.

By assuming that the pixels within the shape-free patch are independent, we can model the facial appearance using a multivariate Gaussian with a diagonal covariance matrix $\boldsymbol{\Sigma}$. In other words, this multivariate Gaussian is the distribution of the facial texture maps $\hat{\boldsymbol{x}}_t$. Let $\boldsymbol{\mu}$ be the Gaussian center and $\boldsymbol{\sigma}$ the vector containing the square root of the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are d-vectors (d is the size of $\boldsymbol{x}$). In summary, the observation likelihood at time t is written as

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

where $\boldsymbol{N}(x_i;\mu_i,\sigma_i)$ is a normal density:

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

We assume that the appearance model summarizes the past observations under an exponential envelope, 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 $\hat{\boldsymbol{x}}_t$ 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., the μ_i's and σ_i's can be updated from time t to time $(t+1)$ using the following equations (see Jepson et al., 2003 for more details on Online Appearance Models):

$$\mu_{i_{(t+1)}} = (1-\alpha)\mu_{i_{(t)}} + \alpha\hat{x}_{i_{(t)}} \tag{7}$$

$$\sigma^2_{i_{(t+1)}} = (1-\alpha)\sigma^2_{i_{(t)}} + \alpha(\hat{x}_{i_{(t)}} - \mu_{i_{(t)}})^2. \tag{8}$$

3.3. Tracking

We consider the state vector $\boldsymbol{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau_a}^{\mathrm{T}}]^{\mathrm{T}}$ encapsulating the 3D head pose and the facial actions. The sought geometrical parameters $\boldsymbol{b}_t$ at time t are estimated using a region-based registration technique that does not need any image feature extraction. For this purpose, we minimize the *Mahalanobis* distance between the warped texture and the current appearance mean—the current Gaussian center $\boldsymbol{\mu}_t$

$$\min_{\boldsymbol{b}_t} D(\boldsymbol{x}(\boldsymbol{b}_t), \boldsymbol{\mu}_t) = \min_{\boldsymbol{b}_t} \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2 \tag{9}$$

The above criterion can be minimized using a Gauss–Newton method where the initial solution is given by the previous known state $\hat{\boldsymbol{b}}_{t-1}$. It is worth noting that the minimization is equivalent to maximizing the likelihood measure given by (5). In the above optimization, the gradient matrix $\frac{\partial \mathscr{W}(\boldsymbol{y}_t, \boldsymbol{b}_t)}{\partial \boldsymbol{b}_t} = \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{b}_t}$ is computed for each frame and is approximated by numerical differences similarly to the work of 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.

3.4. Accuracy evaluation

In (Dornaika and Sappa, 2005), we have evaluated the accuracy of the above proposed monocular tracker. To this end, we have used ground truth data that were recovered by the iterative closest point algorithm (Besl and McKay, 1992)

