Journal of Universal Computer Science, vol. 16, no. 6 (2010), 903-920
submitted: 15/9/09, accepted: 2/3/10, appeared: 28/3/10 © *J.UCS*

3D Head Pose and Facial Expression Tracking using a Single Camera

Lucas D. Terissi, Juan C. Gómez
(Laboratory for System Dynamics and Signal Processing
FCEIA, Universidad Nacional de Rosario
CIFASIS, CONICET, Rosario, Argentina
{terissi, gomez}@cifasis-conicet.gov.ar)

Abstract: Algorithms for 3D head pose and facial expression tracking using a single camera (monocular image sequences) is presented in this paper. The proposed method is based on a combination of feature-based and model-based approaches for pose estimation. A generic 3D face model, which can be adapted to any person, is used for the tracking. In contrast to other methods in the literature, the proposed method does not require a training stage. It only requires an image of the person's face to be tracked facing the camera to which the model is fitted manually through a graphical user interface. The algorithms were evaluated perceptually and quantitatively with two video databases. Simulation results show that the proposed tracking algorithms correctly estimate the head pose and facial expression, even when occlusions, changes in the distance to the camera and presence of other persons in the scene, occur. Both perceptual and quantitative results are similar to the ones obtained with other methods proposed in the literature. Although the algorithms were not optimized for speed, they run near real time. Additionally, the proposed system delivers separate head pose and facial expression information. Since information related with facial expression, which is represented only by six parameters, is independent from head pose information, the tracking algorithms could also be used for facial expression analysis and video-driven facial animation.

Key Words: Computer vision, Head pose tracking, Facial expression, 3D deformable models, Image processing
Category: I.4, I.4.8, I.2.10

1 Introduction

Three-dimensional head pose and facial expression detection and tracking in a video sequence have become important tasks in several computer vision applications, like video surveillance, human-computer interaction, biometrics, vehicle automation, etc. [Murphy-Chutorian and Trivedi, 2009], [Jimenez et al., 2008], [Pallejà et al., 2008]. Determining the 3D head position and orientation is also fundamental in the development of applications such as vision-driven user interfaces, robust facial expression/emotion analysis, face recognition and model-based image coding. Head and facial expression tracking algorithms have to deal with issues such as: significant head motion, changes in orientation or scale, partial face occlusion and changes in lighting conditions. To overcome these difficulties, set of colored markers on the face could be used [Busso and Narayanan, 2007], [Savrana et al., 2006], [Terissi and Gómez, 2007]. In this way, the tracking is simplified but the range of applications becomes very limited because a *make-up* stage is needed, which could be annoying and time-consuming.

Different approaches have been proposed in recent years for tracking moving objects in a scene. The different approaches proposed in the literature could be broadly classified as feature-based [Cristinacce and Cootes, 2006] or model-based [Dornaika and Ahlberg, 2006]. Feature-based approaches rely on tracking local regions of interest, like key points, curves, optical flow, or skin color. In the model-based approach, a 2D or 3D model of the object to be tracked is used to estimate the pose/location of the object in the scene. The model is projected onto the image in order to find correspondences between 3D and 2D object features. These features can be edges, line segments, points, etc. Then, the object-image correspondences are used to compute the pose of the object by matching the model to the tracked features. Several of these model-based algorithms need a training using several images of the object to be tracked in different positions [Li et al., 2007].

The head pose and facial expression tracking system presented in this paper uses both ideas. It make use of a generic 3D face model to determinate the image features to track at each frame of the image sequence as in feature-based approaches. Then, similarly to the case of model-based approaches, the pose is estimated by matching the model to the tracked features. In head pose estimation the 3D face model is considered as a rigid object. For estimating the current facial expression, an adaptation of the procedure to estimate the pose of the face is used, in this case the model is treated as a deformable object. The tracking system proposed in this paper does not require a training stage, requiring only an image of the person's face to be tracked facing the camera, that is used to manually fit the model to it. This fitting procedure is performed through a graphical user interface developed by the authors and it only takes a couple of minutes. To track the head pose and facial expression, online information (previous frames) and offline information (reference images) are considered. The use of offline information, as proposed in [Vacchetti et al., 2004], leads to more robust tracking against situations where face occlusion and jitter in the images are present.

The rest of this paper is organized as follows. In section 2 the generic 3D face model used in this work is described. The tracking algorithms proposed in this paper are presented in section 3 and the algorithms for pose and facial expression estimation are described in section 4. Section 5 shows some simulations and experimental results of the proposed algorithms. Finally, some conclusions are given in section 6.

2 Parameterized 3D Face Model

In this paper, the generic 3D face model *Candide-3* [Ahlberg, 2001] is used for face representation. This 3D face model was developed at Linköping University, Sweden, and it was widely used in computer graphics, computer vision and model-based image-coding applications. The shape of this wireframe face model is given by a set of 3D vertices and triangular patches, see Figure 1. The model defines a set of Shape Units and Animation Units to control the appearance of the face and to animate it, respectively [Ahlberg, 2001]. Shape Units indicate how to deform the model for the purposes of, for instance, changing the position of the eyes, nose and mouth, making the mouth wider, etc. Similarly, Animation Units are used to control the movements of the mouth, eyes, eyebrows, etc.

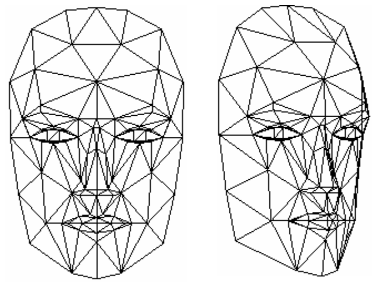


Figure 1: *Candide-3* model.

