A linear estimation method for 3D pose and facial animation tracking.

José Alonso Ybáñez Zepeda
E.N.S.T.
75014 Paris, France
ybanez@ieee.org

Franck Davoine
CNRS, U.T.C.
60205 Compiègne cedex, France.
Franck.Davoine@hds.utc.fr

Maurice Charbit
E.N.S.T.
75014 Paris, France
maurice.charbit@enst.fr

Abstract

This paper presents an approach that incorporates Canonical Correlation Analysis (CCA) for monocular 3D face pose and facial animation estimation. The CCA is used to find the dependency between texture residuals and 3D face pose and facial gesture. The texture residuals are obtained from observed raw brightness shape-free 2D image patches that we build by means of a parameterized 3D geometric face model. This method is used to correctly estimate the pose of the face and the model's animation parameters controlling the lip, eyebrow and eye movements (encoded in 15 parameters). Extensive experiments on tracking faces in long real video sequences show the effectiveness of the proposed method and the value of using CCA in the tracking context.

1. Introduction

Head pose and facial gesture estimation is a crucial task in several computer vision applications, like video surveillance, human-computer interaction, biometrics, vehicle automation, etc. It poses a challenging problem because of the variability of facial appearance within a video sequence. This variability is due to changes in head pose (particularly out-of-plane head rotations), facial expression, or lighting, to occlusions, or a combination of all of them.

Different approaches exist for tracking moving objects, two of them being feature-based and model-based. Feature-based approaches rely on tracking local regions of interest, like key points, curves, optical flow, or skin color [5, 10]. Model-based approaches use a 2D or 3D object model that is projected onto the image and matched to the object to be tracked [9, 7]. These approaches establish a relationship between the current frame and the information that they are looking for. Some popular methods to find this relation use a gradient descent technique like the active appearance models AAMs [4, 15], a statistical based technique using support or relevant vector machines (SVM and RVM) [2, 14], or a regression technique based on the Canonical Correlation Analysis (CCA) (linear or kernel based). CCA is a statistical method which relates two sets of observations, and that is well suited for regression tasks. CCA has recently been used for appearance based 3D pose estimation [11], appearance-based localization [12] and to improve the AAM search [6]. These works highlight the advantages of the CCA to obtain regression parameters that outperform standard methods in speed, memory requirements and accuracy (when the parameter space is not too small).

In this paper we present a model-based approach that incorporates CCA for monocular 3D face pose and facial animation estimation. This approach fuses the use of a parameterized 3D geometric face model with the CCA in order to correctly track the facial gesture corresponding to the lip, eyebrow and eye movements and the 3D head pose encoded in 15 parameters.

Although model-based methods and CCA are traditionally used in the computer vision domain, these two methods together were not already used in the tracking context. We will show experimentally on different public and our own video sequences that, indeed, our CCA approach is well suited to obtain a simple and effective facial pose and gesture tracker.

2. Face representation

The use of a 3D generic face model for tracking purposes has been widely explored in the computer vision community. In this section we show how we use the *Candide-3* face model to acquire the 3D geometry of a person's face and the corresponding texture map for tracking purposes.

2.1. 3D geometric model

The 3D parameterized face model *Candide-3* [1] is controlled by Animation Units (AUs). The wireframe consists of a group of n 3D interconnected vertices to describe a face with a set of triangles. The $3n$-vector $\mathbf{g}$ consists of the concatenation of all the vertices, and can be written in a parametric form as:

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\boldsymbol{\tau}_a, \quad (1)$$

where the columns of $\mathbf{A}$ are face Animation Units and the vector $\boldsymbol{\tau}_a$ contains 69 animation parameters [1] to control facial movements so that different expressions can be obtained. $\mathbf{g}_s = \overline{\mathbf{g}} + \Delta\mathbf{g} + \mathbf{S}\boldsymbol{\tau}_s$ corresponds to the static geometry of a given person's face: $\overline{\mathbf{g}}$ is the standard shape of the *Candide* model, the columns of $\mathbf{S}$ are Shape Units and the vector $\boldsymbol{\tau}_s$ contains 14 shape parameters [1] used to reshape the wireframe to the most common head shapes. The vector $\Delta\mathbf{g}$ can be used if necessary to adapt the 3D model to non-symmetric faces locally by moving vertices individually. $\Delta\mathbf{g}$, $\boldsymbol{\tau}_s$ and $\boldsymbol{\tau}_a$ are initialized manually, by fitting the *Candide* shape to the face shape facing the camera in the first video frame (see Figure 1).