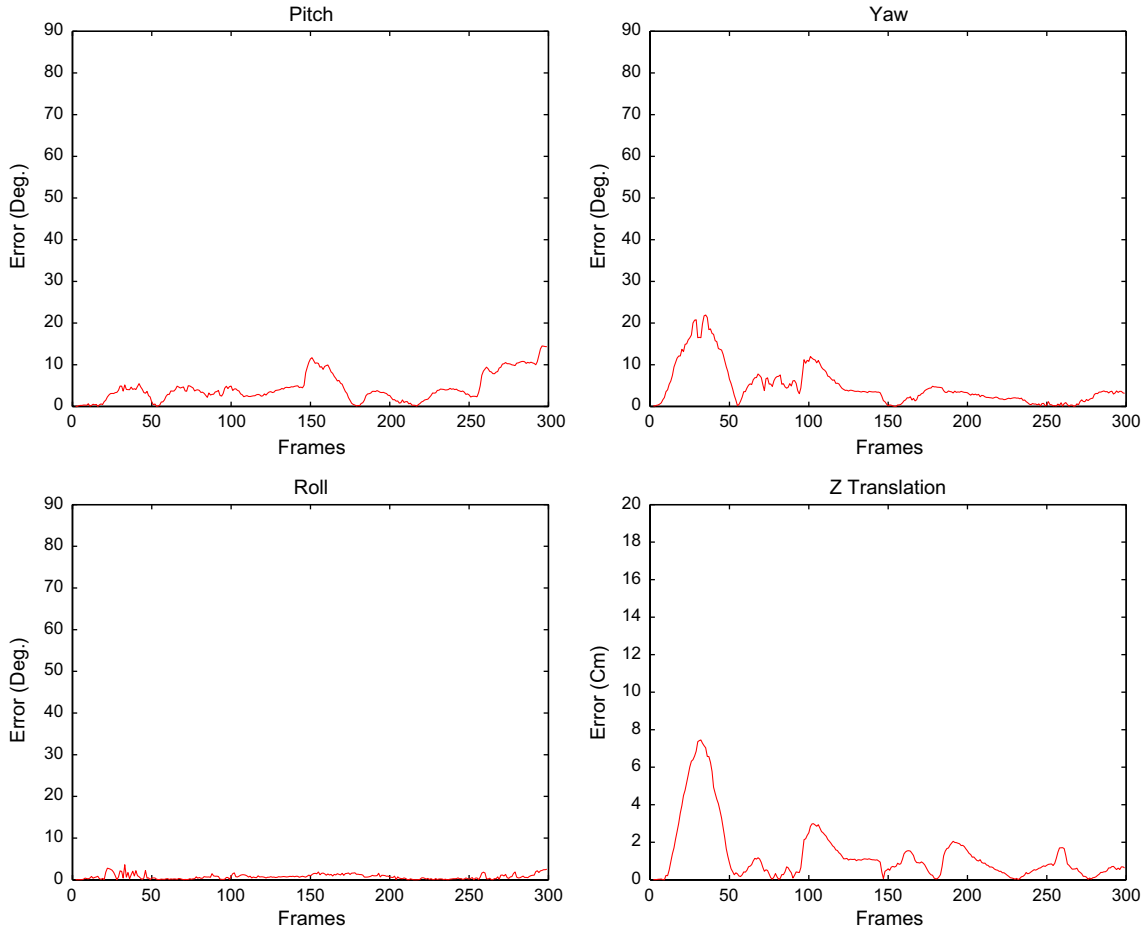

Fig. 2. 3D face motion errors computed by the ICP algorithm associated with a 300-frame sequence.

and dense 3D facial data. Fig. 2 depicts the monocular tracker errors associated with a 300-frame 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. We point out that the facial feature motions obtained by the above appearance-based tracker can be accurately recovered. Indeed, these features (the lips and the eyebrows) have specific textures, so their independent motion can be accurately recovered by the appearance-based tracker.

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] and [www.videredesign.com]). We point out that stereo data are used to refine the static model $\boldsymbol{g}_s$ in the sense that the facial mesh can be more person-specific. Moreover, in our developed framework the stereo matching and reconstruction only concern the vertices of the 3D mesh which corresponds to 0.036% of a 640×480 image.

4. Tracking by aligning texture maps and stereo-based 3D models

In this section, we propose a novel tracking scheme that aims at computing a fast and accurate 3D face motion. To this end, we exploit the tracking results provided by the appearance-based tracker (Section 3) and the availability of a stereo system for reconstructing the mesh vertices. The whole algorithm is outlined in Fig. 3. The three stages are applied to each video frame—stereo pair. Note that the facial actions are already computed using the monocular tracker described in Section 3.

Our approach to 3D face tracking is simple and can be stated as follows: *If the 3D coordinates of the 3D mesh vertices at two different time instants are given in the same coordinate system, then the rigid transform corresponding to the 3D face motion can easily be estimated using a robust 3D point-to-point registration algorithm.* Without loss of generality,

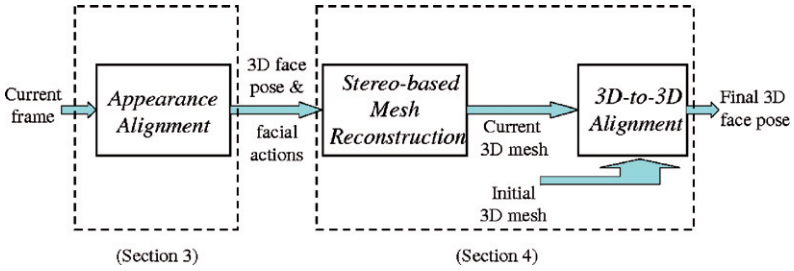

Fig. 3. The main steps of the developed robust 3D face tracker.

the 3D face motion will be expressed with respect to the head coordinate system associated with the first video frame. Recall that upgrading this relative 3D face motion to a 3D head pose, that is expressed in the camera coordinate system, is carried out using the 3D head pose associated with the first video frame. The latter transform can be inferred using a classical 3D pose estimation algorithm. This choice is adopted in order to obtain numerical stability for the 3D registration algorithms. In order to invoke the registration algorithm one has to compute the 3D coordinates of the vertices associated with the two different video frames: the initial video frame and the current one (see Fig. 4). This is carried out using stereo-based data associated with the first frame and the current frame. The proposed algorithm can be summarized as follows.

(1) Invoke the appearance-tracker to recover the current 3D head pose and facial actions (Section 3).
(2) Based on the estimated 3D head pose and facial actions, deform the 3D mesh and project it onto the current frame.
(3) Reconstruct the obtained image points—stereo reconstruction of the mesh vertices (the current frame).
(4) Deform the initial mesh (first frame) according to the current estimated facial actions. Thus, the possible non-rigid movement of the mesh is cancelled out.
(5) Eliminate the vertices that are not consistent with the 3D symmetry test. Recall that for a given person and given facial actions the Euclidean distance between two symmetrical vertices is invariant due to the 3D model symmetry. In Section 5, we will present a more sophisticated scheme that also exploits the 3D shape symmetry.
(6) Carry out a robust 3D registration between the two 3D meshes. The computed 3D rigid displacement corresponds to the actual 3D face motion between the initial frame and the current frame. This step is detailed in Table 1.