The 3D face model is fully described by a $3N$-vector $\mathbf{g}$ consisting of the concatenation of the 3D coordinates $\mathbf{g}_i$, $i = 1, ..., N$, of the N vertices of the model. Vector $\mathbf{g}$ can be written as

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{S}\boldsymbol{\sigma} + \mathbf{A}\boldsymbol{\alpha} \tag{1}$$

where $\bar{\mathbf{g}}$ is the mean shape of the model, and the columns of matrices $\mathbf{S}$ and $\mathbf{A}$ are the Shape and Animation Units, respectively. Vectors $\boldsymbol{\sigma}$ and $\boldsymbol{\alpha}$ contain the shape and animation parameters, respectively.

The pose of the face model is given by its position and orientation with respect to the camera. The pose is defined by six parameters, t_x, t_y and t_z, representing the model translation, and θ_x, θ_y and θ_z, representing the model rotation around three axes. Thus, the head position, facial appearance and facial expression of the 3D face model are given by the following set of parameters

$$\{t_x, t_y, t_z, \theta_x, \theta_y, \theta_z, \boldsymbol{\sigma}, \boldsymbol{\alpha}\} \tag{2}$$

2.1 Perspective Projection Model

The perspective projection of the 3D model onto the 2D image, *i.e.*, the mapping between 3D and 2D vertices, is given by a 3×4 projection matrix $\mathbf{T}$ defined as

$$\mathbf{T} = \mathbf{KP} \tag{3}$$

where $\mathbf{K}$ is the 3×3 camera calibration matrix that depends on the internal parameters of the camera such as focal length, skew coefficient, etc., and $\mathbf{P}$ is the 3×4 pose matrix that depends on θ_x, θ_y, θ_z, t_x, t_y and t_z. The internal parameters of the camera can be estimated by several methods proposed in the literature [Sturm and Maybank, 1999], [Zhang, 2000].

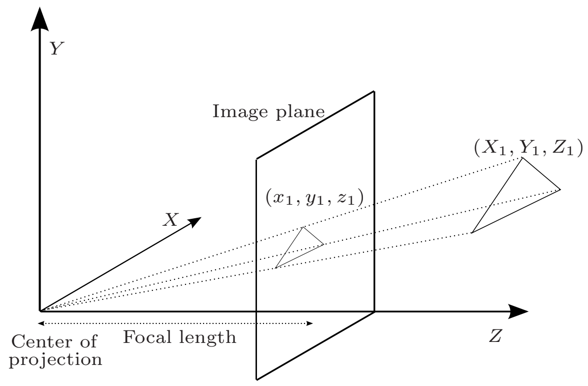


Figure 2: Perspective projection scheme.

The classical calibration methods make use of a calibration pattern of known size such as a 2D or 3D calibration grid with regular patterns painted on it. In the rest of this paper, it is assumed that the camera has been calibrated thus, the camera calibration matrix $\mathbf{K}$ containing the intrinsic camera parameters is known. Then, given a pose $\mathbf{P}$ of the object, the projection of any 3D point onto the image can be computed as

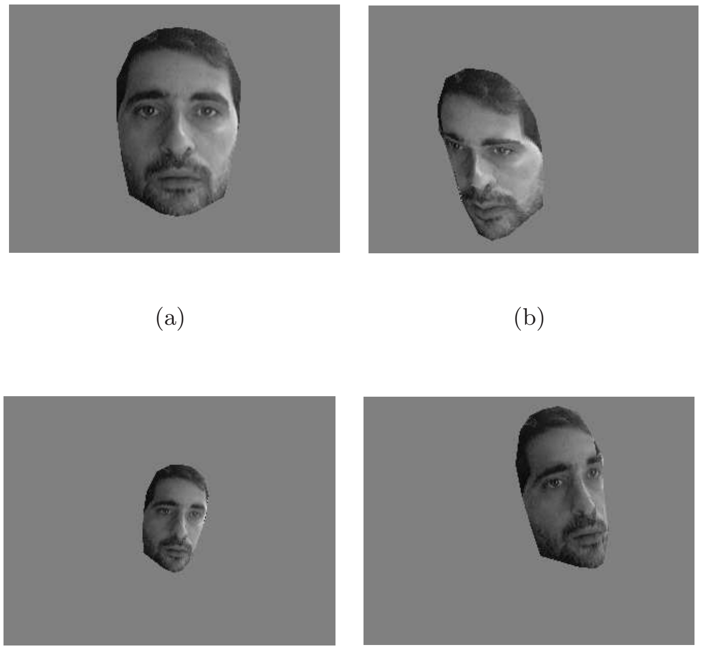
$$[x, y, 1]^T = \mathbf{T}[X, Y, Z, 1]^T \tag{4}$$

where $[x, y]$ is the coordinate vector in the 2D image obtained by the perspective projection of the 3D point $[X, Y, Z]$. This projection is depicted in Fig. 2.

2.2 Face Model Set-Up

In order to track the head pose and facial expression of a particular person's face in an image sequence, the appearance of the generic 3D face model should be adapted to the current person's face. The appearance of the 3D face model is defined by the shape parameter vector $\boldsymbol{\sigma}$. Thus, the values of vector $\boldsymbol{\sigma}$ are computed in order to fit the 3D face model to the person's face. Since vector $\boldsymbol{\sigma}$ only modifies the face model's appearance, it should be computed only once, either automatically or manually, to adapt the 3D face model to the person's face and it remains constant during the tracking. In this work, the generic 3D face model is adapted to the person's face to be tracked using one image of the person facing the camera. The adaptation is obtained by modifying the values of the shape parameter vector $\boldsymbol{\sigma}$ in order to adjust the appearance of the 3D face model to the person's face. To simplify this registration process, the 3D face model is projected onto the image and the values of the shape parameters are manually adjusted using a set of sliders in a graphical interface, as can be seen in Fig. 3. As a result of this process, the shape parameter vector $\boldsymbol{\sigma}$ is obtained and a textured face model is created using the texture from the input face image. The proposed tracking algorithms use this textured face model to create reference images, hereafter referred as "keyframes" (offline information), for estimating the current head pose and facial expression. Figure 4 shows four keyframes created from the texture face model with different positions, *i.e.*, with different poses $\mathbf{P}$.

