Simultaneous facial action tracking and expression recognition using a particle filter

Fadi Dornaika, Franck Davoine

Simultaneous facial action tracking and expression recognition using a particle filter

Fadi Dornaika
Computer Vision Center
Autonomous University of Barcelona
08193 Bellaterra, Barcelona, SPAIN
dornaika@cvc.uab.es

Franck Davoine
HEUDIASYC Mixed Research Unit, CNRS/UTC
Compiègne University of Technology
60205 Compiègne, FRANCE
fdavoine@hds.utc.fr

Abstract

The recognition of facial gestures and expressions in image sequences is an important and challenging problem. Most of the existing methods adopt the following paradigm. First, facial actions/features are retrieved from the images, and then facial expressions are recognized based on the retrieved temporal parameters. Unlike this main stream, this paper introduces a new approach allowing the simultaneous recovery of facial actions and expression using a particle filter adopting multi-class dynamics that are conditioned on the expression. For each frame in the video sequence, our approach is split in two consecutive stages. In the first stage, the 3D head pose is recovered using a deterministic registration technique based on Online Appearance Models. In the second stage, the facial actions as well as the facial expression are simultaneously recovered using the stochastic framework with mixed states. The proposed fast scheme is either as robust as existing ones or more robust with respect to many regards. Experimental results show the feasibility and robustness of the proposed approach.

1. Introduction

Computational facial expression analysis is a challenging research topic in computer vision. It is required by many applications such as human-computer interaction and computer graphic animation. Automatic facial expression recognition did not really start until the 1990s. To classify expressions in still images, many techniques have been proposed such as those based on Neural Nets and Gabor wavelets [1]. Recently, more attention has been given to modeling facial deformation in dynamic scenarios. Still image classifiers use feature vectors related to a single frame to perform classification. Temporal classifiers try to capture the temporal pattern in the sequence of feature vectors related to each frame such as the Hidden Markov Models based methods [3]. A survey on facial expression recognition methods can be found in [7]. Many developed systems have relied on the facial motion encoded by a dense flow between successive image frames. However, flow estimates are easily disturbed by the illumination changes and non-rigid motion. The dominant paradigm involves computing a time-varying description of facial actions/features from which the expression can be recognized, that is, the tracking process is first performed before the recognition process [5, 11]. However, the results of both processes affect each other in some way. Therefore, one expects to gain more robustness if both tasks are simultaneously performed. Such robustness is required when perturbing factors may affect the input data such as partial occlusions, ultra rapid motions, and video streaming discontinuity. Although the idea of merging tracking and recognition is not new, our work addresses two complicated tasks, namely tracking the facial actions and recognizing the expression over time in a monocular video sequence. In the literature, the simultaneous tracking and recognition was used in simple cases. For example, [10] employs a particle filter-based algorithm for tracking and recognizing the motion class of a juggled ball in 2D. Another example is given in [12]. This work proposes a framework allowing the simultaneous tracking and recognizing of human faces using a particle filtering method. The recognition consists in determining the person identity which is fixed for the whole probe video. They utilized a mixed state vector formed by the 2D global face motion (affine transform) and an identity variable. However, this work does not address the facial deformation nor the facial expression recognition.

The rest of the paper is organized as follows. Section 2 describes the deformable 3D face model that we use to create shape-free facial patches from input images. Section 3 describes the problem we are focusing on. It presents the adaptive observation model as well as the facial action

dynamical models. Section 4 describes the proposed approach, that is, (i) the recovery of the 3D head pose by a deterministic registration technique, and (ii) the simultaneous recovery of facial actions and expression using a stochastic framework. Experimental results are given in Section 5.

2. Modeling faces

A deformable 3D model. In our study, we use the 3D face model *Candide*. This 3D deformable wireframe model was first developed for the purpose of model-based image coding and computer animation. The 3D shape of this wireframe model is directly recorded in coordinate form. It is given by the coordinates of the 3D vertices $\mathbf{P}_i, i = 1, \ldots, n$ where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector $\mathbf{g}$; the concatenation of the 3D coordinates of all vertices $\mathbf{P}_i$. The vector $\mathbf{g}$ is written as:

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\,\tau_{\mathbf{a}} \tag{1}$$

where $\mathbf{g}_s$ is the static shape of the model, and $\tau_{\mathbf{a}}$ is the animation control vector. Thus, the state of the 3D wireframe model is given by the 3D head pose parameters (three rotations and three translations) and the control vector $\tau_{\mathbf{a}}$. In this study, we use six modes for the facial Animation Units (AUs) matrix $\mathbf{A}$: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer and outer eyebrow raiser. These AUs are enough to cover most common facial animations (mouth and eyebrow movements). Moreover, they are essential for conveying emotions.