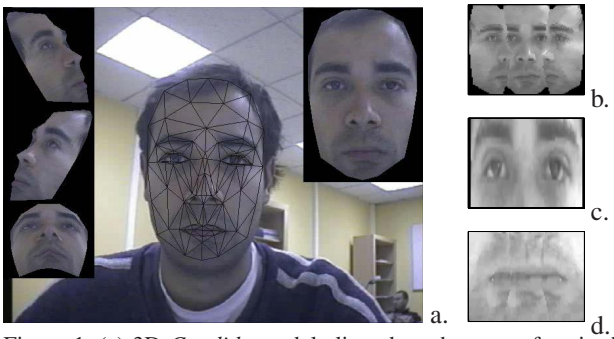

Figure 1. (a) 3D *Candide* model aligned on the target face in the first video frame with the 2D image patch mapped onto its surface (upper right corner) and three other semi-profile synthesized views (left side). (b),(c) and (d) Stabilized face images used for tracking the pose: `SFI_1`, the eyebrows and the eyes: `SFI_2`, the mouth: `SFI_3`, respectively.

The facial 3D pose and animation state vector $\mathbf{b}$ is then given by:

$$\mathbf{b} = \left[\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau}_a^T\right]^T, \quad (2)$$

where $\theta_.$ and $t_.$ components stand respectively for the model rotation around three axes and translation.
In this work, the geometric model $\mathbf{g}(\mathbf{b})$ will be used to crop out underlying image patches from the video frames and to transform faces into a normalized facial shape for tracking purposes, as described in the next section. We will limit the dimension of $\boldsymbol{\tau}_a$ to 9, in order to only track eyebrows, eyes and lips. In that case, the state vector $\mathbf{b} \in \mathbb{R}^{15}$.

2.2. Stabilized face image

We consider here a stabilized 2D shape free image patch (also called a texture map) to represent the facial appearance of the person facing the camera and to represent observations from the incoming video frame $\mathbf{Y}$. The patch is built by warping the rawbrightness image vector lying under the model $\mathbf{g}(\mathbf{b})$ into a fixed size 2D projection of the standard *Candide* model without any expression (i.e. with $\boldsymbol{\tau}_a = 0$). This patch augmented with two semi-profile views of the face, to track rotation in a wider range, is written as $\mathbf{x} = \mathcal{W}(\mathbf{g}(\mathbf{b}), \mathbf{Y})$, where $\mathcal{W}$ is a warping operator (see Figure 1.b). We will see in section 4 how to use other stabilized face images to represent and track the upper and lower facial features of the face (Figures 1.c and 1.d).

3. Integrated tracking framework

In this section, we describe our algorithm for face and facial animation tracking. It is composed of three steps: initialization, learning and tracking. In step one, the shape of the *Candide* model is aligned to the face in the first video frame. Using the stabilized face image (we call it the reference stabilized face), we train the system, in step two, by synthesizing new views of the face with standard computer graphics texture-mapping techniques. CCA is used to learn the relation between the changes in the model parameters and the corresponding residuals between the reference stabilized face and the synthesized faces. Then, the tracking process at time t consists in obtaining the stabilized face image from the incoming frame $\mathbf{Y}_t$ using the estimated state vector $\mathbf{b}_{t-1}$ at time $t-1$ and in computing the difference between this image and the reference stabilized face. The error vector is used to predict the variation in the state vector. Once the state vector is updated, we update the reference stabilized face image and continue with the next incoming frame. The three steps are more precisely described in the following sub-sections.

