STRUCTURE AND APPEARANCE FEATURES FOR ROBUST 3D FACIAL ACTIONS TRACKING

Stéphanie Lefèvre and Jean-Marc Odobez [*]

Idiap Research Institute, Martigny, Switzerland
Ecole Polytechnique Fédérale de Lausanne, Switzerland

ABSTRACT

This paper presents a robust and accurate method for joint head pose and facial actions tracking, even under challenging conditions such as varying lighting, large head movements, and fast motion. This is made possible by the combination of two types of facial features. We use locations sampled from the facial texture whose appearance is initialized on the first frame and adapted over time, and also illumination-invariant patches located on characteristic points of the face such as the corners of the eyes or of the mouth. The first type of features contains rich information about the global appearance of the face and thus leads to an accurate tracking, while the second type guaranties robustness and stability by avoiding drift. We demonstrate our system on the Boston University Face Tracking benchmark, and show it outperforms state-of-the-art methods.

***Index Terms*—** Face tracking, Head pose, Facial actions, 3D face model, Feature-based tracking

1. INTRODUCTION

Face tracking is a challenging task with many applications that has been investigated by researchers for years. The difficulties come from the variability of appearance created by 3D rigid movements (especially self occlusions due to the head pose), non-rigid movements (due to facial expressions), variability of people appearance and illumination variations.

The problem of near-frontal face tracking has been addressed many times in the past. An early and successful approach is to use Principal Component Analysis to model the 2D variations of the face shape (Active Shape Model (ASM)), or of both shape and appearance (Active Appearance Model (AAM) [1]). Some works have extended the use of AAM to larger head movements [2], but the lack of robustness when confronted to large head pose variations is still a typical limitation of these models. Besides the correspondence between the 2D fit and the 3D pose is not straightforward [3].

In parallel, approaches inspired from 3D registration were developped to robustly track faces under large pose variations. They usually rely on a rigid 3D face/head model, which can be a cylinder [4, 5], an ellipsoid [6], or a mesh [7, 8]. The model is fit to the image by matching either local features [7], a facial texture [4, 5, 6] or a sparse facial texture [8]. Such approaches can provide a precise estimate of the pose with no restriction to near-frontal views. However, they are limited to rigid movements. In the best case the tracking is robust to facial actions; in the worst case they will cause the system to lose track; in any case they are not estimated.

The use of a deformable 3D model (like Candide [9]) is an appropriate approach to handle both 3D pose and facial actions. To fit such 3D models to an image, a facial texture [10] or structural features [11] can be used. Illumination variations can be handled through continuous adaptation, but the resulting system is subject to drift. A model-based tracking algorithm was presented in [12], but the method is based on optical flow and therefore not robust to fast motions and fast lighting changes.

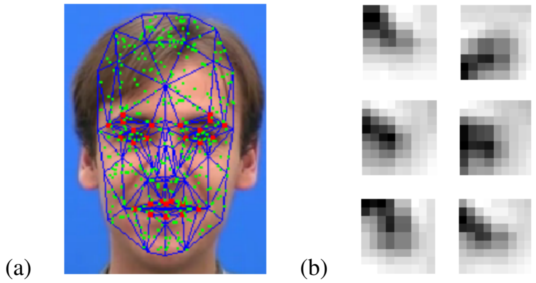

Fig. 1. (a) *Trained* (red dots) and *adaptive* (green dots) features. (b) Samples of the training set for the *trained* feature located on the right corner of the right eye (before removing the patch mean).

The idea conveyed in this paper is to combine two different types of features into a hybrid set in order to achieve a robust and precise tracking. The first set is made of 400 randomly picked locations and their intensity values, forming a sparse facial texture. The intensities are estimated online, we will therefore call them *adaptive* features. Because they are made of single pixels, they are very efficient to compute, and we will show that despite their simplicity they are critical to stabilize the pose estimates over time. This set provides a good model of the current appearance of the face and thus results in a precise frame-to-frame tracking, but it is subject to drift as the model is continuously adapted. Therefore we use a second set of features whose appearance model is fixed. In a similar way to [11], we use 26 local structural features located around the eyebrows, eyes and lips. They consist of 9×9 pixels patches, and a statistical appearance model is learned offline for each of them. For this reason we will refer to them as the *trained* features. Contrary to [11] the appearance model is invariant to illumination, that way we avoid drift. The two sets are illustrated in Fig. 1 (a). The performances of the system are evaluated on the Boston University Face Tracking (BUFT) database (on both Uniform-light and Varying-light datasets) and on two long real video sequences containing facial actions. They show that the combination of our two types of features allows us to outperform recent state-of-the-art techniques [4, 6], especially in the challenging case where lighting changes over time.