The state is given by the 12-dimensional vector $\mathbf{b}$:

$$\mathbf{b} = [\theta_x, \ \theta_y, \ \theta_z, \ t_x, \ t_y, \ t_z, \ {\tau_{\mathbf{a}}}^T \ ]^T \tag{2}$$

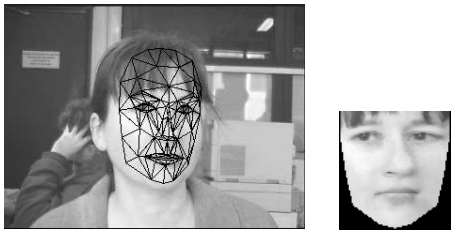
$$= [\mathbf{h}^T, \ {\tau_{\mathbf{a}}}^T \ ]^T \tag{3}$$

where the vector $\mathbf{h}$ represents the six degrees of freedom associated with the 3D head pose.

Shape-free facial patches. A face texture is represented as a shape-free texture (geometrically normalized image). The geometry of this image is obtained by projecting the static shape $\mathbf{g}_s$ using a centered frontal 3D pose onto an image with a given resolution. The texture of this geometrically normalized image is obtained by texture mapping from the triangular 2D mesh in the input image (see figure 1) using a piece-wise affine transform, $\mathcal{W}$. The warping process applied to an input image $\mathbf{y}$ is denoted by:

$$\mathbf{x}(\mathbf{b}) = \mathcal{W}(\mathbf{y}, \mathbf{b}) \tag{4}$$

where $\mathbf{x}$ denotes the shape-free texture patch and $\mathbf{b}$ denotes the geometrical parameters. Several resolution levels can be chosen for the shape-free textures. The reported results are obtained with a shape-free patch of 5392 pixels. Regarding photometric transformations, a zero-mean unit-variance normalization is used to partially compensate for contrast variations. The complete image transformation is implemented as follows: (i) transfer the texture $\mathbf{y}$ using the piecewise affine transform associated with the vector $\mathbf{b}$, and (ii) perform the grey-level normalization of the obtained patch.



(a) **(b)**

Figure 1. **(a)** an input image with correct adaptation. **(b)** the corresponding shape-free facial image.

3. Background and problem formulation

Given a video sequence depicting a moving head/face, we would like to recover, for each frame, the 3D head pose, the facial actions encoded by the control vector $\tau_{\mathbf{a}}$ as well as the facial expression. In other words, we would like to estimate the vector $\mathbf{b}_t$ (equation 3) at time t in addition to the facial expression given all the observed data until time t, denoted $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. In a tracking context, the model parameters associated with the current frame will be handed over to the next frame. Since the facial expression can be considered as a random discrete variable, we should append to the continuous state vector $\mathbf{b}_t$, a discrete state component γ_t to make a mixed state:

$$\begin{pmatrix} \mathbf{b}_t \\ \gamma_t \end{pmatrix} \tag{5}$$

$\gamma_t \in \mathcal{E} = \{1, 2, \ldots, N_\gamma\}$ is the discrete component of the state, drawn from a finite set of integer labels. Each integer label represents one of the six universal expressions: surprise, disgust, fear, joy, sadness and anger. In our study we adopt these facial expressions together with the neutral one, that is, N_γ is set to 7. There is another useful representation of the mixed state which is given by:

$$\begin{pmatrix} \mathbf{h}_t \\ \mathbf{a}_t \end{pmatrix} \tag{6}$$

where $\mathbf{h}_t$ denotes the 3D head pose parameters, and $\mathbf{a}_t$ the facial actions appended with the expression label γ_t, i.e. $\mathbf{a}_t = [{\tau_{\mathbf{a}}}^T, \gamma_t]^T$.

