# LOCAL OR GLOBAL 3D FACE AND FACIAL FEATURE TRACKER.

*José Alonso Ybáñez Zepeda* [1*], *Franck Davoine* [2] *and Maurice Charbit* [1]

[1] GET-ENST, Department TSI
LTCI Laboratory, CNRS
Paris, France

[2] UTC, CNRS
HEUDIASYC Laboratory
Compiègne, France

## ABSTRACT

We present in this paper a solution for 3D face and facial feature tracking using canonical correlation analysis and a 3D geometric model. This model is controlled with 17 parameters (6 for the 3D pose, and 11 for facial animation), and is used to crop out reference 2D shape free texture maps from the incoming input frames. Model parameters are updated via image registration in the texture map space. For registration, we use CCA to learn and exploit the dependency between texture residuals and model parameter corrections. We compare tracking results using two kinds of texture maps: one local (image patches around selected vertices of the 3D model), and one global (the whole image patch under the 3D model). Experiments evaluating the effectiveness on the approaches are reported.

***Index Terms***— Machine vision, feature extraction, tracking, canonical correlation analysis.

## 1. INTRODUCTION

This paper addresses the problem of tracking in a single video the global pose of a face as well as the local motion of its main inner features, due to expressions, for instance, or other facial behaviors. Many popular learning-based or model-based approaches have been proposed in the literature. The second ones generally use a 2D or 3D model that is projected into the image and matched to the face to track [1, 2]. Most approaches rely on image cues like key points, curves, optical flow, appearance or skin color, and make use of linear/nonlinear generative or discriminative statistical models to work with 2D facial shape or global appearance manifolds. A recent work addressing canonical correlation analysis for fast active appearance model search is described in [3]. In our paper, we develop a regression based approach using CCA, in a tracking context. CCA is used to learn and recover pose and facial animation perturbations to apply to a geometric face model, from matching errors: for tracking, the relationship between the changes in the observed image and the changes in the pose and shape of the face model are learned and then applied iteratively to estimate the current pose and shape. Two appearance models are considered: one local (images patches around selected vertices of the 3D model), and one global (the whole image patch under the 3D model).

## 2. FACE REPRESENTATION

We use the *Candide-3* 3D generic face model used in [4]. This 3D parameterized face model 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 as $\mathbf{g} = \mathbf{g}_s + \mathbf{A}\boldsymbol{\tau}_a$, where the columns of $\mathbf{A}$, as described in [4], code 69 face Animation Units and the vector $\boldsymbol{\tau}_a$ is used to control facial mimics so that different expressions can be obtained. $\mathbf{g}_s$ corresponds to the static geometry of a given person's face. $\mathbf{g}_s$ and $\boldsymbol{\tau}_a$ are initialized manually, by fitting the *Candide* shape to the face shape facing the camera in the first video frame. The facial 3D pose and animation state vector $\mathbf{b}$ is then given by:

$$\mathbf{b} = \left[\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau}_a^T\right]^T, \qquad (1)$$

where $\theta_.$ and $t_.$ components stand respectively for the model rotation and translation around three axes. In this work, we limit the dimension of $\boldsymbol{\tau}_a$ to 11, ($\mathbf{b} \in \mathbb{R}^{17}$), in order to track eyebrows, eyes and lips, like in [5].

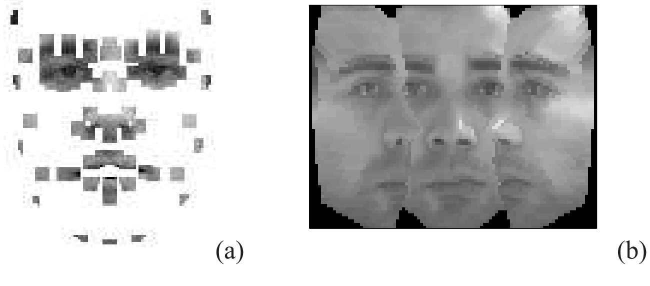
(a) (b)

**Fig. 1**. Local (a) and global (b) facial appearances.

The geometric model $\mathbf{g}(\mathbf{b})$ is used to crop out underlying appearances from video frames $\mathbf{Y}$ to obtain a normalized reference vector for tracking purposes. In our case we use two

[*] Thanks to CONACYT for funding.

approaches, using a local appearance or a global appearance as can be seen in figure 1.

The local appearance is build 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. Then, we select 96 vertices of the *Candide* model and extract independently normalized image patches of size $6 \times 6$ pixels around each point. Then, we concatenate all these patches to get an observation vector.

