Facial action tracking using particle filters and active appearance models

Soumya Hamlaoui and Franck Davoine

HEUDIASYC Mixed Research Unit, CNRS / Compiègne University of Technology
BP 20529, 60205 Compiègne Cedex, France
Soumya.Hamlaoui@hds.utc.fr, Franck.Davoine@hds.utc.fr

Abstract

Tracking a face and its facial features in a video sequence is a challenging problem in computer vision. In this view, we propose a stochastic tracking system based on a particle-filtering scheme. In this paradigm, the unobserved state includes global face pose and appearance parameters coding both shape and texture information of the face. The adopted observations distribution is derived from an Active Appearance Model (AAM). The transition distribution and the particles number are adaptive in the sense that they are guided by an AAM deterministic search. This optimization stage adjusts the explored area of the state space to the quality of the prediction and enables a substantial gain in computing time. The observation model uses a robust distance measure in order to account for occlusions. Experiments on real video show encouraging results.

1. Introduction

This work addresses the problem of tracking in a single video the global motion of a face as well as the local motion of its inner features, due for instance to expressions or eye blinking. This task is required in many emerging applications, like surveillance, teleconferencing, emotional computer interfaces, human-computer communication, motion capture for video synthesis, automated lipreading, driver drowsiness monitoring, etc. Face tracking poses challenging problems because of the variability of facial appearance within a video sequence, most notably due to changes in head pose, expressions, lighting or occlusions. Much research has thus been devoted to the problem of face tracking, as a specially difficult case of non-rigid object tracking. Note that in the applications targeted by this work, the person looks approximately in the direction of the camera. The face remains thus in a near frontal orientation, so that a 2D model of the face is assumed to be able to capture the expected variations.

In the object tracking literature, the following formulation of the tracking problem is conveniently used: at each time step t, the goal is to infer the unobserved state of the object, denoted $\mathbf{x}_t \in \mathscr{X}$, given all the observed data until time t, denoted $\mathbf{z}_{1:t} \equiv (\mathbf{z}_1, \dots, \mathbf{z}_t)$. When tracking a face in 2D, the unobserved state $\mathbf{x}_t$ includes motion or pose parameters like the position, scale and orientation of the face; when facial features are also tracked, the unobserved state should contain parameters describing the face inner motion. The observed data $\mathbf{z}_t$ consists of measurements derived from the current video frame, such as grey level patches, edges, or color histograms. In order to evaluate a hypothesized state, the measurements are actually only considered in the image area corresponding to the hypothesized location. For instance, the most natural measurement consists of the pixel grey level values themselves. Basically, a given state $\mathbf{x}_t$ (motion parameters) is then evaluated by comparing the motion-compensated grey level image patch $\mathbf{g}_{image}(\mathbf{z}_t, \mathbf{x}_t)$ with a grey level template face patch $\mathbf{g}_{model}$.

The tracking task then essentially consists in searching the current state $\hat{\mathbf{x}}_t \in \mathscr{X}$ that matches at best the measurements $\mathbf{z}_t$ in the current image. The tracking history $\hat{\mathbf{x}}_{1:(t-1)}$ is mainly used as a prior knowledge in order to search only a small subset of the state space $\mathscr{X}$.

In a non-probabilistic formulation of the tracking problem, the state $\hat{\mathbf{x}}_t$ is usually seeked so as to minimize an error functional $d\left[\mathbf{g}_{image}(\mathbf{z}_t, \mathbf{x}_t); \mathbf{g}_{model}\right]$, e.g. an euclidean or robust distance. Actually, in a tracking setting, the state is supposed to evolve little between consecutive time steps. The solution is thus searched as a small displacement from the previous frame estimation: $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \widehat{\partial \mathbf{x}_t}$. The optimal displacement is then typicallly obtained by a gradient-like descent method. The well-known Lucas-Kanade algorithm is a particular case of such a formulation, and has been recently generalized in [1]. Instead of being specified by a single face greylevel template $\mathbf{g}_{model}$, the face model can span a subspace of greylevel patches, learnt by principal component analysis (PCA) from a face training set. The error functional is then a distance from the image patch $\mathbf{g}_{image}(\mathbf{z}_t, \mathbf{x}_t)$ to the face subspace, usually taken to be the distance to the projection in face subspace. The subspace modeling allows to account for some variability of the global face appearance.