2. CANDIDE, A DEFORMABLE 3D MODEL

In this work we use the Candide [9] face model, a deformable 3D wireframe model defined by the 3D coordinates of 113 facial feature points. Inter-person and intra-person variations are generated by dis-

[*] This work was supported by the Swiss National Center of Competence in Research and Interactive Multimodal Information Management (IM2). We thank Vincent Lepetit for providing constructive feedback on this work.

placing the vertices of a standard face mesh without expression according to some predefined shape and action units. In our case we consider 14 shape units (such as Head height, Eyebrows vertical position, Eyes vertical position, Mouth vertical position) and 6 action units (such as Lower lip depressor, Outer eyebrow raiser), which are able to cover most common facial actions. A point $\overline{M}_i$ of the standard face mesh is transformed into a new point M_i as follows:

$$M_i(\alpha,\sigma) = \overline{M}_i + S_i.\sigma + A_i.\alpha$$

where S_i and A_i are respectively the 3×14 shape unit matrix and the 3×6 action unit matrix that contain the effect of each shape (respectively action) unit on point $\overline{M}_i$. The 14×1 shape vector σ and the 6×1 action vector α contain values between -1 and 1 that express the intensity of the displacement unit. σ is learned for a person before tracking while α varies from frame to frame.

The camera is not calibrated and we adopt the weak perspective projection model (i.e. we neglect the perspective effect) to map a 3D point M_i to an image point m_i. We will represent the rotation matrix (from object coordinate system to camera coordinate system) by the three Euler angles. Thus the vector of the head pose parameters to estimate can be expressed as $\Theta = [\theta_x \ \theta_y \ \theta_z \ \lambda t_x \ \lambda t_y \ s]$ where λ is a constant, $T = (t_x \ t_y \ t_z)^T$ is the translation matrix (from object coordinate system to camera coordinate system) and s is a scale factor (the Candide model is defined up to a scale factor). The whole state (head pose and facial actions parameters) at time t is defined as follows:

$$X_t = [\Theta_t \ \alpha_t] \ . \qquad (1)$$

3. TRACKING USING A HYBRID SET OF FEATURES

The algorithm is composed of a training and a tracking phase. During training, both the shape parameters and the appearance model for the *trained* features are learned specifically for the person to be tracked. During tracking, we use the downhill simplex optimization method [13] to maximize the posterior probability of the state.

3.1. Learning an appearance model for the *trained* features

The training phase requires a frontal view of the person with neutral expression. This frame will be called the reference frame, it can be the first frame of the sequence for instance. First, the shape parameters are learned by annotating several points on the face in the reference frame (either manually or using an automatic feature detector), and by finding the shape parameters σ that best fit the Candide model to the data points.

The goal is to learn for each point i of the *trained* features set S_{tr} an appearance model of the 9×9 image patch centered at point i valid under variations of pose and illumination. To this end, we extract such a patch in the reference frame, substract the mean value (to make it invariant to illumination changes), and simulate what it would look like under different head rotations by applying a serie of affine transformations to it (assuming the patch is planar). More precisely, for each of the three rotation parameters we sample seven values (from $-45°$ to $45°$). Sample patches are shown in Fig. 1 (b).