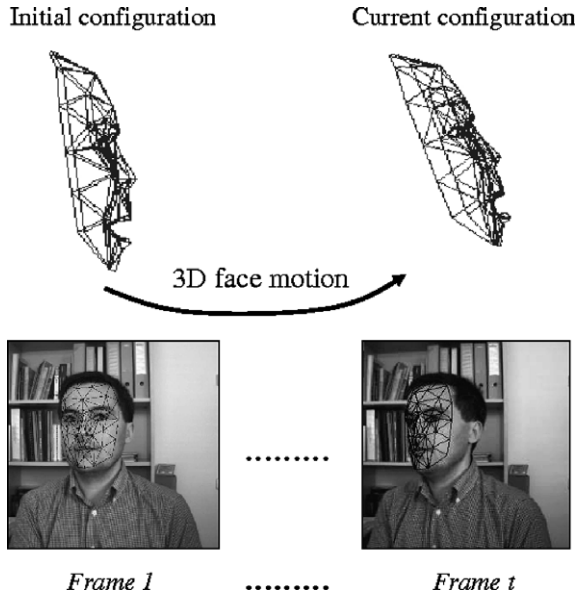

Fig. 4. The relative 3D face motion is recovered using a robust 3D-to-3D registration. Both meshes are expressed in the same coordinate system and use the same facial action parameters.

Table 1
Recovering the relative 3D face motion using online stereo data and robust statistics

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 $\{\boldsymbol{M}_i \leftrightarrow \boldsymbol{S}_i\}$, $i = 1,2,3$. $\boldsymbol{M}_i$ denotes the 3D coordinates of vertex i associated with the first frame, and $\boldsymbol{S}_i$ denotes the 3D coordinates of the same vertex with the current frame t. $\boldsymbol{M}_i$ and $\boldsymbol{S}_i$ are expressed in the same coordinate system
(2) For this subsample, indexed by k ($k = 1, \ldots, K$), compute the 3D rigid displacement $\boldsymbol{D}_k = [\boldsymbol{R}_k|\boldsymbol{T}_k]$, where $\boldsymbol{R}_k$ is a 3D rotation and $\boldsymbol{T}_k$ a 3D translation, that brings these three pairs into alignment. $\boldsymbol{R}_k$ and $\boldsymbol{T}_k$ are computed by minimizing the residual error $\sum_{i=1}^{3} |\boldsymbol{S}_i - \boldsymbol{R}_k \boldsymbol{M}_i - \boldsymbol{T}_k|^2$. This is carried out using the quaternion method (Horn, 1987)
(3) For this solution $\boldsymbol{D}_k$, 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 $\{\boldsymbol{M}_j \leftrightarrow \boldsymbol{S}_j\}, j = 1, \ldots, N$. The squared residual associated with an arbitrary vertex $\boldsymbol{M}_j$ is $|\boldsymbol{S}_j - \boldsymbol{R}_k \boldsymbol{M}_j - \boldsymbol{T}_k|^2$

Solution

(1) For each solution $\boldsymbol{D}_k = [\boldsymbol{R}_k|\boldsymbol{T}_k], k = 1, \ldots, K$, compute the number of inliers among the entire set of vertices (see text). Let n_k be this number
(2) Choose the solution that has the largest number of inlier vertices
(3) Refine the corresponding solution using all its inlier pairs