3.1. Adaptive observation model

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

$$\hat{\mathbf{x}}_t = \mathbf{x}(\hat{\mathbf{b}}_t) = \mathcal{W}(\mathbf{y}_t, \hat{\mathbf{b}}_t) \tag{7}$$

The estimation of $\hat{\mathbf{b}}_t$ from the sequence of images will be presented in the next Section. $\hat{\mathbf{b}}_0$ is initialized manually, according to the face in the first video frame.

The appearance model associated to the shape-free facial patch at time t, A_t, is time-varying on that it models the appearances present in all observations $\mathbf{x}$ up to time $(t-1)$. The appearance model A_t obeys a Gaussian with a center μ and a variance σ. Notice that μ and σ are vectors composed of d components/pixels (d is the size of $\mathbf{x}$) that are assumed to be independent of each other. In summary, the observation likelihood at time t is written as

$$p(\mathbf{y}_t|\mathbf{b}_t) = p(\mathbf{x}_t|\mathbf{b}_t) = \prod_{i=1}^{d} \mathbf{N}(x_i; \mu_i, \sigma_i) \tag{8}$$

where $\mathbf{N}(x; \mu_i, \sigma_i)$ is the normal density.

It can be shown that the appearance model parameters, *i.e.*, μ and σ can be updated using the following equations (see [9] for more details on Online Appearance Models):

$$\mu_{t+1} = (1-\alpha)\,\mu_t + \alpha\,\hat{\mathbf{x}}_t \tag{9}$$

$$\sigma^2_{t+1} = (1-\alpha)\,\sigma^2_t + \alpha\,(\hat{\mathbf{x}}_t - \mu_t)^2 \tag{10}$$

In the above equations, all μ's and σ^2's are vectorized and the operation is element-wise. This technique, also called recursive filtering, is simple, time-efficient and therefore, suitable for real-time applications. The effect of the above equations is that the appearance A_t summarizes the past observations under an exponential envelop [12].

3.2. Facial action dynamical models

Corresponding to each basic expression class, γ, there is a stochastic dynamical model describing the temporal evolution of the facial actions $\tau_{\mathbf{a}(t)}$. It is supposed to be a second-order Markov model. For each basic expression γ, we associate a Gaussian Auto-Regressive Process (ARP) defined by

$$\tau_{\mathbf{a}(t)} = \mathbf{A}_2^{\gamma}\,\tau_{\mathbf{a}(t-2)} + \mathbf{A}_1^{\gamma}\,\tau_{\mathbf{a}(t-1)} + \mathbf{d}^{\gamma} + \mathbf{B}^{\gamma}\,\mathbf{w}_{(t)} \tag{11}$$

in which $\mathbf{w}_{(t)}$ is a vector of 6 independent random $\mathcal{N}(0,1)$ variables. The dynamical parameters of the model are: (i) deterministic parameters $\mathbf{A}_1^{\gamma}, \mathbf{A}_2^{\gamma}$, and $\mathbf{d}^{\gamma}$, and stochastic parameters $\mathbf{B}^{\gamma}$, which determine the coupling of $\mathbf{w}_{(t)}$ into the vector $\tau_{\mathbf{a}(t)}$. The above model can be used in predicting the process from the two most recent values. The predicted value at time t obeys a multivariate Gaussian centered at the deterministic value of (11) with $\mathbf{B}^{\gamma}\mathbf{B}^{\gamma T}$ being its covariance matrix.

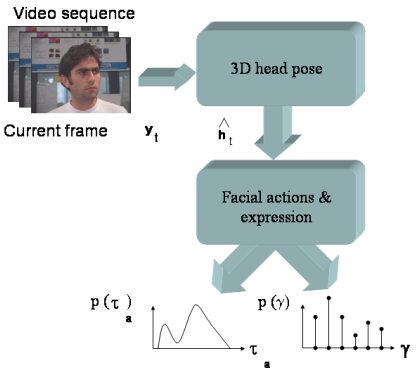


Figure 2. The proposed two-stage approach. In the first stage, the 3D head pose is computed using a deterministic registration technique. In the second stage, the facial actions and expression are simultaneously estimated using a stochastic technique involving multi-class dynamics.