The eigen-tracking method is based on such a principle [3]. Using also principal component analysis, the Active Appearance Models (AAMs) encode the variations of face appearance by learning the shape and texture variations [4]. They enable thus the tracking of both global motion and inner features. In the case of AAMs, the gradient jacobian matrix is pre-computed in order to save processing time. In practice, tracking using the deterministic AAM search appears to work well while the lighting conditions remain stable and only small occlusions are present. However, large occlusions often make the AAM search converge to incorrect positions and loose track of the face.

In probabilistic formulations, the hidden state and the observations are linked by a joint distribution; this statistical framework offers rich modeling possibilities. A Markovian dynamic model describes how the state evolves through time. An observation model specifies the likelihood of each hypothesised state, i.e. the probability that the considered state may generate the observed data. Such generative models represent the variability in the motion and appearance of the object to track. Note that even the non-probabilistic minimization of an error functional can be recast as the maximization of a likelihood:

$$p(\mathbf{z}_t|\mathbf{x}_t) \propto \exp -d\left[\mathbf{g}_{image}(\mathbf{z}_t, \mathbf{x}_t); \mathbf{g}_{model}\right]$$

Based on such a generative model, Bayesian filtering methods recursively evaluate the posterior density of the target state at each time step conditionally to the history of observations until the current time.

Stochastic implementations of Bayesian filtering are generally based on sequential Monte Carlo estimation, also known as particle filtering [5]. Particle filtering approximates the posterior state density by a set of random weighted samples (particles) at each time step. The CONDENSATION algorithm consists in propagating this sample set through time using a dynamic model and in weighting each sample proportionally to its likelihood function value [7]. When compared with the analytical solution provided by the well-known Kalman filter, particle filtering has two advantages: it is not restricted to the case of linear and Gaussian models, and it can maintain multiple hypotheses of the current state, a desirable property when the background is complex and contains distracting clutter.

For video tracking, the CONDENSATION algorithm was first proposed in conjunction with edge measurements produced by an edge detector [7]. Since then, this algorithm has attracted much interest, and other kinds of measurements have given valuable variants. For instance, the color histogram yields fast, deformation- and orientation-robust tracking [8]. However, since color histograms are global, they do not allow to track the motion of internal facial features as is the goal here. A greylevel patch is used as measurement vector by Zhou et al. [10]. In order to cope with the changing appearance of the face, the likelihood is taken to be a mixture of three mean appearance templates, and the parameters of the mixture are re-estimated during the tracking. Their model considers however only global face appearance templates in the likelihood, and global motion parameters in the state vector. Ross et al. [9] propose to constantly update the appearance model of the tracked object to account for the lighting, expression and pose variations but their tracker doesn't handle occlusion well.

The AAM paradigm provides an elegant and simple way to track both the global pose and the internal facial features. The idea proposed in this paper consists in combining the AAM with the CONDENSATION stochastic search in order to augment its robustness to occlusions. Regarding existing works we are aware of combining AAM with temporal dynamics, they model facial behaviours in order to generate video-realistic animated faces (see e.g. [2]). In those papers, the tracking itself uses the AAM frame-by-frame search with no temporal dynamics.

In section 2, we recall the main principles of AAMs and particle filtering, and introduce a few related notations. In section 3, we present the proposed tracking algorithm. In section 4, experimental results are shown on real video. In section 5, we draw concluding remarks and discuss the perspectives opened by this work.

2. Background

2.1 Face Active Appearance Models

A face AAM is a statistical model which describes shape and texture variations of the human face class [4]. The appearance variability is linearly modelled by a Principal Component Analysis (PCA) of shape and texture. The set of model parameters which control the different modes of shape and texture variation are learned from a training set of annotated images. Each training image is manually annotated using a set of landmark points to outline the facial structures.