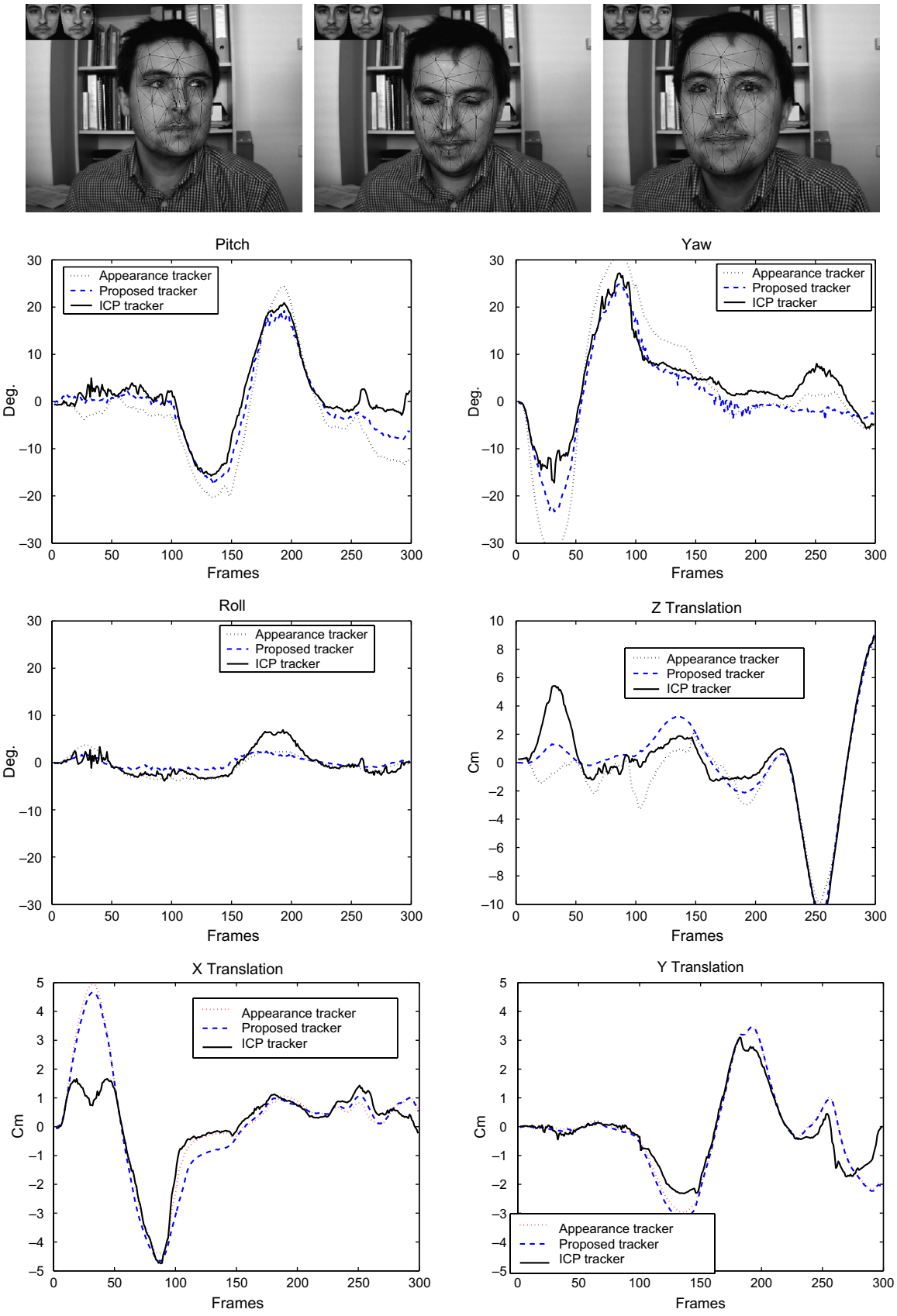

Fig. 5. *Top*: Tracking the face and facial actions in a 300-frame sequence using the proposed tracker. Only frames 22, 179, and 255 are shown. *Bottom*: The six degrees of freedom associated with the relative 3D face motion obtained with the three trackers. The dotted curves correspond to the appearance-based tracker, the dashed ones to the proposed framework, and the solid ones to the ICP algorithm.

As can be seen, the recovered 3D face motion has relied on both the appearance model and the stereo-based 3D data. Steps 4 and 5 have been introduced in order to reduce the number of outlier vertices. Reducing the number of outliers is very useful for obtaining a very fast robust registration in the sense that a few random samples are needed. In step 4, the initial mesh is deformed according to the current facial actions. Thus, a rigid registration technique can be efficiently applied. Recall that for a profile view the vertices associated with the hidden part of the face may have erroneous depth due to occlusion. Thus, step 5 eliminates the vertices with erroneous depth since they do not satisfy the symmetry constraint. Note that the symmetry test is invoked only for the cases where the face is not in a near frontal pose (this is known from the tracked yaw angle). Note that although the appearance-based tracker may provide slightly inaccurate out-of-plane parameters, the corresponding projected mesh onto the current image is still useful for getting the current stereo-based 3D coordinates of the mesh vertices (steps 2 and 3).

We stress the fact that the proposed approach is not similar to a classical stereo-based 3D tracker where feature points are tracked across the image sequence. In our method, there is no feature matching and tracking across the image sequence. Instead, the whole face appearance is tracked in order to accurately locate the facial features (the projection of the mesh vertices) from which the 3D coordinates are inferred.

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 inlier/outlier detection.

4.1. Inlier detection

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

$$\hat{\sigma} = 1.4826\left[1 + \frac{5}{N-3}\right]\sqrt{\overline{M}} \tag{10}$$

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 $|r_j| < 3\hat{\sigma}$.

4.2. Computational cost

On a 3.2 GHz PC, a non-optimized C code of the robust 3D-to-3D registration takes about 10 ms assuming that the number of random samples K is set to 8 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 Table 1. Thus, by appending the robust 3D-to-3D registration to the appearance-based tracker (described in Section 3) a video frame can be processed in about 60 ms.

5. Experimental results

We use the commercial stereo camera system Bumblebee from Point Grey (http://www.ptgrey.com). It consists of two Sony ICX084 color CCDs with 6 mm focal length lenses. The monocular sequence is used by the monocular tracker (Section 3), while the stereo sequence is used by the 3D-to-3D registration technique (Section 4). Fig. 5 (top) shows the face and facial action tracking results associated with a 300-frame sequence (only three frames are shown). The tracking results were obtained using the proposed framework described in Sections 3 and 4. The upper left corner of each image shows the current appearance ($\boldsymbol{\mu}_t$) and the current shape-free texture ($\hat{\boldsymbol{x}}_t$). In this sequence, the nominal absolute depth of the head was about 65 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 run three

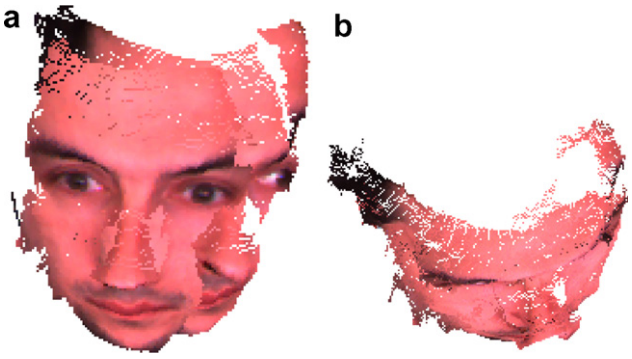

Fig. 6. 3D registration of two facial clouds provided at frames 1 and 39, which are separated by a large yaw angle (about 40°). (a) the range facial data associated to frames 1 and 39, expressed in the same coordinate system. (b) 3D registration using the iterative closet point algorithm.