Figure 3: 3D face model manual adaptation.



(a) (b) (c) (d)

Figure 4: Keyframes created from the textured face model in different positions.

3 Tracking 3D Model Vertices Projections

The tracking method proposed in this paper consists in estimating the pose and facial animation parameters of the 3D model for each frame of a video sequence. As mentioned before, these estimations rely on the correspondence between 3D face model and 2D image features, *i.e.*, the projection of the 3D model vertices onto the current frame. In this section, the proposed algorithms for tracking the 3D model vertices frame by frame are described. The proposed algorithms for head pose and facial expression estimation are described in section 4. For a particular frame, these estimations are performed by taking into account both the previous frame and a keyframe. The following subsections describe the proposed algorithms to find the projection of the model vertices onto the current frame based on the previous frame (subsection 3.1) and based on the keyframe (subsection 3.2).

3.1 Tracking based on Previous Frame

Assuming that the pose of the face at frame $(t-1)$, denoted as $\mathbf{P}_{t-1}$, is known, the projection of each 3D model vertex $\mathbf{g}_i$ at frame $(t-1)$ is also known because it can be computed by Eq. (4). Let $\mathbf{m}_i^{t-1}$ be the projection onto the image at frame $(t-1)$ of the 3D model vertex $\mathbf{g}_i$. At frame t, the new projection $\mathbf{m}_i^t$ of the 3D model vertex $\mathbf{g}_i$ is estimated using the pyramidal Lucas-Kanade optical flow method [Lucas, 1984, Wu, 1995]. This method tries to establish correspondences of invariant features between time-varying images. In this case, the method finds the correspondence between $\mathbf{m}_i^{t-1}$ and $\mathbf{m}_i^t$ in frames $(t-1)$ and t, respectively. In this way, it can be considered that the 3D vertex $\mathbf{g}_i$ is projected onto $\mathbf{m}_i^t$ at frame t. This procedure is illustrated in Fig. 5. The method also computes a value related with the level of confidence in the estimation of the new position of the 2D feature. If this error surpasses a given threshold, the projection is not used for the estimation of the head pose and the facial expression described in section 4.

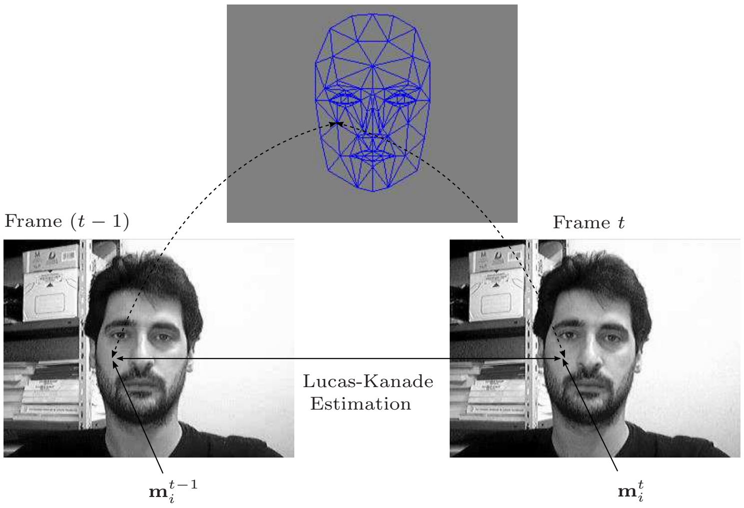


Figure 5: Estimation of the projection $\mathbf{m}_i^t$ of the 3D model vertex $\mathbf{g}_i$ at frame t, based on the location at the previous frame $(\mathbf{m}_i^{t-1})$.

3.2 Keyframe-based Tracking

The above described tracking method, used to find the correspondences between 2D features in two different images, gives good results when there is very small perspective distortion between the two frames. Thus, it is well suited for the case of consecutive frames. However, since the algorithm is applied recursively from frame to frame, it is not robust from the point of view of error accumulation. In addition, it is prone to lose a partially occluded target object and it tends to drift when there is jitter in the images. To overcome these drawbacks, the information from the previous frame is combined with keyframe information. The projection of the 3D model vertices $\mathbf{g}_i$ onto the keyframe image, denoted as $\mathbf{m}_i^k$, are known and they remain fixed. Thus, there is no tracking error accumulation if the projections of the model vertices are estimated from the keyframe. Similarly to the tracking described in the previous section, the keyframe image is used to estimate the projection of the 3D face model vertices onto the current frame t by the pyramidal Lucas-Kanade optical flow method. The projection of the i-th vertex, estimated from the keyframe, is denoted as $\widetilde{\mathbf{m}}_i^t$. Then, it is assumed that the model 3D vertex $\mathbf{g}_i$, projected onto $\mathbf{m}_i^k$ at keyframe, is projected onto $\widetilde{\mathbf{m}}_i^t$ at frame t.

In order to obtain good point matching using the pyramidal Lucas-Kanade optical flow method, the perspective distortion between the current frame image and the keyframe should be relatively small, *i.e.*, the viewpoints, or poses, of the face at the current frame and at the keyframe should be similar, ideally equal. As in the case of estimating the projection from the previous frame, it can be assumed that the pose at the previous frame ($\mathbf{P}_{t-1}$) is close to the pose at the current frame ($\mathbf{P}_t$). Thus, before applying the Lucas-Kanade algorithm, the keyframe is created from the textured face model with pose equal to $\mathbf{P}_{t-1}$. Figure 6 depicts this procedure. In this paper only one keyframe is used, but the proposed algorithms can be adapted to use several keyframes.