For the global model, the patch is build by warping the rawbrightness image vector lying under the model $\mathbf{g}(\mathbf{b})$ into normalized and resized 2D projection of the standard *Candide* model without any expression ($\boldsymbol{\tau}_a = 0$).

The observation vector can be written as $\mathbf{x} = \mathcal{W}(\mathbf{g}(\mathbf{b}), \mathbf{Y})$, where $\mathcal{W}$ is a warping operator, for both models.

## 3. TRACKING PROPOSITION

Our algorithm for face and facial animation tracking is composed of three steps: initialization, learning and tracking. These 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. We get from this the state vector $\mathbf{b}_0$, and the reference vector with the parameters:

$$\mathbf{x}_0^{(ref)} = \mathcal{W}(\mathbf{g}(\mathbf{b}_0), \mathbf{Y}_0). \tag{2}$$

### 3.2. Training

For our algorithm, we are interested in identifying and quantifying the linear relationship between two data sets: the change in state of the *Candide* model and the associated facial appearance variations. We propose to use a *Canonical Correlation Analysis* (CCA) to find the linear relations between these two data sets [6, 7]. CCA finds pairs of directions or basis vectors for two sets of variables, such that the projections of the variables onto these directions are maximally correlated.

Let $\mathbf{A}_1 \in \mathbb{R}^{m \times n}$ and $\mathbf{A}_2 \in \mathbb{R}^{m \times p}$ be two centered data sets. The maximum number of basis vectors that can be found is $min(n, p)$. In our case, the matrix $\mathbf{A}_1$ contains the difference between the $n$-dimensional training observation vectors $\mathbf{x}_{Training} = \mathcal{W}(\mathbf{g}(\mathbf{b}_{Training}), \mathbf{Y}_0)$ and the reference $\mathbf{x}_0^{(ref)}$, and $\mathbf{A}_2$ contains the variation in the $p$-dimensional state vector $\Delta\mathbf{b}_{Training}$ given by $\mathbf{b}_{Training} = \mathbf{b}_0 + \Delta\mathbf{b}_{Training}$. The $m$ training points are chosen empirically from a non-regular symmetric grid centered on the initial state vector. The sampling is dense close to the origin and coarse as it moves away from it. If we map our data to the directions $\mathbf{w}_1$ and $\mathbf{w}_2$ we obtain two new vectors defined as:

$$\mathbf{z}_1 = \mathbf{A}_1\mathbf{w}_1 \quad \text{and} \quad \mathbf{z}_2 = \mathbf{A}_2\mathbf{w}_2. \tag{3}$$

The problem consists in finding vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ that maximize the correlation 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 [7] to obtain all the canonical correlation basis vectors. With these vectors, the general solution consists in performing a linear regression between $\mathbf{z}_1$ and $\mathbf{z}_2$, that after some mathematical manipulation lead us to [5]:

$$\Delta\mathbf{b}_t = (\mathbf{x}_t - \mathbf{x}_t^{(ref)})\mathbf{G}, \tag{4}$$

where $\mathbf{G}$, 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 reference face image, as the two that can be seen in figure 1, from the incoming frame by means of the state at the previous time, as:

$$\mathbf{x}_t = \mathcal{W}(\mathbf{g}(\mathbf{b}_{t-1}), \mathbf{Y}_t), \tag{5}$$

and then make the difference between this image and the reference face image $\mathbf{x}_t^{(ref)}$. This gives an error vector from which we estimate the changes in state with (4). Then we can write the state vector update equation as:

$$\hat{\mathbf{b}}_t = \mathbf{b}_{t-1} + (\mathbf{x}_t - \mathbf{x}_t^{(ref)})\mathbf{G}. \tag{6}$$

We iterate a fixed number of times (5, in practice) and estimate another $\hat{\mathbf{b}}_t$ according to (6) and update the state vector. Once the iterations are done, for robustness purposes we update $\mathbf{x}_{t+1}^{(ref)} = \alpha\mathbf{x}_t^{(ref)} + (1 - \alpha)\hat{\mathbf{x}}_t$, with $\alpha = 0.99$ obtained from experimental results.

## 4. IMPLEMENTATION

The algorithm has been implemented on a PC with a 3.0 GHz P4 processor and a NVIDIA Quadro NVS 285 graphic card. Our non optimized code uses OpenGL for texture mapping and OpenCV for video capture. We retain the following eleven animation parameters, for facial gesture tracking: (1) upper lip raiser, (2) jaw drop, (3) mouth stretch, (4) lip corner depressor, (5) left eyebrow lowerer, (6) right eyebrow lowerer, (7) left outer eyebrow raiser, (8) right outer eyebrow raiser, (9) eyes closed, (10) left eyeball's yaw, and (11) right eyeball's yaw.