3.1. Initialization

The *Candide* model is placed manually over the first video frame $\mathbf{Y}_0$ at time $t = 0$ and reshaped to the person's face. From this model we generate three semi-profile synthesized views (see Figure 1.a) in order to verify the accuracy of the alignment. Once the model is aligned, we obtain the state vector $\mathbf{b}_0$, and the reference stabilized face image:

$$\mathbf{x}_0^{(ref)} = \mathcal{W}(\mathbf{g}(\mathbf{b}_0), \mathbf{Y}_0). \quad (3)$$

3.2. Training

Due to the high dimensionality that arises when working with images, the use of a linear mapping to extract some linear features is common in the computer vision domain. One of the most prominent methods for dimensionality reduction is *Principal Component Analysis* (PCA) which deals with one data space and identifies directions of high variance. However, in our case, we are interested in identifying and quantifying the linear relationship between two data

sets: the change in state of the *Candide* model and the corresponding facial appearance variations. Using first a PCA and then trying to find the linear relation between two projected data sets can lead to a loss of information, as PCA-features might not be well suited for regression tasks. In our case we propose to use a *Canonical Correlation Analysis* (CCA) to find linear relations between two sets of random variables [3, 13]. CCA finds pairs of directions or basis vectors for two sets of m vectors, one for $\mathbf{Q}_1 \in \mathbb{R}^{m\times n}$ and the other for $\mathbf{Q}_2 \in \mathbb{R}^{m\times p}$, such that the projections of the variables onto these directions are maximally correlated.

Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be the centered versions of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ respectively. The maximum number of basis vectors that can be found is $min(n, p)$. If we map our data to the directions $\mathbf{w}_1$ and $\mathbf{w}_2$ we obtain two new vectors defined as:

$$\mathbf{z}_1 = \mathbf{A}_1\mathbf{w}_1 \quad \text{and} \quad \mathbf{z}_2 = \mathbf{A}_2\mathbf{w}_2. \tag{4}$$

and we are interested in finding the correlation between them, which is defined as:

$$\rho = \frac{\mathbf{z}_2^T\mathbf{z}_1}{\sqrt{\mathbf{z}_2^T\mathbf{z}_2}\sqrt{\mathbf{z}_1^T\mathbf{z}_1}}. \tag{5}$$

Our problem consists in finding vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ that maximize (5) subject to the constraints $\mathbf{z}_1^T\mathbf{z}_1 = 1$ and $\mathbf{z}_2^T\mathbf{z}_2 = 1$.

In this work, we use the numerically robust method proposed in [13]. We compute singular value decompositions of the data matrices $\mathbf{A}_1 = \mathbf{U}_1\mathbf{D}_1\mathbf{V}_1^T$ and $\mathbf{A}_2 = \mathbf{U}_2\mathbf{D}_2\mathbf{V}_2^T$, and then, the following the singular value decomposition: $\mathbf{U}_1^T\mathbf{U}_2 = \mathbf{U}\mathbf{D}\mathbf{V}^T$, to finally get:

$$\mathbf{W}_1 = \mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{U} \quad \text{and} \quad \mathbf{W}_2 = \mathbf{V}_2\mathbf{D}_2^{-1}\mathbf{V}, \tag{6}$$

where matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ contain the full set of *canonical correlation basis vectors*. In our case, the matrix $\mathbf{A}_1$ contains the difference between the training observation vectors $\mathbf{x}_{Training} = \mathcal{W}(\mathbf{g}(\mathbf{b}_{Training}), \mathbf{Y}_0)$ and the reference $\mathbf{x}_0^{(ref)}$, and the matrix $\mathbf{A}_2$ contains the variation in the state vector $\Delta\mathbf{b}_{Training}$ given by $\mathbf{b}_{Training} = \mathbf{b}_0 + \Delta\mathbf{b}_{Training}$. The m training points were chosen empirically from a non-regular grid around the vector state obtained at initialization (Figure 2).