different methods using the same video sequence. The first method is given by the appearance-based tracker (Section 3). The second method is given by the proposed framework (Sections 3 and 4). Note that the number of random samples used by the proposed method is set to 8. The third method is given by the ICP registration between dense facial surfaces where the facial surface model is set to the one obtained with the first stereo pair (Dornaika and Sappa, 2005). Since the ICP registration results are accurate we can use them as ground-truth data for the relative 3D face motions. Fig. 6 is an example showing how the 3D head pose/motion associated with frame 39 is estimated using the ICP algorithm. Fig. 6a illustrates the range facial data associated with frames 1 and 39, expressed in the same coordinate system. One can easily see that the face has performed a vertical rotation of about 40 degrees. Fig. 6b shows the 3D registration using the ICP algorithm.

The lower part of Fig. 5 shows the computed relative 3D face motions obtained with the three methods. The current relative 3D face motion is given by its rigid displacement

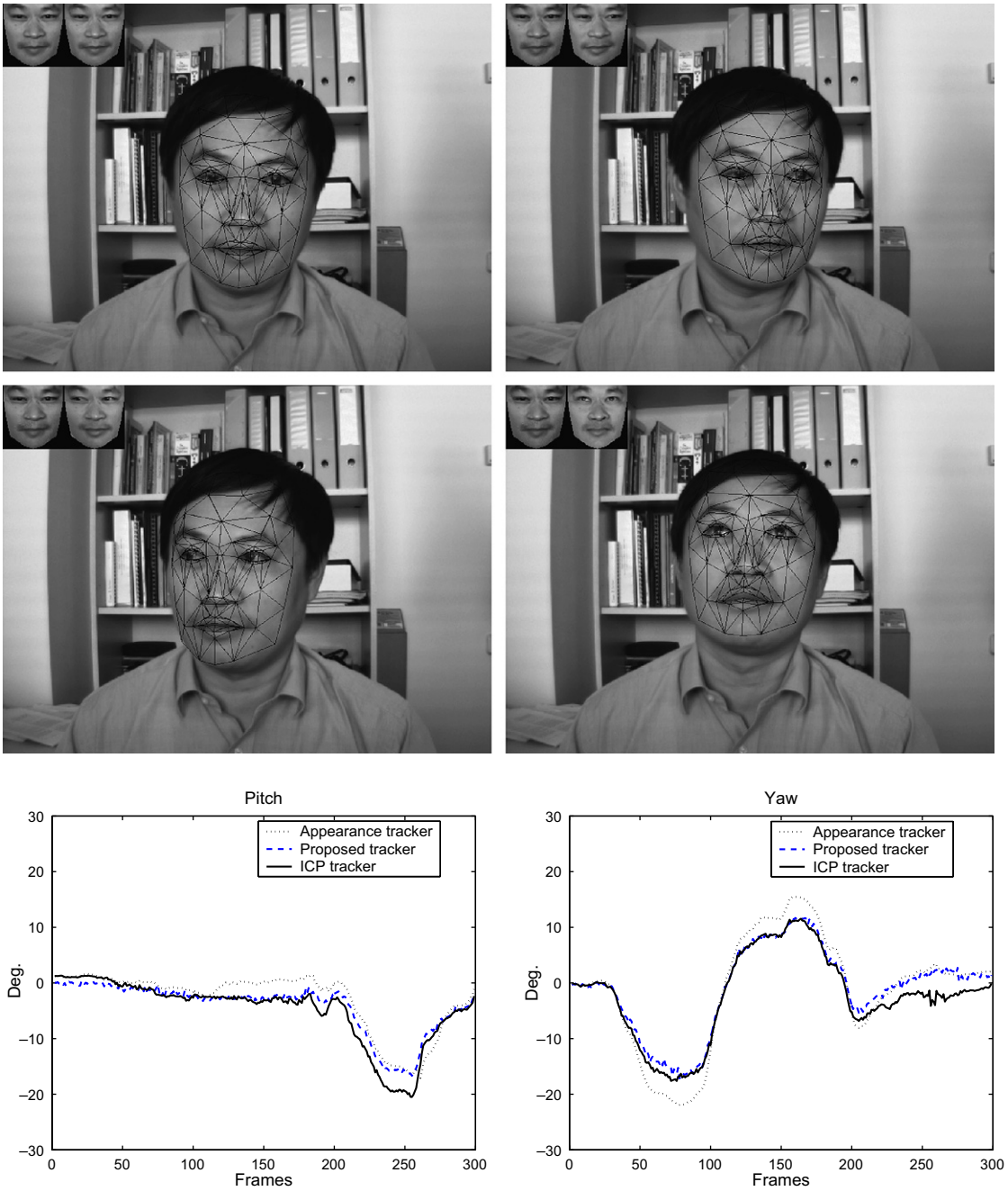

Fig. 7. *Top*: The relative 3D face motion associated with another video sequence. *Bottom*: The pitch and yaw angles associated with the relative 3D face motion obtained with the three trackers.

with respect to its initial pose. This figure shows the three angles and the three translations as a function of time. The dotted curves correspond to the appearance-based tracker, the dashed ones to the proposed framework, and the solid ones to the ICP algorithm. From these curves, we can see that the proposed framework has outperformed the appearance-based tracker since the curves become close to those computed by the ICP algorithm—the ground-truth data. In this case, the first facial surface used by the ICP algorithm contained about 20,000 3D points.