Based on the algorithm described in section 3, we have implemented two trackers, one using a local appearance model, and another one using three global appearance sub-models.

The tracker based on the local appearance model uses small 6 × 6 patches around 96 vertices of the *Candide* model, cropped from the normalized frontal view of the face texture as the ones depicted in figure 1. For training, we use 748 state vectors with the corresponding appearance variations for the pose, the upper face region and the mouth region.

The tracker based on global appearances uses three reference face images sequentially: one to track the head pose as depicted in figure 1, one to track the lower face animation parameters, and a last one to track the upper face animation parameters [5]. These face images are respectively composed of 96 × 72, 86 × 28, and 88 × 42 pixels. For training, we use 317 state vectors with the corresponding appearance variations for the pose, 240 for the upper face region and 200 for the mouth region.

## 5. EXPERIMENTAL RESULTS

For validation purposes, we use the talking face video made available from the *FGnet Working Group*[1], for both pose and facial animation tracking. This sequence is supplied with ground truth data. It consists of 5000 frames (about 200 seconds of recording), with a resolution of 720 × 576, taken from a video of a person engaged in 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. In order to evaluate the behavior of our algorithms we calculated for each point the standard deviation of the distances between the ground truth and the estimated coordinates.

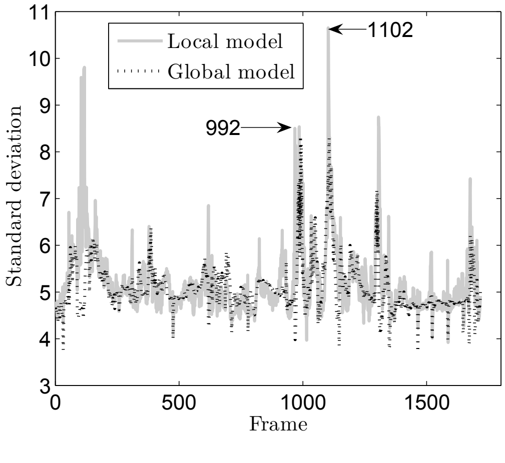

**Fig. 2**. Mean standard deviation evolution.

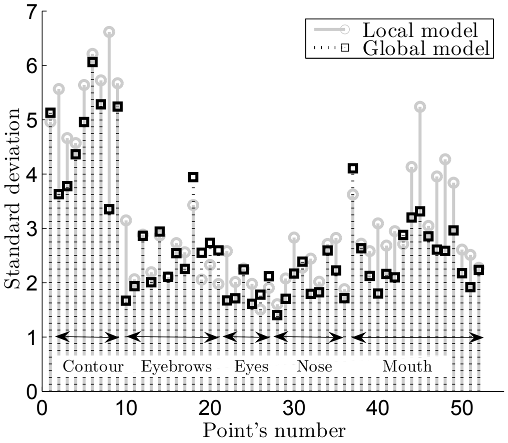

**Fig. 3**. Standard deviation of each point.

We see in figure 2 that the mean standard deviation of the 52 facial points stays around a constant value for both trackers with some peaks. These peaks correspond to important facial movements. In case of frame 992 the rotation around the $y$ axe corresponds to $36.62°$. In frame 1102, the rotations around on the $x$, $y$ and $z$ axes are respectively $-13.3°$, $18.9°$ and $-10.5°$. The tracker using the local model, presents slight oscillations when visually compared to the global model tracker, that can also be see from results depicted in figure 2. This is because the local model tracker is more sensitive to out-of-plane rotations and facial gestures.

Figure 3 depicts the standard deviation over the whole video sequence for each point. The points with the greater standard deviation correspond to those on the outer contour of the face. The precision of these points is strongly related to the correctness of the estimated pose parameters. In this figure we can see that apart from some points, the behavior of the global model tracker outperforms that of the local model one. This can be explained from the fact that the global model uses more information about the face than the local model, especially if we consider that we synthesize two profile views of the face for the global model. This makes the global model more robust to rotations, as can be seen in the frames shown in figure 5. The average time for pose and facial animation tracking is about 26 ms per frame for the local model tracker and about 46 ms per frame for the global model tracker if we exclude the time for video read, decompression and write/display operations. The average time for training is 29.1 and 23.2 seconds for the local and global model tracker respectively. Finally, to test the robustness of the trackers to illumination changes we used the challenging 967 frame long video sequence given by the Polytechnic University of Madrid[2]. Sample results with both trackers are shown in figure 4. We observe the same behaviors as before: the global appearance based tracker is more robust to significant pose variations, and the local appearance based tracker track more