4 Head Pose and Facial Expression Estimation

Head pose and facial expression of the 3D face model are parameterized by matrix $\mathbf{P}$ and animation parameter vector $\boldsymbol{\alpha}$, respectively. Figure 7 depicts a block diagram of the proposed algorithm for head pose and facial expression estimation at each video frame. As it can be seen, head pose and facial expression are estimated in two separate stages. In a first stage, head pose is estimated based on the current frame t, the previous frame $(t-1)$ and the keyframe using $\mathbf{P}_{t-1}$. Then, the keyframe is updated using the new head pose $\mathbf{P}_t$ and it is merged with the current and previous frames, to estimate the animation parameter vector $\boldsymbol{\alpha}$. Both stages use the methods described in subsections 3.1 and 3.2 to find the correspondence between the 3D model vertices and the pixel position in each image. The tracking system also includes a stage to detect when the tracker loose the target. This block determines if the tracking should be re-initialized or not. The following subsections describe the proposed head pose and facial expression estimation algorithms.

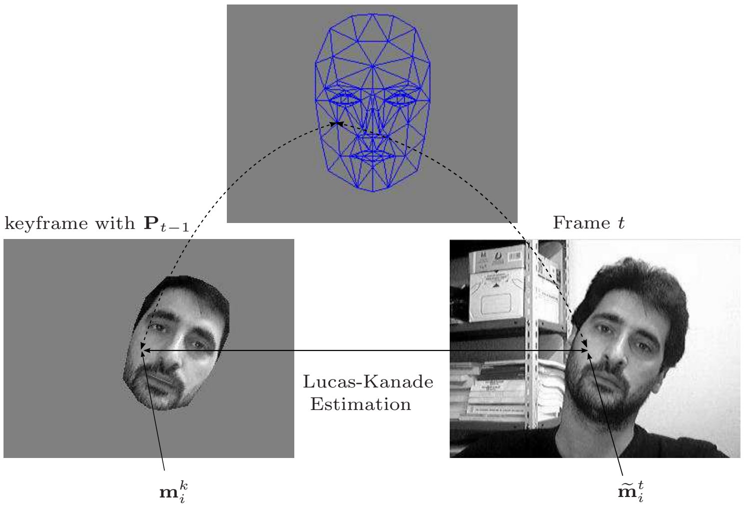


Figure 6: Estimation of the projection $\widetilde{\mathbf{m}}_i^t$ of the 3D model vertex $\mathbf{g}_i$ at frame t, based on the location at the keyframe ($\mathbf{m}_i^k$).

4.1 Head Pose Estimation

The head pose is estimated based on the projection of a set of 3D face model vertices onto the current frame. In order to estimate the head pose independently of the facial expression, this set is composed by 3D vertices which positions are not related with facial expressions such as mouth and eyebrows movements. Let $\mathcal{Q}_p$ be a set with the indices of the model vertices used for head pose estimation. To estimate $\mathbf{P}_t$, the methods described in sections 3.1 and 3.2 are used to compute the projections of the vertices with indices in $\mathcal{Q}_p$ on the current frame from the previous frame and the keyframe, respectively. Then, the head pose at time t is estimated by searching the matrix $\mathbf{P}$ that best match the projections of the 3D vertices at $\mathbf{m}_i^t$ and $\widetilde{\mathbf{m}}_i^t$ computed from the previous frame and the keyframe, respectively. Optimal values of $\mathbf{P}$ can be computed by solving the following minimization problem

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \{V_p(\mathbf{P}, \mathcal{Q}_p)\} \tag{5}$$

$$\begin{aligned} V_p(\mathbf{P}, \mathcal{Q}_p) = \sum_{i \epsilon \mathcal{Q}_p} \Big[&\rho_T \left(\|\varphi_p(\mathbf{g}_i, \mathbf{P}) - \mathbf{m}_i^t\|^2 \right) + \\ &+ \rho_T \left(\|\varphi_p(\mathbf{g}_i, \mathbf{P}) - \widetilde{\mathbf{m}}_i^t\|^2 \right) \Big] \end{aligned} \tag{6}$$

where $\varphi_p(\mathbf{g}_i, \mathbf{P})$ denotes the projection of 3D vertex $\mathbf{g}_i$ given the pose $\mathbf{P}$ and ρ_T is the Tukey M-estimator used for reducing the influence of wrong matches

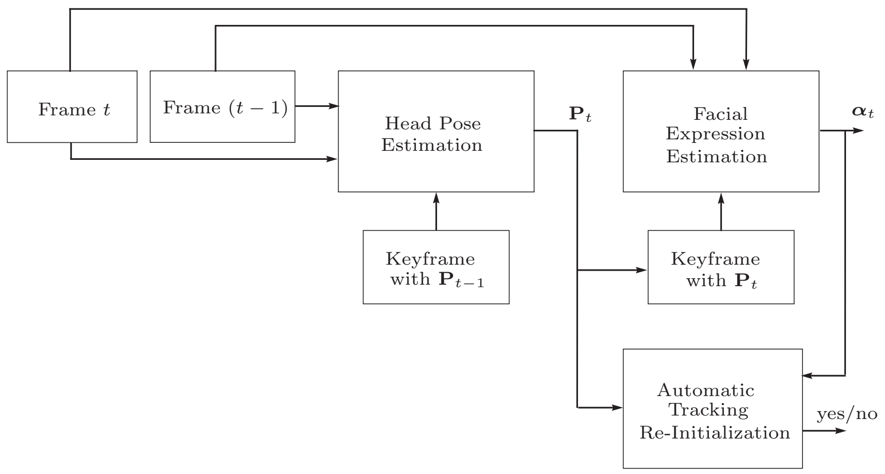


Figure 7: Schematic representation for head pose and facial expression estimation at frame t.