Once we have obtained all the canonical correlation basis vectors, the general solution consists in performing a linear regression between $\mathbf{z}_1$ and $\mathbf{z}_2$. However, if we develop (5) for each pair of directions with the assumptions made above, we get $\|\mathbf{A}_1\mathbf{w}_1 - \mathbf{A}_2\mathbf{w}_2\|^2 = 2(1-\rho)$ similarly as in [8]. Based on our experiments, we observe that $\rho \approx 1$, and so, we can substitute matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ by $\Delta\mathbf{b}_t$ and $(\mathbf{x}_t - \mathbf{x}_t^{(ref)})$ in the relation $\mathbf{A}_1\mathbf{w}_1 \approx \mathbf{A}_2\mathbf{w}_2$ to come to:

$$\Delta\mathbf{b}_t\mathbf{w}_2 = (\mathbf{x}_t - \mathbf{x}_t^{(ref)})\mathbf{w}_1. \tag{7}$$

This is true for all the *canonical variates*, so we substitue equations (6) to get a result for all the directions:

$$\Delta\mathbf{b}_t = (\mathbf{x}_t - \mathbf{x}_t^{(ref)})\mathbf{G}, \tag{8}$$

where the matrix $\mathbf{G} = \mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{U}\mathbf{V}^T\mathbf{D}_2\mathbf{V}_2^T$ encodes the linear model used by our tracker, which is explained in the following section.

3.3. Tracking

The tracking process consists in estimating the state vector $\Delta\mathbf{b}_t$ when a new video frame $\mathbf{Y}_t$ is available. In order to do that, we need, first, to obtain the stabilized face image, from the incoming frame by means of the state at the preceding time, as:

$$\mathbf{x}_t = \mathcal{W}(\mathbf{g}(\mathbf{b}_{t-1}), \mathbf{Y}_t), \tag{9}$$

and then make the difference between this image and the reference stabilized face image $\mathbf{x}_t^{(ref)}$. This gives an error vector from which we estimate the changes in state with (8). Then we can write the state vector update equation as:

$$\hat{\mathbf{b}}_t = \mathbf{b}_{t-1} + (\mathbf{x}_t - \mathbf{x}_t^{(ref)})\mathbf{G}. \tag{10}$$

We iterate a fixed number of times (5, in practice) and estimate another $\hat{\mathbf{b}}_t$ according to equation (10) and update the state vector. Once the iterations are done, we update the reference stabilized face image according to:

$$\mathbf{x}_{t+1}^{(ref)} = \alpha\mathbf{x}_t^{(ref)} + (1-\alpha)\hat{\mathbf{x}}_t \tag{11}$$

with $\alpha = 0.99$ obtained from experimental results. In [16], CCA is compared KCCA for pose tracking. We observe similar tracking performances, with larger run-time requirements for the KCCA-based method.

4. Implementation

The algorithm has been implemented on a PC with a 3.0 GHz Intel Pentium IV processor and a NVIDIA Quadro NVS 285 graphic card. Our non optimized implementation uses OpenGL for texture mapping and OpenCV for video capture. We used a standard desktop Winnov analog video camera to generate the sequences we use for tests.

We retain the following nine animation parameters of the *Candide* model, for tracking facial gestures:

(1) upper lip raiser
(2) jaw drop
(3) mouth stretch
(4) lip corner depressor
(5) eyebrow lowerer
(6) outer eyebrow raiser
(7) eyes closed
(8) yaw left eyeball
(9) yaw right eyeball

Based on the algorithm described in section 3, we have implemented, for comparison purposes, three versions of

the tracker combining different stabilized face images. The first version of the algorithm uses a stabilized face image (SFI_1, in Figure 1) to estimate simultaneously the 6 head pose parameters and the 9 facial animation parameters. The second version uses a stabilized face image (SFI_1) to estimate simultaneously the head pose and the lower face animation parameters (parameters (1) to (4)) and then, starting from the previously estimated state parameters, we use a stabilized face image (SFI_2, in Figure 1) to estimate the upper face animation parameters (5) to (9). Finally, the third version of the tracker uses three stabilized face images sequentially: one to track the head pose (SFI_1), one to track the lower face animation parameters (SFI_3, in Figure 1), and a last one (SFI_2) to track the upper face animation parameters. SFI_1, SFI_2 and SFI_3 are respectively composed of 96×72, 86×28, and 88×42 pixels.