From this training set we compute the 1×81 mean vector μ_i and the 81×81 covariance matrix Σ_i. Given these learned values, the likelihood model for a normalized 9×9 image patch Z_i centered at point i is defined as:

$$p_i(Z_i) \propto e^{-\rho(d_{\Sigma_i}(Z_i,\mu_i),\tau_{tr})} \qquad (2)$$

where ρ is a robust function (we used the truncated quadratic function), τ_{tr} is the threshold above which a measurement is assumed to be an outlier and d_{Σ_i} is the Mahalanobis distance defined by:

$$d_{\Sigma_i}(Z_i,\mu_i) = [(Z_i-\mu_i)^T \Sigma_i^{-1} (Z_i-\mu_i)]^{\frac{1}{2}} \ .$$

3.2. Tracking phase

Our objective is to find the state X_t (as defined in Eq. (1)) which maximizes the posterior probability $p(X_t|Z_{1:t})$ of the state X at time t given observations Z_t from time 1 to time t. Under standard assumptions, this probability can be approximated by:

$$p(X_t|Z_{1:t}) \propto p(Z_t|X_t) \cdot p(X_t|\hat{X}_{t-1}) \ . \qquad (3)$$

This expression is characterized by two terms: the likelihood $p(Z_t|X_t)$, which expresses how good are observations given a state value, and $p(X_t|\hat{X}_{t-1})$ which represents the dynamics, i.e. the state evolution (with $\hat{X}_{t-1}$ being the previous estimate of the state). These terms are described below.

Likelihood modeling: As discussed in the introduction, our observations are collected around the projected positions of 3D points, i.e. the points $\{M_i\}_{i \in S_{tr}}$ for the *trained* features and $\{N_i\}_{i \in S_{ad}}$ for the *adaptive* features. More precisely, given the state X_t observations for the *trained* features $Z_t^{tr} = \{Z_{i,t}(X_t)\}_{i \in S_{tr}}$ will be 9×9 zero-mean patches collected around the projected points $\{m_i(X_t)\}_{i \in S_{tr}}$, i.e. $Z_{i,t}(X_t) = patch(m_i(X_t))$. Similarly, observations for the *adaptive* features $Z_t^{ad} = \{Z_{i,t}\}_{i \in S_{ad}}$ will be single intensity values at the projected points $\{n_i(X_t)\}_{i \in S_{ad}}$, i.e. $Z_{i,t}(X_t) = intensity(n_i(X_t))$. Assuming conditional independence between the features given the state, we have:

$$p(Z_t|X_t) = \prod_{i \in S_{tr}} p_i(Z_{i,t}(X_t)) \cdot \prod_{i \in S_{ad}} p_i(Z_{i,t}(X_t))$$

where the appearance model p_i for the *trained* features was described in Eq. (2). For the *adaptive* features, we assume a similar model, i.e.

$$p_i(Z_{i,t}(X_t)) \propto e^{-\rho(Z_{i,t}(X_t)-\mu_{i,t},\tau_{ad})}$$

where $\mu_{i,t}$ is the current appearance template of the feature. This template is recursively updated after each time step, according to:

$$\mu_{i,t+1} = (1-\lambda_U) \cdot \mu_{i,t} + \lambda_U \cdot Z_{i,t}(\hat{X}_t)$$

where λ_U is the adaptation rate (set to 0.1 in our experiments).

Dynamics modeling: We assume conditional independance between the state components, and for each component a null velocity dynamical model. We thus have:

$$p(X_t|\hat{X}_{t-1}) = \prod_{i=1:N_p} \mathcal{N}(X_{i,t}; \hat{X}_{i,t-1}, \sigma_{d,i})$$

where N_p is the number of components of the state, $X_{i,t}$ denotes the i^{th} component of X_t, and $\{\sigma_{d,i}\}_{i=1:N_p}$ are the noise standard deviations. They define how large we assume the difference in the state between two successive frames can be.

Occlusion handling: In practice, instead of maximizing the posterior defined in Eq. (3), we minimize its negative logarithm. In addition, because the face model is 3D, we can infer if the feature points might be occluded from the geometry of the mesh and the

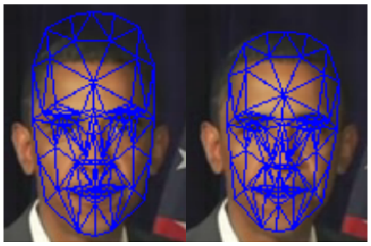

Fig. 3. Final fit (enlarged) on frame 1276 of the Speech video sequence with the Adaptive (left) and Hybrid (right) methods. Using the *adaptive* features alone is prone to drift (in this example the eyes are fit on the eyebrows) while the combination of our *trained* and *adaptive* features makes our tracker both accurate and robust.