Fig. 7 shows the computed relative 3D face motions (pitch and yaw angles) obtained with another video sequence.

Since the ICP algorithm works with rigid surfaces, the faces depicted in the sequences of Figs. 5 and 7 were somehow neutral. Fig. 8 illustrates a video sequence depicting a person performing simultaneous head motions and facial expressions. Fig. 9 shows the computed relative 3D face motions (the three out-of-plane parameters plus the horizontal translation) associated with this sequence. For this

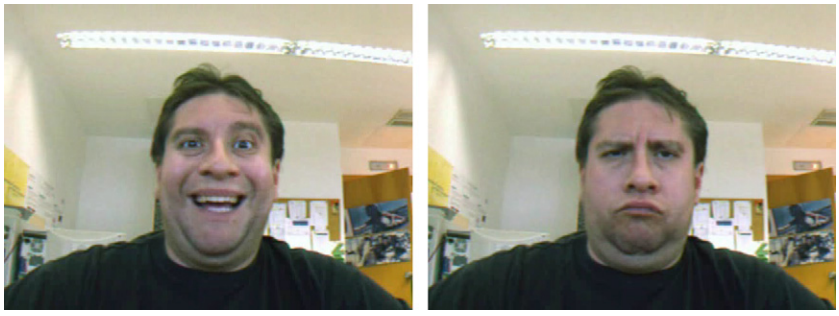

Fig. 8. A video sequence depicting a person performing simultaneous head motions and facial expressions.

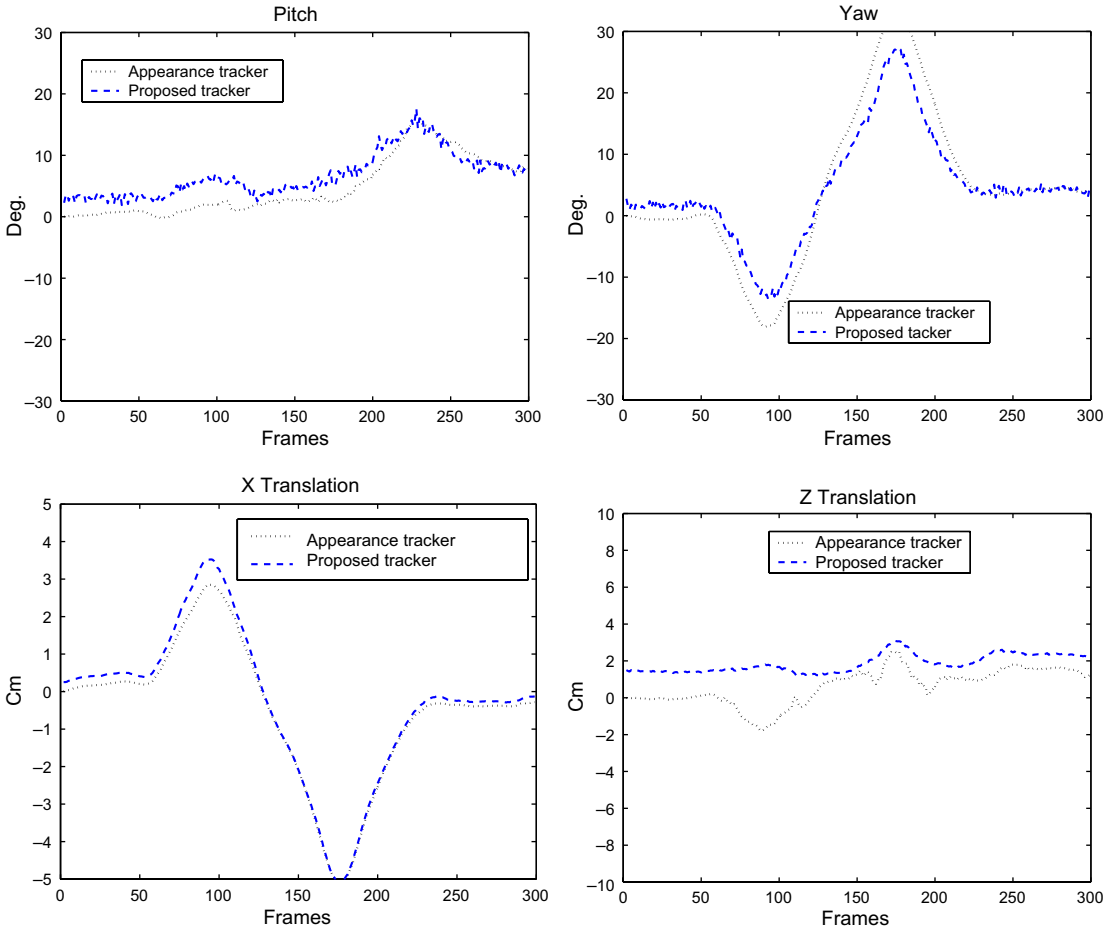

Fig. 9. The relative 3D face motion associated with the video sequence of Fig. 8.