[Huber, 1981]. Finally, the minimization problem in Eq. (5) is solved by the Levenberg-Marquardt algorithm [Levenberg, 1944, Marquardt, 1963] using the actual pose as initial value for the estimation. Since the initial conditions are close to the actual values, the algorithm converges in few iterations.

4.2 Facial Expression Estimation

The idea for estimating the facial expression, parameterized by vector $\boldsymbol{\alpha}$, is similar to the one used to estimate the head pose described above, *i.e.*, search for the values of $\boldsymbol{\alpha}$ that best match the estimated projections of a set of 3D vertices onto the current frame. In this case, the 3D face model is not rigid but it is treated as a deformable model, where the position of its vertices are modified by $\boldsymbol{\alpha}$ as describes Eq. (1). As was described in subsection 2, the Animation Units matrix $\mathbf{A}$ controls the movements of several parts of the face model. Without loss of generality, in this work only the following six actions were considered, which are enough to cover most common facial expressions (mouth and eyebrow movements).

- Jaw drop.
- Lip corner depressor.
- Eyebrow lowerer.
- Lip stretcher.
- Upper lip raiser.
- Outer eyebrow raiser.

Let $\mathcal{Q}_F$ be a set containing the indices of the model vertices related with the above mentioned actions. Analogously to the case of head pose estimation, the projections of the vertices in $\mathcal{Q}_F$ onto the current frame from the previous frame and from the keyframe are computed. In this case, the keyframe is previously updated to the pose $\mathbf{P}_t$, as is depicted in Fig. 7. Denoting with $\mathbf{m}_i^t$ and $\widetilde{\mathbf{m}}_i^t$ the projection computed from the previous frame and the keyframe images, respectively, optimal values of $\boldsymbol{\alpha}$ can be computed by minimizing the following cost function,

$$V_F(\boldsymbol{\alpha}, \mathbf{P}_t, \mathcal{Q}_F) = \sum_{i \epsilon \mathcal{Q}_F} \Big[\rho_T \left(\|\varphi_\alpha(\mathbf{g}_i, \mathbf{P}_t, \boldsymbol{\alpha}) - \mathbf{m}_i^t\|^2 \right) + \\ + \rho_T \left(\|\varphi_\alpha(\mathbf{g}_i, \mathbf{P}_t, \boldsymbol{\alpha}) - \widetilde{\mathbf{m}}_i^t\|^2 \right) \Big] \quad (7)$$

where $\varphi_\alpha(\mathbf{g}_i, \mathbf{P}_t, \boldsymbol{\alpha})$ denotes the projection of the 3D vertex $\mathbf{g}_i$ given the pose $\mathbf{P}_t$ and the animation parameter vector $\boldsymbol{\alpha}$. That is,

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \{V_F(\boldsymbol{\alpha}, \mathbf{P}_t, \mathcal{Q}_F)\} \quad (8)$$

Finally, the minimization problem in Eq. (8) is solved by the Levenberg-Marquardt algorithm [Levenberg, 1944, Marquardt, 1963].

4.3 Tracking Initialization and Automatic Re-Initialization

To detect the head pose in the first frame, where no information about the previous frame is available, the algorithm proposed in [Viola and Jones, 2004] is used. This algorithm requires the head to be facing the camera and it delivers a rectangular region containing the face. If a face is detected in the current image, the system tries to determinate if the detected face corresponds to the person's face to be tracked, *i.e.*, the one used in the manual adaptation of the 3D face model. This decision is made by comparing the image of the detected face with the keyframe facing the camera. To perform this comparison, the keyframe is scaled according to the size of the detected face, and it is correlated with the current frame. If the computed correlation surpasses a given threshold, the system decides that the face has been detected. Then, the above described estimation algorithms are used to compute the current pose and facial expression and the tracking continues recursively. If the algorithm proposed in [Viola and Jones, 2004] does not detect a face at the current frame or if the computed correlation is smaller than the threshold, the current frame is discarded and the same procedure is applied on the next frame. Thus, tracking will not be started until the person's face is facing the camera.

As many tracking algorithms, the tracker proposed in this paper can break down in some conditions, such as when the image gets blurred or the quality is not sufficient for the feature detection or when the target disappears from the visual field of the camera. In order to avoid the divergence of the tracking algorithm, the keyframe is compared with the region of the current image that corresponds to head pose and facial expression estimation. This comparison can be done at every frame or periodically. When the tracker loose the target, the difference between the keyframe and the current estimated face will be increased. If this difference surpasses a given threshold, the tracking is re-initialized. This procedure prevents tracker's divergence, but it also allows a continuous tracking when the person's face disappears and appears into the camera visual field.