For training, we use 317 training state vectors with the corresponding appearance variations for the pose, 240 for the upper face region and 200 for the mouth region. The same points are used in the three implemented versions. These vectors correspond to variations of $\pm 20°$ for the rotations, $\pm 10.5\%$ of the face width for translations, and animation parameter's values corresponding to valid facial expressions. We chose these points empirically, from a symmetric grid centered on the initial state vector. The sampling is dense close to the origin and coarse as it moves away from it (see Figure 2). Due to the high dimensionality of our state vectors, even after the separation into three models, we did not use all the combinations between the chosen points. It is important to say that we consider the lower and the upper face animation parameters as mutually independent.

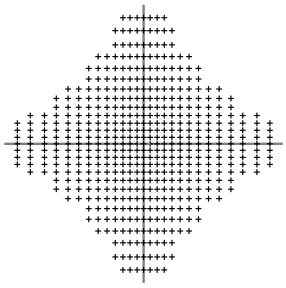

Figure 2. 2D representation of the sampled *Candide* parameters.

5. Experimental results

For validation purposes, we use in this paper the video sequences described in [9] [1] for pose tracking, and the talking face video made available from the *Face and Gesture Recognition Working group* [2], for both pose and facial animation tracking as these sequences are supplied with ground truth data. In this section, we show and analyze quantitatively the performance of the tracker over the two types of video sequences.

3D pose tracking. Video sequences provided in [9] are 200 frames long, with a resolution of 320×240, 30 *fps.*, taken under uniform illumination, where the subjects perform free head motion including translations and both in-plane and out-of-plane rotations. Ground truth has been collected via a "Flock of Birds" 3D magnetic tracker. Figure 3 shows the estimated pose compared with the ground data. We use here the first version of our tracker based on the stabilized face image SFI_1. Temporal shifts can be explained because the center of the coordinate systems used in [9] and ours are slightly different. In our case, the three axes cross close to the nose, due to the *Candide* model specification, and in the ground truth data, the 3D magnetic tracker is attached on the subject's head. We check experimentally (on all the provided video sequences) the stability and precision of the tracker and do not observe divergences of the tracker.

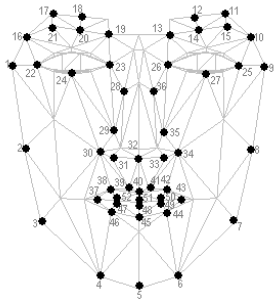

Figure 4. *Candide* model with the corresponding talking face ground truth's points used for evaluation.

Simultaneous pose and facial animation tracking. The talking face video sequence consists of 5000 frames, with a resolution of 720×576, taken from a video of a person engaged in conversation. This corresponds to about 200 seconds of recording. The sequence was taken as part of an experiment designed to model the behavior of the face in natural conversation. For practical reasons (to display varying parameter values on readable graphs) we used 1720 frames of the video sequence, where the ground truth consists of characteristic 2D facial points annotated semi-automatically. From 68 annotated points per frame, we select 52 points that are closer to the corresponding *Candide* model points, as can be seen in Figure 4. In order to evaluate the behavior of our algorithm we calculated for each point the standard deviation of the distances between the ground truth and the estimated coordinates divided by the face width. Figure 5 depicts the standard deviation over the whole video sequence for each point using the three implementations of our algorithm. We can see that the points