The face shape consists then of the spatial coordinates of the landmark points. The training set shapes are aligned to the Procrustes mean shape of the training set. The corresponding textures are described by the intensity of the pixels inside the area delimited by the shapes. These textures are all warped to the mean shape. A PCA is then applied to shape and texture data, denoted respectively $\mathbf{s}$ and $\mathbf{g}$. Each training face is then represented by shape and texture model parameters $\mathbf{b}_s$, $\mathbf{b}_g$:

$$\mathbf{s} = \mathbf{s}_m + \phi_s \mathbf{b}_s \qquad \mathbf{g} = \mathbf{g}_m + \phi_g \mathbf{b}_g$$

where $\mathbf{s}_m$, $\mathbf{g}_m$ are respectively the mean shape and texture, ϕ_s, ϕ_g are the eigenvectors of shape and texture covariance matrices. A third PCA is then performed on a concatenated shape and texture parameters $\mathbf{b}$, to obtain a combined model vector $\mathbf{c}$:

$$\mathbf{b} = \phi_c \mathbf{c}$$

where:

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix}$$

$\mathbf{W}_s$ is an estimated weighting matrix between shape and texture and ϕ_c is a set of eigenvectors.

From the combined appearance model vector $\mathbf{c}$, a new instance of shape and texture can be generated:

$$\mathbf{s}_{model}(\mathbf{c}) = \mathbf{s}_m + \mathbf{Q}_s \mathbf{c} \qquad \mathbf{g}_{model}(\mathbf{c}) = \mathbf{g}_m + \mathbf{Q}_g \mathbf{c}$$

In this paper, our approach is tested using person-specific appearance models. The model is trained using 20 annotated still images of the person to be tracked. Figure 1 shows the second mode variation of the combined model vector $\mathbf{c}$ through $\pm$ 1 standard deviation from the mean for one of the training persons.

Figure 1: Second combined mode of appearance variation for one training person, i.e. texture instance $\mathbf{g}_{model}(\mathbf{c})$ warped back into shape instance $\mathbf{s}_{model}(\mathbf{c})$, with c_2 = mean - std, mean, mean + std and $\mathbf{c} = (0, c_2, 0, \ldots, 0)^T$.

In order to match a target face in a given image, the shape and texture instances have to be translated, scaled and rotated. This affine transformation can be represented by a vector of four 2D pose parameters $\mathbf{p} = (t_x, t_y, \alpha, \theta)$. Those parameters denote respectively the x and y centers of gravity of the shape, and the scaling factor and inplane rotation relatively to the learnt mean shape.

The AAM paradigm provides an iterative gradient-like search technique in order to compute automatically the pose and appearance parameters $(\hat{\mathbf{p}}, \hat{\mathbf{c}})$ that best approximate the target face in the image [4]. The minimized criterion is the norm of the error vector

$$\mathbf{r}(\mathbf{p}, \mathbf{c}) = \mathbf{g}_{model}(\mathbf{c}) - \mathbf{g}_{im}(\mathbf{p}, \mathbf{c}) \tag{1}$$

where $\mathbf{g}_{model}(\mathbf{c})$ denotes the model face texture and $\mathbf{g}_{im}(\mathbf{p}, \mathbf{c})$ the image texture sampled at the hypothesized pose $\mathbf{p}$ and shape $\mathbf{s}_{model}(\mathbf{c})$. Starting from an initial guess $(\check{\mathbf{p}}, \check{\mathbf{c}})$, the optimal corrections to apply, $(\partial \mathbf{p}, \partial \mathbf{c})$, are linear functions of the error vector:

$$\partial \mathbf{p} = \mathbf{R}_{\mathbf{p}}.\mathbf{r}(\check{\mathbf{p}}, \check{\mathbf{c}}) \qquad \partial \mathbf{c} = \mathbf{R}_{\mathbf{c}}.\mathbf{r}(\check{\mathbf{p}}, \check{\mathbf{c}}) \tag{2}$$

The matrices $\mathbf{R}_{\mathbf{p}}$ and $\mathbf{R}_{\mathbf{c}}$ can be precomputed from training data, as in [4].