Learning the second-order auto-regressive models Given a training sequence $\tau_{\mathbf{a}(1)}, \cdots, \tau_{\mathbf{a}(T)}$ (provided for instance by our deterministic tracker [6]), belonging to the same expression γ, it is well-known that a Maximum Likelihood estimator provides a closed-form solution for the corresponding model parameters $\mathbf{A}_1^{\gamma}, \mathbf{A}_2^{\gamma}, \mathbf{d}^{\gamma}$, and $\mathbf{B}^{\gamma}$ [2, 10].

3.3. The transition matrix

Our framework requires a transition matrix $\mathbf{T}$ whose entries $T_{\gamma',\gamma}$ describe the probability of transition between two expression labels γ' and γ. Although these entries can be learned from training videos, we have found that a realistic setting works well. We adopt a 7×7 symmetric matrix whose diagonal elements are close to one. The rest of the percentage is distributed equally among the expressions. In this model, transitions from one expression to another expression without going through the neutral one are allowed.

4. Approach

Since at any given time, the 3D head pose parameters are independent from the facial expression and its actions, our basic idea is to split the estimation of the unknown parameters into two main stages. For each input video frame $\mathbf{y}_t$, these two stages are invoked in order to recover the mixed

state $[\ \mathbf{h}_t^T,\ \ \mathbf{a}_t^T\ ]^T$. Our proposed approach is illustrated in Figure 2. In the first stage, the six degrees of freedom associated with the 3D head pose (encoded by the vector $\mathbf{h}_t$) are recovered using a deterministic registration technique similar to our proposed appearance-based tracker [6]. In the second stage, the facial actions and the facial expression (encoded by the vector $\mathbf{a}_t = [\tau_{\mathbf{a}(t)}^T, \gamma_t]^T$) are simultaneously estimated using a stochastic framework based on a particle filter. Since $\tau_{\mathbf{a}(t)}$ and γ_t are highly correlated, their simultaneous estimation will be more robust and accurate than methods estimating them in sequence. In the following, we present the estimation of the parameters associated with the current frame $\mathbf{y}_t$.

4.1. 3D head pose

The purpose of this stage is to estimate the six degrees of freedom associated with the 3D head pose at frame t, that is, the vector $\mathbf{h}_t$. Our basic idea is to recover the current 3D head pose parameters from the previous 12-vector $\hat{\mathbf{b}}_{t-1} = [\hat{\theta}_{x(t-1)}, \hat{\theta}_{y(t-1)}, \hat{\theta}_{z(t-1)}, \hat{t}_{x(t-1)}, \hat{t}_{y(t-1)}, \hat{t}_{z(t-1)}, \hat{\tau}_{\mathbf{a}(t-1)}]^T = [\ \hat{\mathbf{h}}_{t-1},\ \ \hat{\tau}_{\mathbf{a}(t-1)}\ ]^T$ using a region-based registration technique. In other words, the current input image $\mathbf{y}_t$ is registered with the current appearance model A_t. For this purpose, we minimize the *Mahalanobis* distance between the warped texture and the current appearance mean,

$$\min_{\mathbf{h}} e(\mathbf{h}_t) = \min d\left[\mathbf{x}(\mathbf{b}_t), \mu_t\right] = \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2 \quad (12)$$

The above criterion can be minimized using an iterative gradient-like descent method where the starting solution is set to the previous solution $\hat{\mathbf{h}}_{t-1}$. The appearance parameters, i.e. the vectors μ_t and σ_t, are known using the recursive equations (9) and (10). During the above optimization process the facial actions are set to the constant values $\hat{\tau}_{\mathbf{a}(t-1)}$. Handling outlier pixels (caused for instance by occlusions) is performed by replacing the quadratic function by the Huber's cost function [6, 8]. More details about this type of optimization can be found in [6].

The gradient matrix associated with the 3D head pose parameters, $\frac{\partial \mathbf{x}_t(\mathbf{b}_t)}{\partial \mathbf{h}}$, is approximated by numerical differences similarly to [4]. It is computed for each input frame.