[1] www.cs.bu.edu/groups/ivc/HeadTracking/

[2] www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html

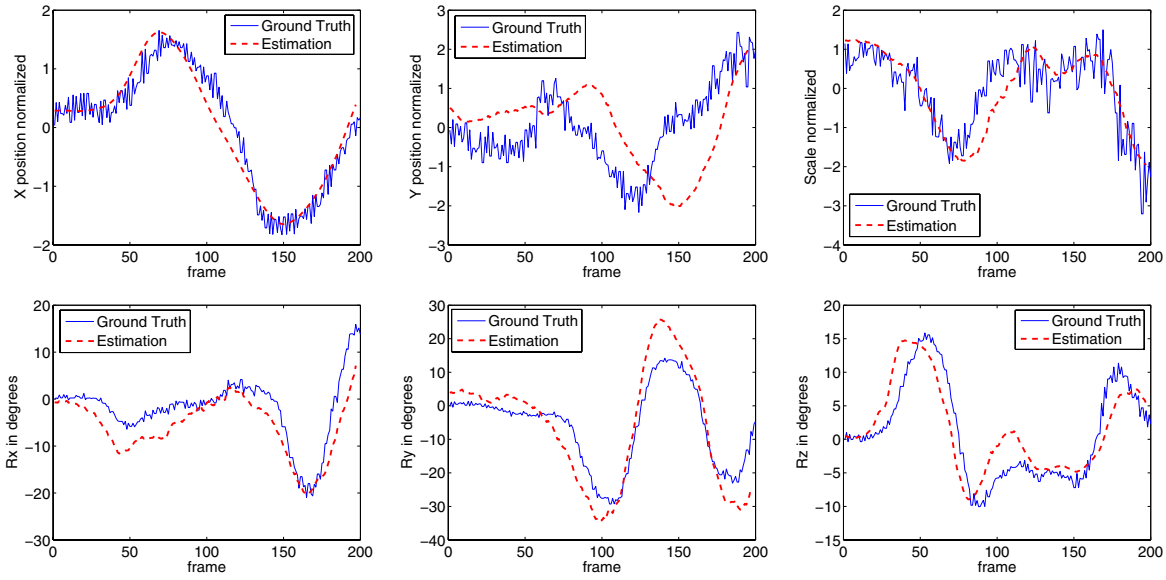

Figure 3. 3D pose tracking results: the graphs show the estimated 3D pose parameters during tracking compared to ground truth.

with the greater standard deviation correspond to those on the contour of the face. The precision of these points is strongly related to the correctness of the estimated pose parameters. The best performance, as expected, is obtained with the third version of our algorithm based on three stabilized face images to estimate first the pose, then the lower face animation parameter and finally the upper face animation parameters. When using a single model, the tracker presents some imprecision. The second version of the algorithm (based on the two stabilized face images `SFI_1` and `SFI_2`) improves the estimation of the upper face animation parameters. However, when estimating the eyes' movement separately, the tracking is improved. The fact of going from two to three models presented an improvement only for the points corresponding to the mouth, but no further improvements were obtained for the pose estimation. Based on these results, we retain the third version of the tracker to explore its robustness.

The α parameter affects the way we update the reference stabilized face image in (11). From experiments we find that $\alpha = 0.99$ is a good choice. It is important to say that when ther is no update, i.e. $\alpha = 1$, the tracker diverged. We see that the mean standard deviation of the 52 facial points stays approximately constant with some peaks. These peaks correspond to important facial movements. In the case of frame 993 the rotation around the y axis corresponds to $36.62°$. In frame 1107, the rotations around on the x, y and z axes are respectively $-13.3°$, $18.9°$ and $-10.5°$. We observe on the whole video sequence that even if peak values are large, the tracker still performs correctly. Figure 7 shows sample frames extracted from the whole talking face