head pose. To do so, we introduce for each feature i a *visibility* factor $v_i(X)$. This factor will be 1 when the feature is visible under pose X, and 0 otherwise. In practice, to avoid discontinuities in our objective function, we use a smoothed version: the visibility factor is defined as $v_i(X) = f_{\tau_v}(\vec{n_i}.\vec{z})$, where $\vec{n_i}$ is the normal to the mesh triangle the feature i belongs to, $\vec{z}$ the direction of the z axis, and f_{τ_v} a sigmoid function with its inflexion point at $\tau_v = 0.1$. The *visibility* of a feature point i is taken into account in the likelihood terms of the error function so that the weight of a feature varies with its *visibility*. Thus the error to be minimized is:

$$E(X_t, Z_{1:t}) = -\sum_{i\in S_{tr}} v_i(X_t).log(p_i(Z_{i,t}(X_t)))$$
$$-\sum_{i\in S_{ad}} v_i(X_t).log(p_i(Z_{i,t}(X_t))) - \sum_{i=1}^{N_p} log(p(X_{i,t}|\hat{X}_{i,t-1}))\ .$$

Optimization method: Minimization of the above equation is performed by the downhill simplex method, a non-linear and iterative optimization method. It does not require to derivate the error function (which would be difficult to extract in our case) and it maintains multiple hypothesis (which ensures robustness) during the optimization phase. The dimension of the state space being quite large, the optimization is done in two steps: we first run the optimization algorithm to estimate the pose parameters Θ_t, then we estimate the whole state X_t.

4. EXPERIMENTS AND RESULTS

The proposed algorithm currently processes an average of 3 frames per second. So far we made no attempts to make it run in real-time, but we believe that the frame rate could be much higher. Videos illustrating the results presented below can be found at `http://www.idiap.ch/~slefevre/Videos_ICME09.htm`.

4.1. Head pose estimation - the BUFT database

The BUFT database [4] is composed of two datasets in which the subjects perform free head motion (including translations and both in-plane and out-of-plane rotations). Each video is 200 frames long and has a resolution of 320 × 240. Ground truth was collected via a "Flock of Birds" 3D magnetic tracker. The Uniform-light dataset contains 45 video sequences (five subjects, nine sequences each). The Varying-light dataset contains 27 video sequences (three subjects, nine sequences each) taken under changing illumination (e.g. illumination changing only on one side of the face).

Performance measurements: For any frame in a video sequence, we define the estimation error e_i as:

$$e_i^2 = e_{pan,i}^2 + e_{tilt,i}^2 + e_{roll,i}^2 \tag{4}$$

where $e_{pan,i}$ is the estimation error for the pan angle for frame i. It is defined by $e_{pan,i} = |\theta_{y,i} - \hat{\theta}_{y,i}|$, where $\theta_{y,i}$ is the ground truth for the pan angle for frame i, and $\hat{\theta}_{y,i}$ is the estimate for the pan angle for frame i. The definitions are similar for the tilt and roll.

We can define the robustness of a tracker as the number N_s of frames successfully tracked. For the sake of our analysis, we defined the track as lost when e_i exceeds a certain threshold. This threshold was set manually by inspecting different sequences where the track was lost and then measuring the corresponding error as given by Eq. (4). A similar procedure was applied in [4]. We will call P_s the percentage of frames successfully tracked over all the video sequences. The accuracy of a tracker is defined as the mean pan, tilt and roll angle errors over the set of all tracked frames S_s:

$$E_m = \frac{1}{3}(E_{pan} + E_{tilt} + E_{roll})$$

where $E_{pan} = \frac{1}{N_s}\sum_{i\in S_s} e_{pan,i}$ (and similarly for the tilt and roll).

Results: We compared the performances of six trackers; the results are shown in Table 1. The "Hybrid" approach corresponds to the implementation of the method described in this paper. The "Trained" (respectively "Adaptive") approach corresponds to the same modeling but only taking into account the *trained* (respectively *adaptive*) features.