4.2. Simultaneous facial actions and expression

In this stage, our goal is to simultaneously infer the facial actions as well as the expression label associated with the current frame t given (i) the observation model (Eq.(8)), (ii) the dynamics associated with each expression (Eq.(11)), and (iii) the 3D head pose for the current frame computed by the deterministic approach (see Section 4.1). This will be performed using a particle filter paradigm. Thus, the statistical inference of such paradigm will provide a posterior distribution for the facial actions $\tau_{\mathbf{a}(t)}$ as well as a Probability Mass function for the facial expression γ_t .

Since the 3D head pose, $\hat{\mathbf{h}}_t$ is already computed, we are left with the mixed state $\mathbf{a}_t = [\tau_{\mathbf{a}(t)}^T, \gamma_t]^T$. The dimension of the vector $\mathbf{a}_t$ is 7. Here we will employ a particle filter algorithm allowing the recursive estimation of the posterior distribution $p(\mathbf{a}_t|\mathbf{x}_{1:(t)})$ using a particle set. This is approximated by a set of J particles $\{(\mathbf{a}_t^{(0)}, w_t^{(0)}), \ldots, (\mathbf{a}_t^{(J)}, w_t^{(J)})\}$. Once this distribution is known the facial actions as well as the expression can be inferred using some loss function such as the MAP or the mean. Figure 3 illustrates the proposed two-stage approach. More precisely, it shows how the current posterior $p(\mathbf{a}_t|\mathbf{x}_{1:(t)})$ can be inferred from the previous posterior $p(\mathbf{a}_{t-1}|\mathbf{x}_{1:(t-1)})$ using a particle filter algorithm.

On a 3.2 GHz PC, a non-optimized C code of the approach computes the 3D head pose parameters in 25 ms and the facial actions/expression in 31 ms, if the patch resolution is 1310 pixels and the number of particles is 100.

5. Experimental results

Figure 4 displays the application of the proposed approach to a 748 frame-long test video sequence. The top plot illustrates the probability of each expression as a function of time (frames). For the sake of clarity, only four of seven curves are displayed. The bottom of this figure shows the tracking results associated with three frames. The upper left corner of these frames depicts the appearance mean and the current shape-free facial texture. The middle of this Figure illustrates the segmentation of the used test video by merging the frame-wise expression probabilities associated with non-neutral frames. For this sequence, the maximum probability was correctly indicating the actual displayed expression.

In the above experiment, the total number of particles is set to 200. We have found that there is no significant difference in the estimated facial actions and expressions when the tracking is run with 100 particles, which is due to the use of learned multi-class dynamics. Figure 5 shows the tracking results associated with two test video sequences depicting out-of-plane head motions.

One class dynamics versus multi-class dynamics. In order to show the advantage of using multi-class dynamics and mixed states, we have conducted the following experiment. We have used a particle filter for tracking the facial actions. However, this time the state is only composed by the facial actions and the dynamics are replaced with a simple noise model, i.e. motion is modelled as a random noise. Figures 6.**a** and 6.**b** show the tracking results associated with the same input frame: **(a)** displays the tracking results