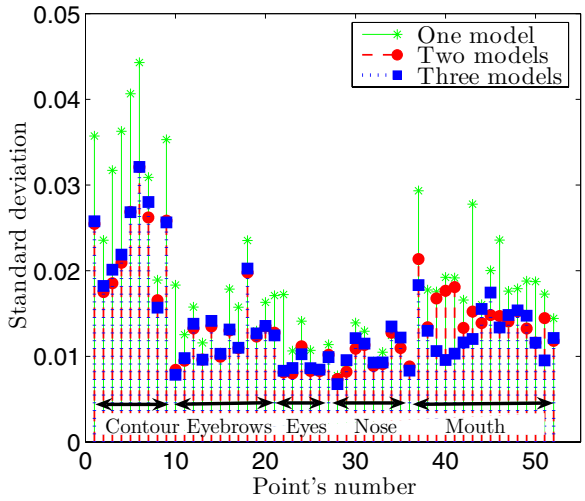

Figure 5. Standard deviation of the 52 facial points w.r.t. the face width, using the three version of our algorithm, to track both the pose and facial animation parameters.

video sequence and from different video sequences part of the data set in [9] and from a webcam. We can appreciate the robustness of the tracker even in the case of cluttered backgrounds.

Experiments were conducted to evaluate the sensitivity of the facial animation tracker in the case of imprecise 3D pose estimations. Given that the tracker first estimates the 3D pose of the face and then, based on this estimation, it estimates the lower and the upper face animation parameters,

Figure 7. Frames from different video sequences showing the pose and gesture tracking. From top to bottom: Talking face video sequence, two La Cascia's video sequence and two video sequences from a webcam.

we added some random noise to the six pose parameters before estimating the facial animation, within the following intervals: $\pm 10\%$ of the estimated head width added to the three translation parameters, and $\pm 3°$ added to the three rotation parameters. In order to prove the robustness of the facial animation estimation, we have added some gaussian noise to the estimated pose parameters with a standard deviation of 10% of the training intervals described in Section 4. Figure 8 shows the stability of the "eyebrow lowerer" animation parameter estimation even if the six pose parameters have been previously altered.

Features like eyebrows and eye closure tend to change rapidly during emotional expressions. We show in Figure 9 the time evolution of the "eye closed" animation parameter, and observe on the graph that 18 eye blinks and one long period with closed eyes around frame 100 are correctly detected. This is confirmed when looking at the talking subject in natural conversation in the video sequence.

6. Conclusion

We have presented a method that is capable of tracking both 3D pose and facial animation parameters from individuals in monocular video sequences in real time. The tracking approach is simple from the training and tracking points of view, robust even to slight illumination changes and precise when the out-of-plane face rotation angles stay in the interval $\pm 30°$. The technique can still be improved. As regards immediate extensions, the method will be combined with a facial feature detection algorithm to re-synchronize the tracking in case of divergence. Future work will also address the robustness of the tracker to more important illumination changes (this could be integrated in the $\mathbf{G}$ matrix), and its sensitivity to depth initialization of the model.

References

[1] J. Ahlberg. Candide-3 – an updated parameterized face. Technical Report LiTH-ISY-R-2326, Linkoping University, Sweden, Jan 2001. 1, 2

[2] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, August 2004. 1

[3] M. Borga, T. Landelius, and H. Knutsson. A unified approach to PCA, PLS, MLR and CCA. Report LiTH-ISY-R-1992, SE-581 83 Linköping, Sweden, November 1997. 3

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, June 2001. 1

[5] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *Proc. of BMVC*, Edinburgh, UK, September 2006. 1