2.2 CONDENSATION

The CONDENSATION algorithm employs the Monte Carlo technique of factored sampling in order to recursively approximate the posterior state density. Approximation is done by means of the empirical distribution of a system of particles. The particles explore the state space following independent realizations from a state evolution model, and are redistributed according to their consistency with the observations, the consistency being measured by a likelihood function.

The sketch of the CONDENSATION algorithm is recalled in Figure 2. For a good introduction, the reader is refered to the seminal paper of Isard and Blake [7]. At Step $2d$, the state $\hat{\mathbf{x}}_t$ could also be estimated using the mean $\hat{\mathbf{x}}_t = \mathbb{E}[\mathbf{x}_t | \mathbf{z}_{1:t}] \approx \sum_{n=1}^{N} \pi_t^{(n)} \mathbf{a}_t^{(n)}$; since both estimates appeared very similar in the experiments, the MAP will be used in the following.

1. Initialization $t = 0$: Generate N state samples $\mathbf{a}_0^{(1)}, \ldots, \mathbf{a}_0^{(N)}$ according to the prior density $p(\mathbf{x}_0)$ and assign them identical weights, $\pi_0^{(1)} = \ldots = \pi_0^{(N)} = 1/N$.
2. At time step t, we have N weighted particles $(\mathbf{a}_{t-1}^{(n)}, \pi_{t-1}^{(n)})$ that approximate the posterior distribution of the state $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:(t-1)})$ at previous time step.
   (a) Resample the particles proportionally to their weights, i.e. keep only particles with high weights and remove particles with small ones. Resampled particles have the same weights.
   (b) Draw N particles according to the dynamic model $p(\mathbf{x}_t|\mathbf{x}_{t-1} = \mathbf{a}_{t-1}^{(n)})$.
   These particles approximate the predicted distribution $p(\mathbf{x}_t|\mathbf{z}_{1:(t-1)})$.
   (c) Weight each new particle proportionally to its likelihood:
   $$\pi_t^{(n)} = \frac{p(\mathbf{z}_t|\mathbf{x}_t = \mathbf{a}_{t-1}^{(n)})}{\sum_{m=1}^{N} p(\mathbf{z}_t|\mathbf{x}_t = \mathbf{a}_{t-1}^{(m)})} \tag{3}$$
   The set of weighted particles approximates the posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$.
   (d) Give an estimate of the state $\hat{\mathbf{x}}_t$ as the MAP:
   $$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{z}_{1:t}) \approx \arg\max_{\mathbf{a}_t^{(n)}} \pi_t^{(n)}$$

Figure 2: CONDENSATION algorithm.

3. Proposed scheme: AAM-based CONDENSATION

We propose to adapt the CONDENSATION algorithm to our tracking task in three aspects, each being detailed below:

- the state space spans the global and inner motion of the face;
- the observation model is based on sampled pixel grey level patches and a previously trained AAM subspace;
- the dynamics are adaptive as in [10].

3.1 State space spans global and local motion

The state vector $\mathbf{x}_t$ contains the parameters to infer about the object:

- the face global 2D pose $\mathbf{p}_t = (t_x, t_y, \alpha, \theta)^T$
- the facial actions, contained in the AAM shape and texture, which are themselves captured in a compact way by the combined appearance parameter vector $\mathbf{c}_t$. Our experiments suggest that for a person-specific AAM, retaining only the first $K = 4$ modes of the appearance parameter $\mathbf{c}_t$ allows to span the facial changes of interest and provides satisfying tracking results.

The state vector is thus of dimension $4 + K = 8$:

$$\mathbf{x}_t = (\mathbf{p}_t, \mathbf{c}_t)^T$$

3.2 AAM-based observation model

The observation model consists of the likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$, according to which the particles are weighted in the formula (3) of the CONDENSATION algorithm. This likelihood indicates the probability that a hypothesized state $\mathbf{x}_t = (\mathbf{p}_t, \mathbf{c}_t)^T$ gives rise to the observed data. This probability should be high whenever there is a good match between:

- the image patch sampled at the hypothesized pose and shape, $\mathbf{g}_{im}(\mathbf{p}_t, \mathbf{c}_t)$
- the hypothesized appearance of the face, given by the model texture $\mathbf{g}_{model}(\mathbf{c}_t)$.