1. **Initialization.** $t = 0$:
   - Initialize the 3D head pose $\hat{\mathbf{h}}_0$
   - Generate J state samples $\mathbf{a}_0^{(1)}, \ldots, \mathbf{a}_0^{(J)}$ according to some prior density $p(\mathbf{a}_0)$ and assign them identical weights, $w_0^{(1)} = \ldots = w_0^{(J)} = 1/J$
2. **Tracking.** At time step $t \leftarrow t + 1$, get the input frame $\mathbf{y}_t$. Compute the corresponding 3D head pose, $\hat{\mathbf{h}}_t$, using the deterministic method outlined in Section 4.1. We have J weighted particles $(\mathbf{a}_{t-1}^{(j)}, w_{t-1}^{(j)})$ that approximate the posterior distribution $p(\mathbf{a}_{t-1}|\mathbf{x}_{1:(t-1)})$ at previous time step
   (a) Resample the particles proportionally to their weights, *i.e.* particles with high weights are duplicated and particles with small weights are removed. Resampled particles have the same weights
   (b) Draw J particles $\mathbf{a}_t^{(j)}$ according to the dynamic model $p(\mathbf{a}_t|\mathbf{a}_{t-1} = \mathbf{a}_{t-1}^{(j)})$. The obtained new particles approximate the predicted distribution $p(\mathbf{a}_t|\mathbf{x}_{1:(t-1)})$. For multi-class dynamics and mixed states this is done in two steps
   Discrete: Draw an expression label $\gamma_t^{(j)} = \gamma \in \mathcal{E} = \{1, 2, \ldots, N_\gamma\}$ with probability $T_{\gamma',\gamma}$, where $\gamma' = \gamma_{t-1}^{(j)}$
   Continuous: Compute $\tau_{\mathbf{a}(t)}^{(j)}$ as
   $$\tau_{\mathbf{a}(t)}^{(j)} = \mathbf{A}_2^\gamma \tau_{\mathbf{a}(t-2)}^{(j)} + \mathbf{A}_1^\gamma \tau_{\mathbf{a}(t-1)}^{(j)} + \mathbf{d}^\gamma + \mathbf{B}^\gamma \mathbf{w}_{(t)}^{(j)}$$
   where $\gamma = \gamma_t^{(j)}$ and $\mathbf{w}_{(t)}^{(j)}$ is a 6-vector of standard normal random variables
   (c) Compute the shape-free texture $\mathbf{x}(\mathbf{b}_t^{(j)})$ according to (4) where $\mathbf{b}_t^{(j)} = [\ \hat{\mathbf{h}}_t^T, \quad \tau_{\mathbf{a}(t)}^{(j)T}\ ]^T$
   (d) Weight each new particle proportionally to its likelihood:
   $$w_t^{(j)} = \frac{p(\mathbf{x}_t|\mathbf{b}_t^{(j)})}{\sum_{m=1}^J p(\mathbf{x}_t|\mathbf{b}_t^{(m)})}$$
   The new set approximates the posterior $p(\mathbf{a}_t|\mathbf{x}_{1:t})$
   (e) Set the probability of each basic expression $\gamma^\star \in \mathcal{E} = \{1, 2, \ldots, N_\gamma\}$ to
   $$P(\gamma^\star) = \sum_{m=1}^J \begin{cases} w_t^{(m)} & \text{if } \gamma_t^{(m)} = \gamma^\star \\ 0 & \text{otherwise} \end{cases}$$
   (f) Set the facial actions to $\hat{\tau}_{\mathbf{a}(t)} = \sum_{m=1}^J w_t^{(m)} \tau_{\mathbf{a}(t)}^{(m)}$
   (g) Set the geometrical parameters as $\hat{\mathbf{b}}_t = [\hat{\mathbf{h}}_t^T, \quad \hat{\tau}_{\mathbf{a}(t)}^T\ ]^T$
   (h) Based on $\hat{\mathbf{b}}_t$, update the appearance (Eqs. (7), (9), (10)) as well as the 3D pose gradient matrix. Go to 2.

Figure 3. Inferring the 3D head pose, the facial actions and expression. A particle filter is used for the simultaneous recovery of the facial actions and expression.

obtained with a particle filter adopting a single class dynamics. **(b)** displays the tracking results obtained with our proposed approach adopting the six auto-regressive models. As can be seen, by using the learned multi-class dynamics, the facial action tracking becomes considerably more accurate (see the adaptation of the mouth region in both figures).

A deterministic approach versus our proposed stochastic approach. We have cut about 40 frames from a test video. These frames overlap with a surprise transition. The altered video is then tracked using two different methods: (i) a deterministic approach based on a registration technique estimating both the head and facial action parameters [6], and (ii) our stochastic approach. Figures 7.**a** and 7.**b** show the tracking results associated with the same input frame immediately after the cut. Note the difference in accuracy between the stochastic approach **(b)** and the deterministic one **(a)** (see the eyebrow and mouth region in both figures).

A challenging example. We have created a challenging test video. For this 1600-frame long test video, we have asked our subject to display facial gestures and expressions in an arbitrary way, duration, and order. Figure 8 illustrates the probability mass distribution as a function of time. As can be seen, the surprise, joy, anger, disgust, and fear are markedly and correctly detected.

6. Conclusion

In this paper, we have proposed a stochastic framework for the simultaneous tracking of facial actions and recognition of expression. Experiments show the robustness of the proposed method. The method is view-independent and does not require any learned facial texture. The latter property makes it more flexible. The proposed method could include other facial gestures in addition to the universal expressions.

References

[1] M. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In *IEEE Int. Conference on Systems, Man and Cybernetics*, 2004.