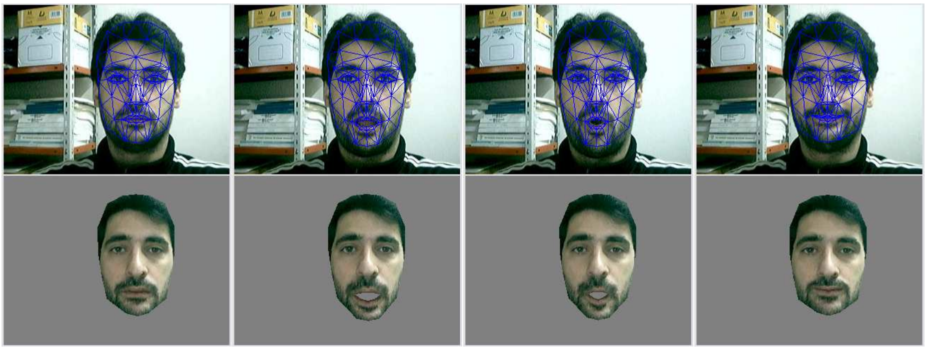
5 Experimental Results

To evaluate the performance of the head pose and facial expression tracking algorithms proposed in this paper, two video databases were used, *viz.* , the database described in [La Cascia et al., 2000][1] for head pose tracking, and a database[2] compiled by the authors of this paper. The databases consist of videos with persons talking and moving their faces in front of a camera. The videos were recorded using a standard webcam at a rate of 30 frames per second, with a resolution of 320×240 pixels. In order to test the ability of the proposed algorithms to work properly in different situations, videos containing partial face occlusions and changes in the distance between the camera and the face were recorded. The algorithms were implemented in C++ and the evaluation was performed on a 2.53GHz Intel Core 2 Duo processor. Although the implementation of the algorithms was not optimized for speed, the tracking can be made up to 22 fps. A further optimization of the algorithms would allow them to run in real-time. To perceptually evaluate the tracking algorithms, the wireframe of the face model, with the estimated pose and facial expression, was superimposed over each video frame. Additionally, a synthesized textured head model was animated to be compared with the actual video. A sequence of frames of the actual video and the animated model is shown in Fig. 8(a).

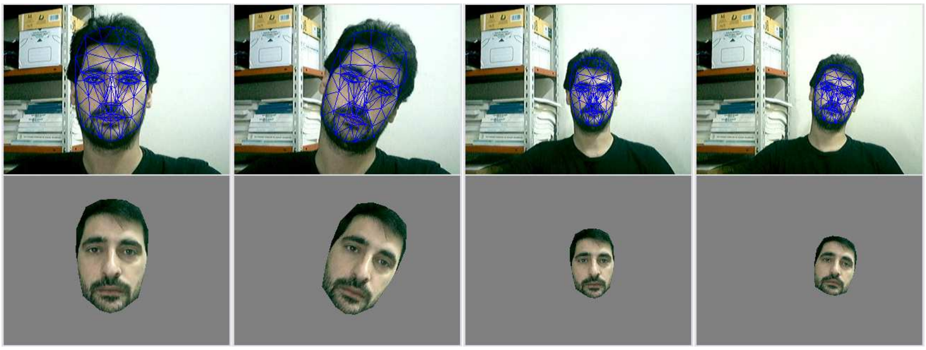
Figure 8(a) shows a sequence of frames from video *video_1.avi* where only mouth movements (no face occlusion or head movements) are present. An accurate tracking of the mouth movements can be observed from the figure. A sequence of frames from video *video_2.avi* where head movements and changes in the distance to the camera are present is shown in Fig. 8(b). Also here, an accurate tracking can be observed. The algorithm compensates for the changes in scale of the object being tracked.

Figure 9(a) shows a sequence of frames from video *video_3.avi* where there are scale changes and partial occlusion of the face. The video sequence shows

[1] `http://www.cs.bu.edu/groups/ivc/HeadTracking/`
[2] `http://www.fceia.unr.edu.ar/lsd/mrg/audiovisual/jucs09_videos/`



(a)

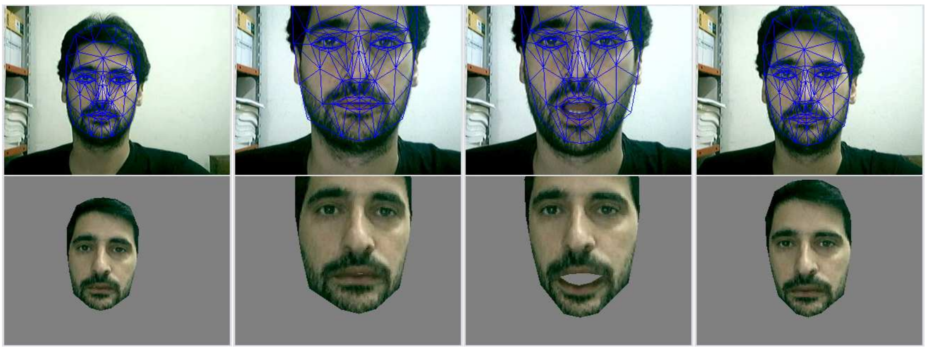


(b)

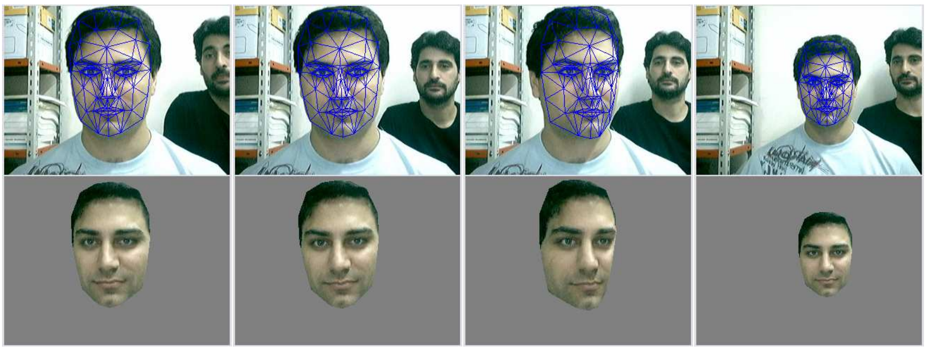
Figure 8: (a) Frames from *video_1.avi* where mouth movements are tracked correctly. (b) Frames from *video_2.avi* where head movements and changes in the distance to the camera are present.

the tracked face getting closer to the camera and partially coming out from the visual field of the camera. It can be observed that the face is tracked correctly even when partial occlusion of the face takes place.