The adopted likelihood function has thus the following form:

$$p(\mathbf{z}_t|\mathbf{x}_t) = C \exp -d\left[\mathbf{g}_{model}(\mathbf{c}_t); \mathbf{g}_{im}(\mathbf{p}_t, \mathbf{c}_t)\right] \tag{4}$$

where C is the normalizing constant of this distribution, and the texture distance $d\,[;]$ is an error measure, summed over all L pixels of both textures:

$$d\left[\mathbf{g}; \mathbf{g}'\right] = \sum_{\ell=1}^{L} \rho\left(\frac{g_\ell - g'_\ell}{\sigma_\ell}\right) \tag{5}$$

This error is weighted by the standard deviation σ_ℓ of each pixel, in order to account for face parts with higher variability. The error function $\rho()$ can be chosen in different ways:

- a simple square error function $\rho(x) = \frac{1}{2}x^2$ yields a weighted euclidean distance $d\,[;]$ and a gaussian density $p(\mathbf{z}_t|\mathbf{x}_t)$;
- a robust error function can be used instead (see e.g. [6]); in our experiments, we tested the following function:

$$\rho(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \le h \\ h|x| - \frac{1}{2}h^2 & \text{if } |x| > h \end{cases} \tag{6}$$

  where h is a fixed threshold above which the difference $|x|$ is considered to be an outlier. Such a robust measure reduces the influence of occluded pixels, which would otherwise dominate the total error measure (5) and rule out a potentially good state candidate.

3.3 Adaptive dynamics

The state transition model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is used in the CONDENSATION algorithm in order to draw the particles approximating the predicted distribution (Step 2*b* of Figure 2). Following the ideas developed in Zhou *et al.* [10], the dynamics used here are adaptive by having the following model for state evolution:

$$\mathbf{x}_t = \hat{\mathbf{x}}_{t-1} + \mathbf{v}_t + \mathbf{S}_t\mathbf{u} \tag{7}$$

where

- $\hat{\mathbf{x}}_{t-1}$ is the estimate of the state vector at the previous time step,
- the velocity $\mathbf{v}_t$ indicates the predicted shift in pose / appearance
- the random component $\mathbf{u}$ is a vector of $4 + K = 8$ independent normal random variates having zero mean and unit variance
- the diagonal matrix $\mathbf{S}_t = \mathrm{diag}(\sigma_t^{(t_x)}, \ldots, \sigma_t^{(c_4)})$ specifies the standard deviation of the random draw for each pose/appearance parameter.

The predicted shift $\mathbf{v}_t = (\partial\mathbf{p}, \partial\mathbf{c})^T$ is obtained by an AAM search in the current frame[1], using the update equations (2). The search is initialized with the previous state estimate $(\check{\mathbf{p}}, \check{\mathbf{c}}) = (\hat{\mathbf{p}}_{t-1}, \hat{\mathbf{c}}_{t-1}) = \hat{\mathbf{x}}_{t-1}$. The predicted state yielded by this search will be denoted by $\tilde{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \mathbf{v}_t$. This deterministic search aims to focus the particle drawing in a region that is most likely to contain good candidates, and thus reduce the volume of the state space to explore.

According to the state transition model (7), pose / appearance parameters are drawn around the predicted state $\tilde{\mathbf{x}}_t$ with dispersions (standard deviations) given by $\mathbf{S}_t$. Those dispersions should be adaptive: the generated particles need to explore a wide area around $\tilde{\mathbf{x}}_t$ only when the predicted state $\tilde{\mathbf{x}}_t$ gives a "poor" solution. To measure the quality of the predicted state

[1] In [10], the shift in global pose $\mathbf{v}_t = \partial\mathbf{p}$ is also computed by using the current frame, a principle of constant brightness constraint being applied to calculate the predicted motion.