[2] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 2000.

[3] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang. Facial expression recognition from video sequences: Temporal and static modeling. *Computer Vision and Image Understanding.*, 91(1-2):160–187, 2003.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–684, 2001.

[5] F. Dornaika and F. Davoine. Facial expression recognition in continuous videos using dynamic programming. In *IEEE International Conference on Image Processing*, 2005.

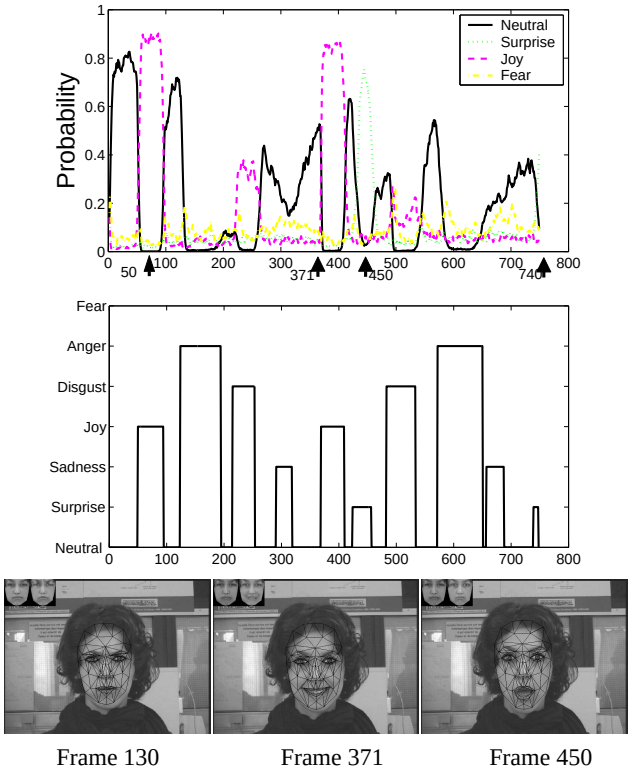


Figure 4. Simultaneous tracking and recognition associated with a 748-frame-long video sequence. The top illustrates the probability of each expression as a function of time (frames). The middle shows the video segmentation results, and the bottom the tracked facial actions associated with three frames.

[6] F. Dornaika and F. Davoine. Head and facial animation tracking using appearance-adaptive models and particle filters. In *IEEE Workshop on Real-Time Vision for Human-Computer Interaction, Washington DC*, July 2004.

[7] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[8] P. Huber. *Robust Statistics*. Wiley, 1981.

[9] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311, 2003.

[10] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.

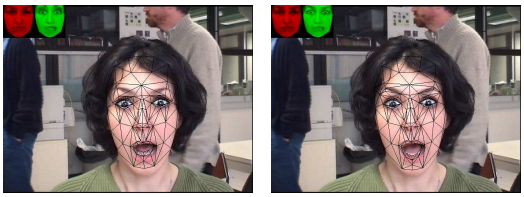
[11] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):699–714, 2005.

[12] S. Zhou, R. Chellappa, and B. Mogghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13(11):1473–1490, 2004.

Figure 5. Simultaneous tracking and recognition associated with two different video sequences depicting non-frontal head poses.

(a) **(b)**

Figure 6. Method comparison: One class dynamics **(a)** versus multi-class dynamics **(b)** (see Section 5).



(a) **(b)**

Figure 7. Method comparison: Deterministic approach **(a)** versus our stochastic approach **(b)** after a simulated video streaming discontinuity (see Section 5).

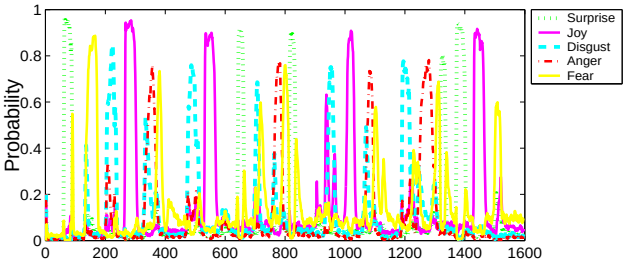


Figure 8. The probability of each expression as a function of time (frames) associated with a 1600-frame-long video.