The results obtained by the Hybrid method (see examples in Fig. 2) are noticeably better than the results of [4], regarding both robustness and accuracy on the tracked frames. The performances are comparable to those of the methods proposed in [6, 5]. But our major achievement is to handle the much more challenging Varying-light dataset without loss of robustness and accuracy, while these previous methods have not been demonstrated on this second dataset.

Using *trained* features only (i.e. the Trained method) results in a loss of robustness and precision. We observed that every failure of the tracker was caused by a large pan rotation. The tracker relies on 26 features all gathered at mainly three regions of the face (the two eyes and the mouth). Thus the loss of information caused by the disappearance of a few features on one face side (e.g. one eye when the pan becomes superior to 30 °) creates a large uncertainty around the pan axis; this is enough to make the tracker lose track.

The method relying on *adaptive* features only (i.e. the Adaptive method) performs quite well on both datasets, but the pose estimation is less precise than the Hybrid method. Furthermore this method is prone to drift. This cannot be seen on the sequences from the BUFT database as they are too short, but when applied to longer sequences we can observe that the tracker starts to drift at some point. An example is shown in Fig. 3: by the end of the sequence the eyes are fit on the eyebrows, the eyebrows are fit on the forehead and the top of the head is fit on the background.

4.2. Facial actions estimation

Unfortunately, no ground truth is available for facial actions videos, and we measured the results quality only visually. The reader is invited to refer to the videos found at the url given above. We tested our system on two longer video sequences containing facial expressions. The first one is the Talking Face video from PRIMA - INRIA Rhone-Alpes, a 5000 frames long video of a person engaged in a conversation. The second one is a 2500 frames long speech from a politician in a TV broadcast. Facial actions are mostly located

	Uniform-light dataset					Varying-light dataset				
Approach	P_s	E_{pan}	E_{tilt}	E_{roll}	E_m	P_s	E_{pan}	E_{tilt}	E_{roll}	E_m
La Cascia et al. [4]	75%	5.3°	5.6°	3.8°	3.9°	85%	-	-	-	-
Xiao et al. [5]	100%	3.8°	3.2°	1.4°	2.8°	-	-	-	-	-
Morency et al. [6]	100%	5.0°	3.7°	2.9°	3.9°	-	-	-	-	-
Hybrid	**100%**	**4.4°**	**3.3°**	**2.0°**	**3.2°**	**100%**	**4.1°**	**3.5°**	**2.3°**	**3.3°**
Trained	95%	4.6°	4.5°	2.0°	3.7°	98%	4.1°	3.8°	2.3°	3.4°
Adaptive	100%	4.7°	3.3°	2.0°	3.3°	100%	4.9°	4.0°	2.7°	3.9°

Table 1. Comparison on the BUFT database of robustness and accuracy as defined in Section 4.1 between our approach and state-of-the-art face trackers. On the relatively simple Uniform-light dataset, our approach outperforms the method recently proposed by Morency et al. and the method proposed by La Cascia et al. It performs slightly worse than the method proposed by Xiao et al. However, it performs very well on the much more challenging Varying-light dataset, on which none of these methods were successfully demonstrated.