$\tilde{\mathbf{x}}_t = (\tilde{\mathbf{p}}_t, \tilde{\mathbf{c}}_t)$, we average the texture error over the L pixels of the textures:

$$\varepsilon_t = \frac{2}{L} \sum_{\ell=1}^{L} \rho \left(\frac{g_\ell^{model}(\tilde{\mathbf{c}}_t) - g_\ell^{im}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{c}}_t)}{\sigma_\ell} \right) \quad (8)$$

Since this error is a measure of variance, its square root $\sqrt{\varepsilon_t}$ is used to scale the standard deviations of the pose/appearance drawing:

$$(\sigma_t^{(t_x)}, \ldots, \sigma_t^{(c_4)})^T = R_t(\sigma_0^{(t_x)}, \ldots, \sigma_0^{(c_4)})^T$$

where $(\sigma_0^{(t_x)}, \ldots, \sigma_0^{(c_4)})$ are fixed reference standard deviations, and R_t is a diagonal matrix:

$$R_t = diag(R_t^{(t_x)}, \ldots, R_t^{(c_4)})$$

where $R_t^{(i)}$ are scaling factors associated to the 8 components of the state vector (subscripted by the index i) and proportional to $\sqrt{\varepsilon_t}$, with according bounding values $[R_{min}^{(i)}; R_{max}^{(i)}]$:

$$R_t^{(i)} = \max(\min(\sqrt{\varepsilon_t}, R_{max}^{(i)}), R_{min}^{(i)})$$

When $R_t^{(i)}$ are large, the predicted distribution has a high variance and requires therefore a large number of particles to approximate it. In other words, the larger is the area of the state space subregion covered by the predicted distribution, the more particles are needed to explore it. This suggests having an adaptive number N_t of particles, using the formula:

$$N_t = N_0 \frac{1}{8} \sum_{i=1}^{8} R_t^{(i)}$$

4. Experimental results

The proposed method was implemented in non-optimized C++ and tested on a PC running WinXP at 2.4 GHz with 512 Mb of RAM.

Figure 3: AAM-based adaptive CONDENSATION tracking, for frames 69, 109 and 188. On each image, the drawn shape shows the estimated state $\hat{\mathbf{x}}_t$; the model and image texture $\mathbf{g}_{model}(\mathbf{c}_t)$ and $\mathbf{g}_{im}(\mathbf{p}_t, \mathbf{c}_t)$ are displayed in the lower right corner

Results are first shown for a video sequence where a face in near-frontal view undergoes large variations in pose, expressions and lighting (see Figure 3). The tracking of both global pose and facial features appears satisfying. Setting $N_0 = 500$, the number of particles N_t evolves between about 20 and 80, and increases each time the change in pose and/or appearance is rapid; using such adaptive dynamics allows to process on average 2 frames per second. This represents a drastic improvement over a method using a zero-velocity state evolution model, which required 1000 particles to successfully track this sequence (according to experiments not shown here).

The performance of our approach was also tested in presence of occlusions. We compared it with a purely deterministic AAM tracking, where the optimal state configuration is obtained by $\hat{\mathbf{x}}_t = \tilde{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \mathbf{v}_t$ (using the notations defined

Figure 4: Tracking on a video sequence with occlusions, frames 921, 928 and 940. Top row: deterministic AAM tracking. Bottom row: CONDENSATION based tracking.

in paragraph 3.3). As is highlighted in the top row of Figure 4, when the occlusion occurs, the deterministic search appears to be trapped in an incorrect local optimum, and the tracking diverges thus from that moment. This problem is overcome by the stochastic tracking: the occlusion induces a high texture error ε_t for the predicted state $\tilde{\mathbf{x}}_t$, and consequently the variance of drawn particles and their number N_t are increased (see the peaks in Figure 5).

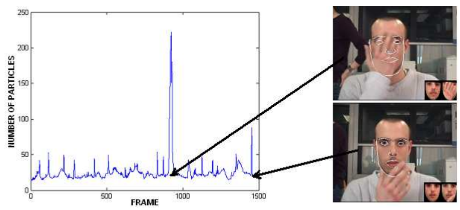

Figure 5: Evolving number of particles N_t on the video sequence with occlusions. The nearly full occlusion of frame 921 induces a high peak, while a partial occlusion occurring around frame 1400 induces a lower peak.