[6] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pat-*

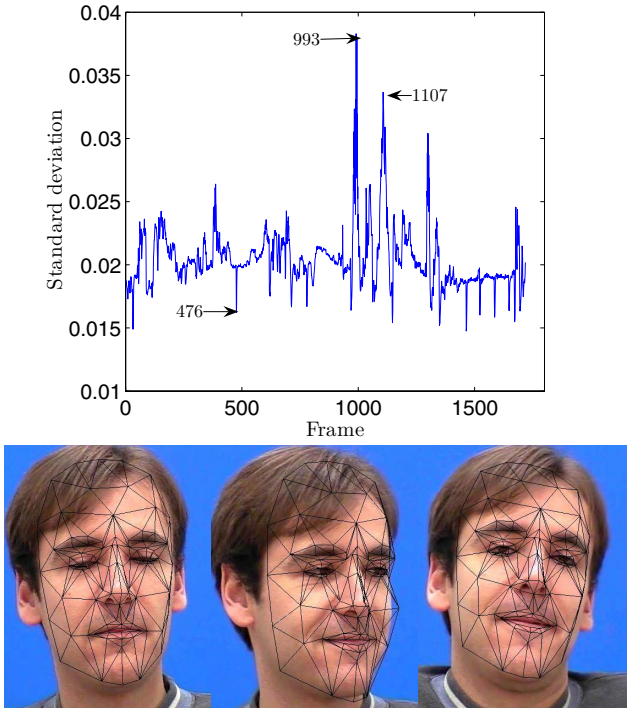

Figure 6. Standard deviation of all the points w.r.t. the face width for each video frame, and three frames of the sequence corresponding, to frame 476 to frame, 993, and 1107 respectively, using the CCA algorithm.

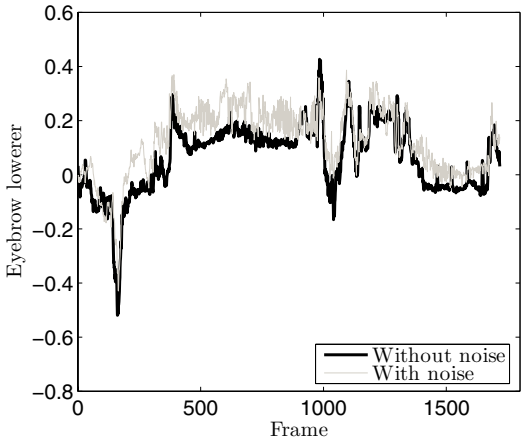

Figure 8. Estimated eyebrow lowerer animation parameter with and without perturbations applied on the previously estimated pose parameters.

tern Analysis and Machine Intelligence, 28(10):1690–1694, Oct. 2006. 1

[7] F. Dornaika and F. Davoine. On appearance based face and facial action tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(9):1107– 1124, September 2006. 1

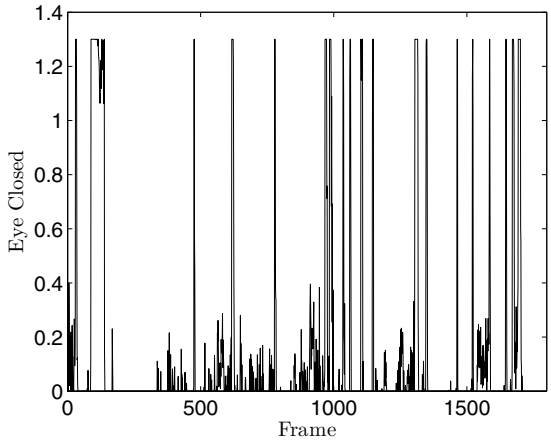

Figure 9. Temporal evolution of the "eye closed" animation parameter.

[8] D. R. Hardoon, J. Shawe-Taylor, and O. Friman. KCCA for fMRI analysis. *The Medical Image Understanding and Analysis conference*, 2004. 3

[9] M. La Cascia, S. Sclaroff, and V. Athitsos. 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):322–336, April 2000. 1, 4, 5

[10] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 244–250, 2004. 1

[11] T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 36(9):1961–1973, 2003. 1

[12] D. Skocaj and A. Leonardis. Appearance based localization using CCA. *Proceedings of the 9th Computer Vision Winter Workshop*, pages 205–214, February 2004. 1

[13] D. Weenink. Canonical correlation analysis. In *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam, Netherlands*, volume 25, pages 81–99, 2003. 3

[14] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1292–1304, August 2005. 1

[15] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 535 – 542, June 2004. 1

[16] J. A. Ybáñez, F. Davoine, and M. Charbit. Face tracking using canonical correlation analysis. In *International Conference on Computer Vision Theory and Applications*, pages 396–402, Barcelona, Spain, March 2007. 3