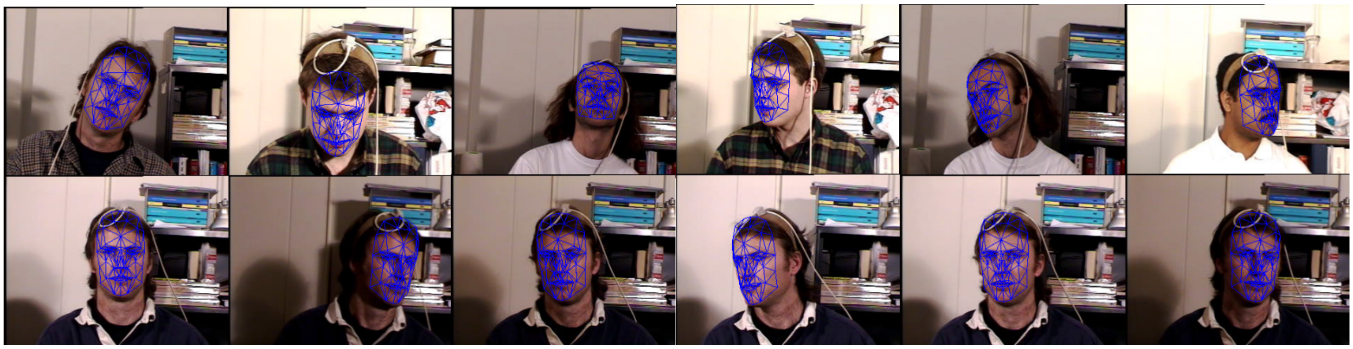

Fig. 2. Example results for the Hybrid method on the BUFT database. Top row: Images from six different sequences of the Uniform-light dataset. Bottom row: Images from one sequence of the Varying-light dataset. Example result sequences can be found on the website.

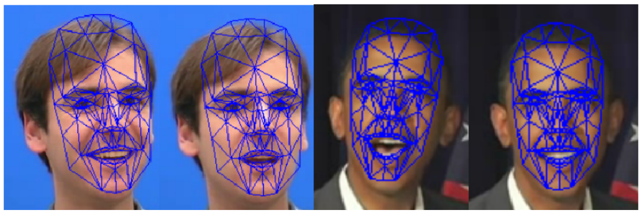

Fig. 4. Example results (enlarged) for the Hybrid method on the Talking Face and Speech video sequences.

around the mouth, the tracker is able to follow these movements quite precisely, as can be seen in Fig. 4.

5. CONCLUSION

In this paper we presented a novel algorithm for tracking robustly and precisely faces in video sequences using a deformable 3D face model. A hybrid set of features was presented, composed of *adaptive* features (pixels from a sparse facial texture whose appearance model is adapted over time) and *trained* features (illumination-invariant local structural features with a fixed appearance model). The precisiom of the first type of features is combined with the stability of the second type of features into a maximum likelihood framework. The resulting system can estimate the head pose and the facial actions under constant or changing illumination, with no restriction to near-frontal views. We demonstrated that our method outperforms state-of-the art systems when tested on the Boston University Face Tracking database (on both Uniform-light and Varying-light datasets). We also proved the stability of the system as well as its ability to track facial actions in long real video sequences.

6. REFERENCES

[1] T.F. Cootes, G.J. Edwards, and C.J. Taylor, “Active appearance models,” in *ECCV*, 1998, vol. 2.

[2] R. Gross, I. Matthews, and S. Baker, “Active appearance models with occlusion,” *IVC*, vol. 24, no. 6, pp. 593–604, 2006.

[3] J. Xiao, S. Baker, and I. Matthews, “Real-time combined 2d+3d active appearance models,” in *CVPR*, 2004.

[4] 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,” in *PAMI*, 2000, vol. 22.

[5] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn, “Robust full-motion recovery of head by dynamic templates and re-registration techniques,” *Int. J. of Imag. Syst. and Technol.*, vol. 13, no. 1, pp. 85–94, 2003.

[6] L.-P. Morency, J. Whitehill, and J. Movellan, “Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation,” in *AFGR*, 2008.

[7] L. Vacchetti, V. Lepetit, and P. Fua, “Stable real-time 3d tracking using online and offline information,” *PAMI*, vol. 26, no. 10, pp. 1385–1391, 2004.

[8] M. Lozano and K. Otsuka, “Real-time visual tracker by stream processing,” *J. of Sign. Process. Syst.*, pp. 674–679, 2008.

[9] J. Ahlberg, “Candide 3 - an updated parameterised face,” Tech. Rep. LiTH-ISY-R-2326, Linköping University, Sweden, 2001.

[10] F. Dornaika and F. Davoine, “On appearance based face and facial action tracking,” in *Trans. Circ. and Syst. for Video Technol.*, 2006, vol. 16.

[11] Y. Chen and F. Davoine, “Simultaneous tracking of rigid head motion and non-rigid facial animation by analyzing local features statistically,” in *BMVC*, 2006, vol. 2.

[12] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *IJCV*, vol. 38, no. 2, pp. 99–127, 2000.

[13] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.