The particles cover thus a greater area of the state space, hopefully including the correct solution $\mathbf{x}_t^*$; as those particles are evaluated by the robust distance likelihood (4) and (5), the retained solution $\hat{\mathbf{x}}_t$ stands better chances to be a good candidate, which allows to correct the deterministic search (bottom row of Figure 4).

The effectiveness of the appearance model, largely proved in literature, remains conditioned by the fact that the tracked appearance must be beforehand learned and modelled. This modelling is sensitive to the recording conditions of the training images. In order to cure this problem, our current works consist to replace the AAM by an adaptive appearance estimated *on the fly* (on-line).

In this case, the general tracking algorithm principle remains the same in the sense that we use the same adaptive dynamics described on paragraph (3.3). However, the likelihood function, given by the equation (4), is now based on the distance between the image texture sampled at the hypothesized state and the model texture estimated and updated on the fly. The new texture model $g_{fly}(t)$ is initialized manually using the face texture in the first image of the video sequence and updated at each time step t using the following equation:

$$g_{fly}(t) = (1-\alpha) g_{fly}(t-1) + \alpha g_{im}(t, \hat{\mathbf{x}}_{t-1})$$

where α is a forgetting factor which determines the importance of the texture model update. $g_{im}(t, \hat{\mathbf{x}}_{t-1})$ is the current image texture estimated according to the state hypothesis selected at $t-1$, $\hat{\mathbf{x}}_{t-1}$. In this case, the hidden state space encodes the pose $\mathbf{p}_t$ and the first four modes of the shape parameters $\mathbf{b}_s$ obtained from the face model:

$$\mathbf{x}_t = (\mathbf{p}_t, \mathbf{b}_s)^T$$

This new model is robust vis-a-vis the lighting variations and occlusion schemes. The tracking problem is adapted to each target face without being conditioned by a preliminary training of its appearance. Some results are presented in (Figure 6).

Figure 6: Tracking pose and facial actions when replacing the texture model previously obtained from the AAM by an adaptive texture updated on the fly. Frames 75, 470 and 1310. $\alpha = 0.2$.

5. Conclusion

For the purpose of tracking the 2D global pose of a face and its inner facial actions, this paper proposes to combine an adaptive particle filtering scheme with an active appearance model. The state vector is composed of four pose parameters and four combined appearance parameters. The likelihood measures the fit between the hypothesized model texture and the image texture sampled at the hypothesized location and shape; a robust distance accounts for occluded pixels. Following the ideas of [10], the dynamics in state space are guided by a deterministic AAM search; this allows to reduce significantly the number of particles, which is only increased when the AAM search fails to converge to a satisfying solution. The experiments show that the proposed algorithm can successfully track a face and its facial actions undergoing quick motion and nearly full occlusions.

We also proposed to replace the texture model given by the AAM by an adaptive texture estimated on the fly to account for a necessary beforehand learning of the tracked appearance. Now that a robust tracking system is available, we can study the recognition of facial actions: the input being given by the combined appearance parameters at each time step, different recognition approaches can be tested, from a simple linear discriminant analysis on still frames, to dynamic graphical models. In this regard, the particle filter paradigm provides a natural inference framework for richer models — for instance, the facial action to be recognized could be included as a discrete component of the state vector.

6. References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. Journal of Computer Vision*, 56(3):221–255, February 2004.

[2] F. Bettinger, T.F. Cootes, and C.J. Taylor. Modelling facial behaviours. In *Proc. BMVC 2002*, volume 2, pages 797–806, 2002.

[3] M. Black and A. Jepson. Eigen-tracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. Journal of Computer Vision*, 36(2):101–130, 1998.

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

[5] A. Doucet, J. F. G. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[6] P. J. Huber. *Robust statistics*. Wiley, 1981.

[7] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, 1998.

[8] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proc. Europ. Conf. Computer Vision*, pages 661–675, 2002.

[9] D. Ross, J. Lim, and M.-H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proceedings of the European Conference on Computer Vision*, 2004.

[10] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. on Image Processing*, To appear, 2004.