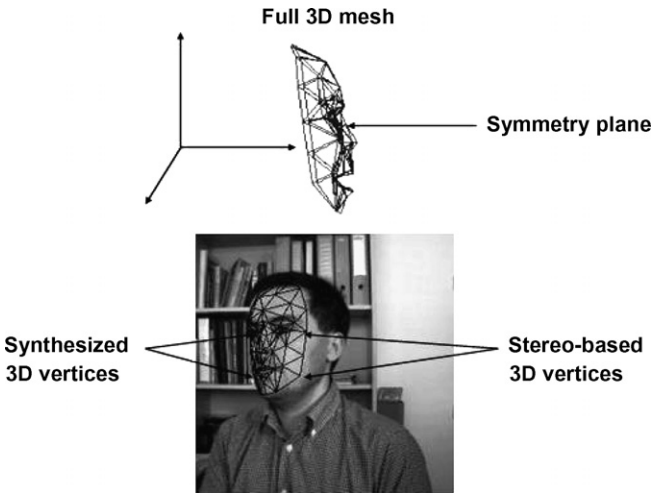

Fig. 10. The full 3D face mesh is reconstructed using both the stereo-based data (for the best exposed half) and the symmetrical shape property (for the other half).

sequence, we have not used the ICP algorithm since the facial surface undergoes a rigid and large non-rigid motion. As can be seen, in general the motion parameters have been improved by using online stereo data.

5.1. The use of a full 3D facial mesh

In this section, we propose an alternative to the use of the symmetry test mentioned earlier (step 5—the vertex elimination step). This scheme is also based on the shape symmetry of the 3D face model. This scheme can work for frames in which the face is in a profile view, and will replace step 5 in the proposed tracking algorithm. The proposed scheme synthesizes the half part of the face model that is not well exposed to the camera using the distances between symmetrical vertices and the orientation of the symmetry plane. Recall that this orientation is computed from the reconstructed vertices belonging to the symmetry plane. In theory, since the whole 3D mesh will be used by the 3D registration technique regardless of the face orientation, it is expected that the tracking results become more accurate than those obtained using only the non-occluded vertices. However, the possible inaccuracy affecting the symmetry plane orientation might limit this advantage. Fig. 10 illustrates the reconstruction of a full online 3D mesh using both the stereo data and the shape symmetry.

Fig. 11 shows the tracking results associated with the same video sequence depicted in Fig. 8 but the proposed tracker uses the whole 3D mesh. Again, the dotted curves

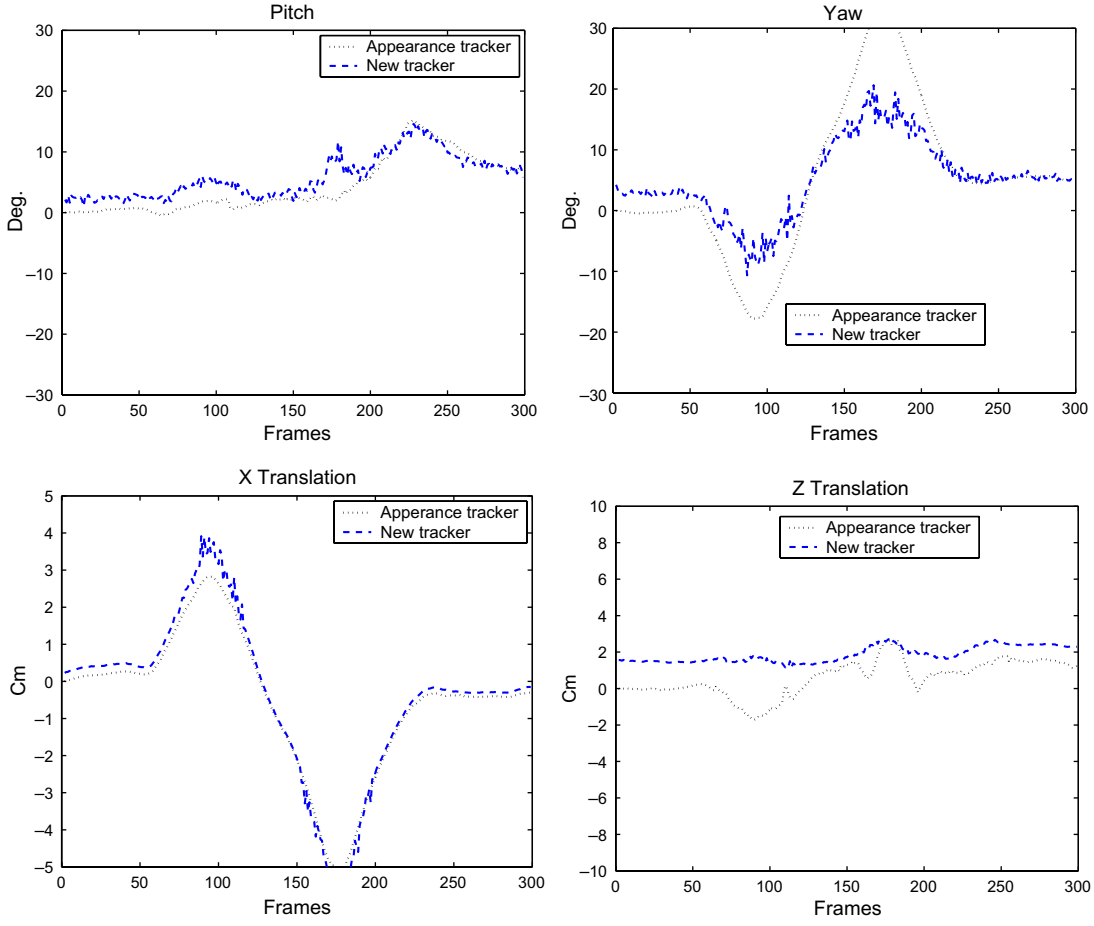

Fig. 11. The relative 3D face motion associated with the video sequence of Fig. 8 using a full 3D facial mesh.