[1] www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/

[2] http://www.dia.fi.upm.es/∼pcr/downloads.html

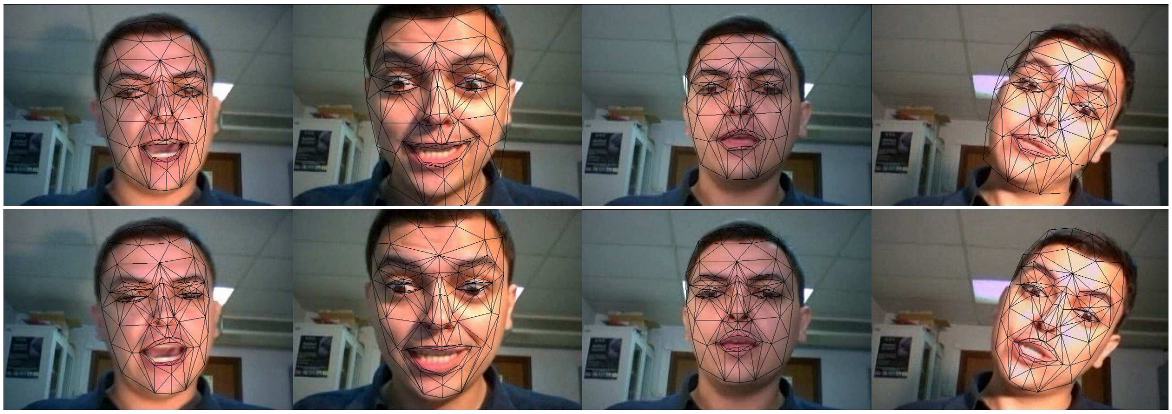

**Fig. 4**. Top row: frames obtained with the local model. Bottom: frames obtained with the global model

accurately the facial features if the pose is correctly estimated.

## 6. CONCLUSION

We have presented two approaches to track both 3D pose and facial animation parameters of individuals in monocular video sequences. The principal advantage of the local model approach is that it is faster and robust enough when there are neither strong out-of-plane rotations nor important facial gesture. It also presents a robust behavior when faced to important illumination changes. However, we can conclude that the global approach represents a better solution for real world conditions, where important rotations can appear and facial gestures are expected. It is important to notice that this global model approach is simple, from the training and tracking point of views, robust and accurate 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 important illumination changes.

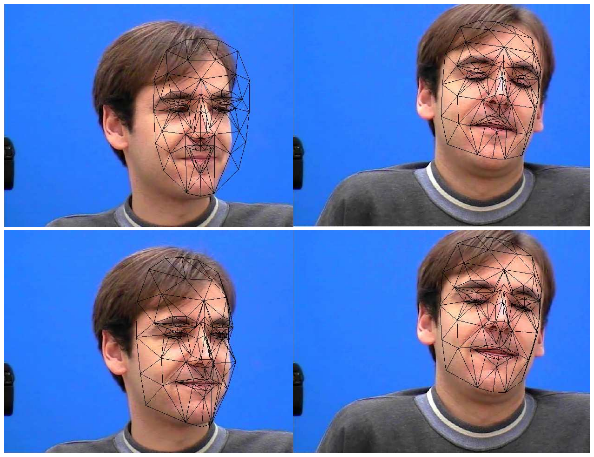

**Fig. 5**. Top row: frames 992 and 1102 obtained with the local model.Bottom: same frames obtained with the global model

## 7. REFERENCES

[1] 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*, vol. 22, no. 2, pp. 322–336, April 2000.

[2] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1107– 1124, September 2006.

[3] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, "Fast active appearance model search using canonical correlation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1690–1694, Oct. 2006.

[4] J. Ahlberg, "Candide-3 – an updated parameterized face.," Tech. Rep. LiTH-ISY-R-2326, Linkoping University, Sweden, Jan 2001.

[5] José Alonso Ybáñez, Franck Davoine, and Maurice Charbit., "Linear tracking of pose and facial features," in *10th IAPR Conference on Machine Vision Applications*, Tokyo, Japan, May 2007.

[6] 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.

[7] David Weenink, "Canonical correlation analysis," in *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam, Netherlands*, 2003, vol. 25, pp. 81–99.