To evaluate the performance of the proposed tracking algorithms in the presence of more than one face in the scene, a video (*video_4.avi*) with two persons was recorded. A sequence of frames of this video is shown in Fig. 9(b). It can be observed that the tracking results are not affected by the presence of other faces in the scene. Figure 10 shows a sequence of frames from video *jam1.avi*, included



(a)



(b)

Figure 9: (a) Frames from *video_3.avi* where partial occlusion occurs. (b) Frames from *video_4.avi* where there are two persons in the scene.

in the database described in [La Cascia et al., 2000], where head translation and rotation are present.

The database described in [La Cascia et al., 2000] provides ground truth data collected via a "Flock of Birds" 3D magnetic tracker attached on the subjects head. However, due to the *Candide 3* model specification, the pose of the model used by the proposed algorithms is referenced to a coordinate system that is located near the nose, thus, both coordinate systems are different. For that reason, ground truth data was manually collected to evaluate the algorithms quantitatively. In Fig. 11, the estimated 3D pose parameters during tracking (solid line)

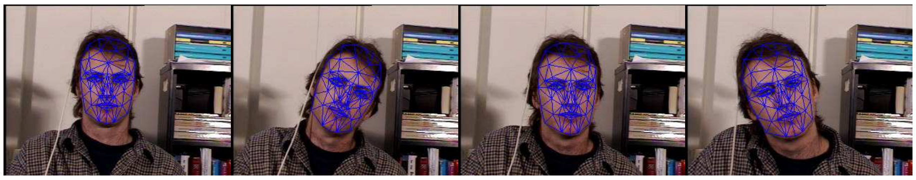


Figure 10: Frames from video *jam1.avi*, included in the database described in [La Cascia et al., 2000], where head translation and rotation are present.

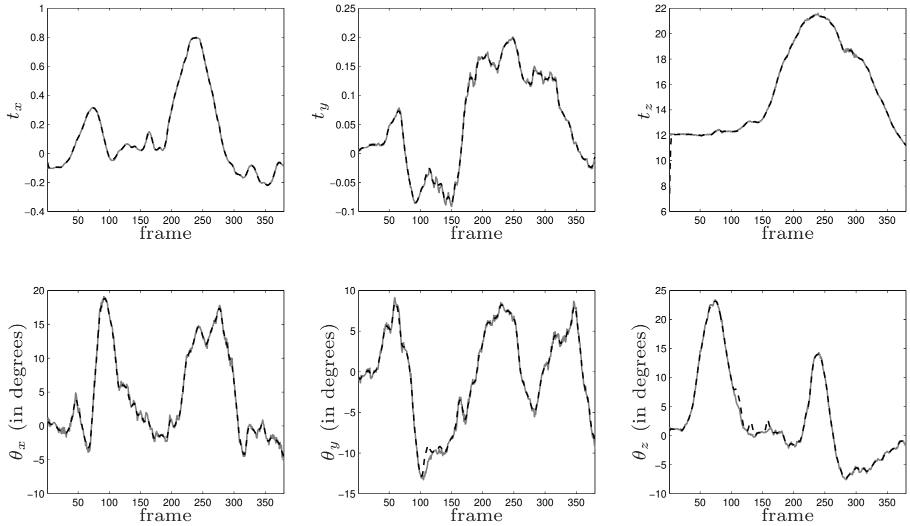


Figure 11: 3D head pose tracking. The graphs show the estimated 3D pose parameters during tracking (solid line) compared to ground truth (dashed line), computed from the sequence *video_2.avi*.

and ground truth data (dashed line) are compared, computed from the sequence *video_2.avi*. It can be seen that the head pose is tracked correctly, even in the presence of significant head translation and rotation. Figure 12 shows the comparison between estimated facial animation parameters (solid line) and ground truth data (dashed line) extracted from the sequence *video_1.avi*. In particular, the parameters associated with mouth movements are shown. Also here, it can be observed that the parameters are accurately estimated.

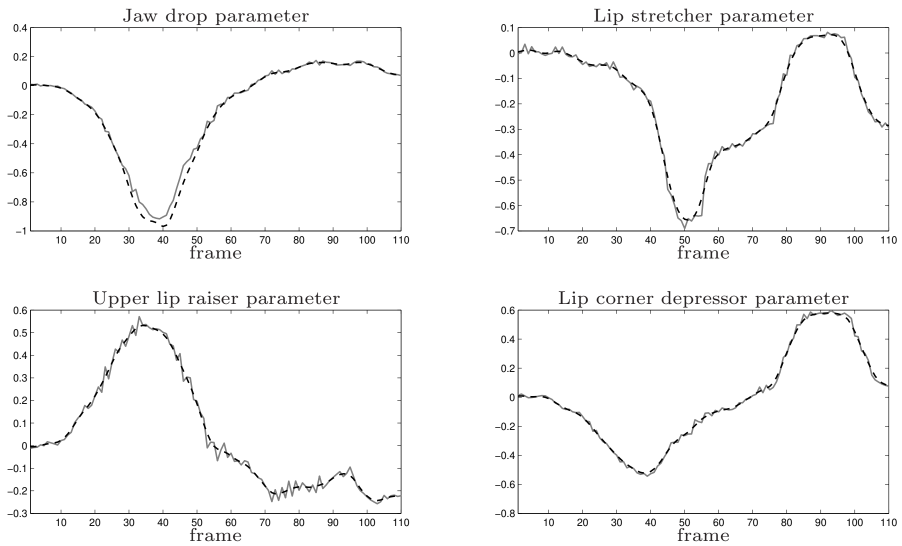


Figure 12: Facial expression tracking. The graphs show the estimated facial expression parameters during tracking (solid line) compared to ground truth (dashed line), computed from the sequence *video_1.avi*.