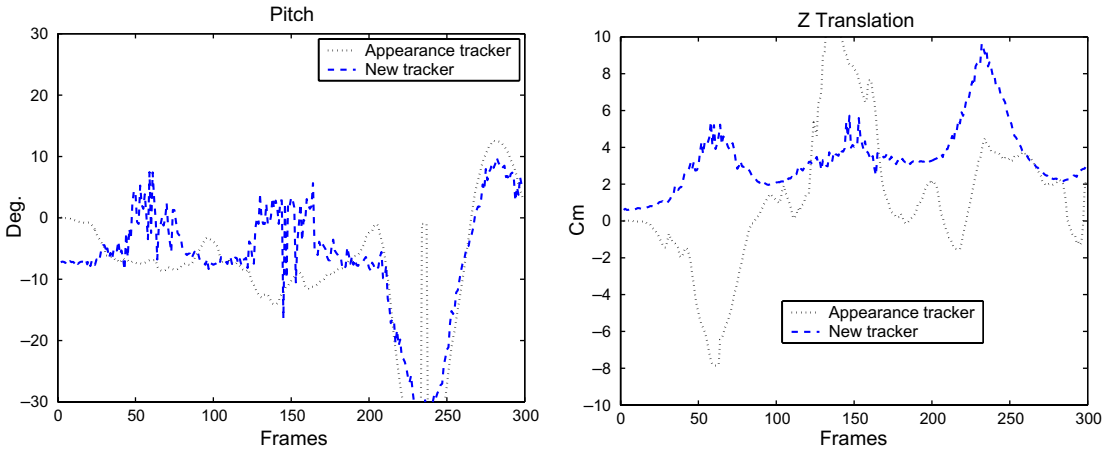

Fig. 12. The relative 3D face motion associated with the video sequence of Fig. 10 using a full 3D facial mesh.

correspond to the appearance-based tracker, and the dashed ones to the proposed framework—the New tracker. By comparing Figs. 9 and 11, one can see that there is a slight discrepancy associated with the yaw angle estimation (the dashed curves), which is due to the use of two different schemes exploiting the 3D shape symmetry. For this sequence, the absolute depth of the subject's face was about 65 cm.

Fig. 12 shows the tracking results associated with the video sequence depicted in Fig. 10. For this sequence, the absolute depth of the subject face was about 90 cm. As can be seen, by comparing Figs. 11 and 12, the gained accuracy provided by the proposed approach—which incorporates stereo data—will increase as the absolute depth increases.

6. Conclusion

In this paper, we have proposed a robust 3D face and facial action 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.

Although the joint use of 3D facial data and the ICP algorithm as a 3D head tracker could be attractive, the significant computational cost of the ICP algorithm prohibits real-time performance. Moreover, this algorithm is intended for registering rigid objects.

Acknowledgment

This work was supported in part by the MEC project TRA2004-06702/AUT and The Ramón y Cajal Program.

References

Ahlberg, J., 2001. CANDIDE-3 – an updated parametrized face. Technical Report LiTH-ISY-R-2326. Department of Electrical Engineering. Linköping University, Sweden.

Ahlberg, J., 2002. An active model for facial feature tracking. EURASIP J. Appl. Signal Process. 2002 (6), 566–571.

Besl, P., McKay, N., 1992. A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Machine Intell. 14 (2), 239–256.

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

Chetverikov, D., Stepanov, D., Kresk, P., 2005. Robust Euclidean alignment of 3D point sets: The trimmed iterative closet point algorithm. Image Vision Comput. 23, 299–309.

Cootes, T.F., Edwards, G.J., Taylor, C.J., 2001. Active appearance models. IEEE Trans. Pattern Anal. Machine Intell. 23 (6), 681–684.

Dornaika, F., Davoine, F., 2006. On appearance based face and facial action tracking. IEEE Trans. Circuits Systems Video Technol. 16 (9), 1107–1124.

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

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

Fitzgibbon, A.W., 2003. Robust registration of 2D and 3D point sets. Image Vision Comput. 21, 1145–1153.

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

Jepson, A.D., Fleet, D.J., El-Maraghi, T.F., 2003. Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Machine Intell. 25 (10), 1296–1311.

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

Matthews, I., Baker, S., 2004. Active appearance models revisited. Internat. J. Comput. Vision 60 (2), 135–164.

Moreno, F., Tarrida, A., Andrade-Cetto, J., Sanfeliu, A., 2002. 3D real-time tracking fusing color histograms and stereovision. In: IEEE Internat. Conf. on Pattern Recognition.

Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. John Wiley & Sons, New York.

Xiao, J., Baker, S., Matthews, I, Kanade, T., 2004. Real-time combined 2D + 3D active appearance models. In: IEEE Internat. Conf. on Computer Vision and Pattern Recognition.