6 Conclusions and Future Work

In this paper, algorithms for markerless 3D head pose and facial expression tracking, based on monocular image sequences (single camera), were presented. A combination of feature-based and model-based approaches for pose estimation is the base of the proposed algorithms. The algorithms were evaluated both perceptually and quantitatively with two video databases. The tracking algorithm is made person-independent by adapting a generic 3D face model to the particular face. Both perceptual and quantitative results are similar to the ones obtained with other state-of-the-art methods. In contrast to other methods in the literature, the proposed method does not require a training stage. It only requires an image of the person's face to be tracked facing the camera, to which the model is fitted manually through a graphical user interface. This fitting procedure can be performed in a couple of minutes. To perceptually evaluate the tracking algorithms, the wireframe of the face model, with the estimated pose and facial expression, was superimposed over each video frame. Additionally, a synthesized textured head model was animated to be compared with the actual video. Simulation results show that the proposed tracking method correctly estimates the head pose and facial expression, even when occlusions, changes in the distance to the camera and presence of other persons in the scene, occur. Although the algorithms were not optimized for speed, they run near real time at 22 fps. The proposed system delivers separate head pose and facial expression information. Since information related with facial expression is independent from head pose information, the tracking algorithms could be used for facial expression analysis as well.

Future work will focus on the development of algorithms for adapting the appearance of the generic 3D face model to the person's face automatically rather than manually, and also on the speed optimization of the presented algorithms, in order for them to run in real-time. The tracking system presented in this paper will be integrated to the speech-driven facial animation system described in [Terissi and Gómez, 2008]. In particular, extracted facial expression information will be combined with speech information.

References

[Ahlberg, 2001] Ahlberg, J.: "An updated parameterized face"; Technical Report LiTH-ISY-R-2326; Department of Electrical Engineering, Linköping University, Sweden (2001).

[Busso and Narayanan, 2007] Busso, C., Narayanan, S. S.: "Interrelation between speech and facial gestures in emotional utterances: A single subject study"; IEEE Transactions on Audio, Speech, and Language Processing; 15, 8 (2007), 2331–2347.

[La Cascia et al., 2000] La 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, 2 (2000), 322-336.

[Cristinacce and Cootes, 2006] Cristinacce, D., Cootes, T.: "Feature detection and tracking with constrained local models"; Proceedings of the British Machine Vision Conference; 929–938; Edinburgh, 2006.

[Dornaika and Ahlberg, 2006] Dornaika, F., Ahlberg, J.: "Fitting 3D face models for tracking and active appearance model training"; Image and Vision Computing; 24 (2006), 1010–1024.

[Huber, 1981] Huber, P.: Robust Statistics; Wiley, 1981.

[Jimenez et al., 2008] Jimenez, J., Gutierrez, D., Latorre, P.: "Gaze-based interaction for virtual environments"; Journal of Universal Computer Science; 14, 19 (2008), 3085–3098.

[Levenberg, 1944] Levenberg, K.: "A method for the solution of certain nonlinear problems in least-squares"; The Quarterly of Applied Mathematics; 2 (1944), 164–168.

[Li et al., 2007] Li, Z., Fu, Y., Yuan, J., Huang, T., Wu, Y.: "Query driven localized linear discriminant models for head pose estimation"; Procedding of IEEE International Conference on Multimedia and Expo; 1810–1813; Beijing, 2007.

[Lucas, 1984] Lucas, B. D.: Generalized Image Matching by the Method of Differences; Ph.D. thesis; Robotics Institute, Carnegie Mellon University (1984).

[Marquardt, 1963] Marquardt, D.: "An algorithm for least-square estimations of non-linear parameters"; SIAM Journal on Applied Mathematics; 11 (1963), 431–441.

[Murphy-Chutorian and Trivedi, 2009] Murphy-Chutorian, E., Trivedi, M. M.: "Head pose estimation in computer vision: A survey"; IEEE Transactions on Pattern Analysis and Machine Intelligence; 31, 4 (2009), 607–626.

[Pallejà et al., 2008] Pallejà, T., Rubión, E., Teixido, M., Tresanchez, M., del Viso, A. F., Rebate, C., Palacin, J.: "Using the optical flow to implement a relative virtual mouse controlled by head movements"; Journal of Universal Computer Science; 14, 19 (2008), 3127–3127.

[Savrana et al., 2006] Savrana, A., Arslana, L. M., Akarunb, L.: "Speaker-independent 3D face synthesis driven by speech and text"; Signal Processing; 86, 10 (2006), 2932–2951.

[Sturm and Maybank, 1999] Sturm, P., Maybank, S.: "On plane-based camera calibration: A general algorithm, singularities, applications"; Conference on Computer Vision and Pattern Recognition; 432–437; 1999.

[Terissi and Gómez, 2007] Terissi, L. D., Gómez, J. C.: "Facial motion tracking and animation: An ICA-based approach"; Proceedings of 15th European Signal Processing Conference; 292–296; Poznań, Poland, 2007.

[Terissi and Gómez, 2008] Terissi, L. D., Gómez, J. C.: "Audio-to-visual conversion via HMM inversion for speech-driven facial animation"; Lecture Notes in Computer Science; 5249 (2008), 33–42.

[Vacchetti et al., 2004] Vacchetti, L., Lepetit, V., Fua, P.: "Stable real-time 3D tracking using online and offline information"; IEEE Transactions on Pattern Analysis and Machine Intelligence; 26, 10 (2004), 1385–1391.

[Viola and Jones, 2004] Viola, P., Jones, M. J.: "Robust real-time face detection"; International Journal of Computer Vision; 57, 2 (2004), 137–154.

[Wu, 1995] Wu, Q. X.: "Correlation-relaxation-labeling framework for computing optical flow-template matching from a new perspective"; IEEE Transactions on Pattern Analysis and Machine Intelligence; 17, 8 (1995), 843–853.

[Zhang, 2000] Zhang, Z.: "A flexible new technique for camera calibration"; IEEE Transactions on Pattern Analysis and Machine Intelligence; 22 (2000), 1330–1334.