Head and Facial Action Tracking: Comparison of two Robust Approaches

Romain Hérault, Franck Davoine, Yves Grandvalet

Head and Facial Action Tracking: Comparison of two robust approaches

HERAULT Romain DAVOINE Franck GRANDVALET Yves
HEUDIASYC, UMR CNRS 6599 / UTC, France, romain.herault@hds.utc.fr

Abstract

In this work, we address a method that is able to track simultaneously 3D head movements and facial actions like lip and eyebrow movements in a video sequence. In a baseline framework, an adaptive appearance model is estimated online by the knowledge of a monocular video sequence. This method uses a 3D model of the face and a facial adaptive texture model. Then, we consider and compare two improved models in order to increase robustness to occlusions. First, we use robust statistics in order to downweight the hidden regions or outlier pixels. In a second approach, mixture models provides better integration of occlusions. Experiments demonstrate the benefit of the two robust models. The latter are compared under various occlusions.

Keywords

3D Head tracking, facial action tracking, 3D model, adaptive appearance model, occlusion, robust statistics, mixture models.

1. Introduction

Head and facial action tracking pose 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 an especially difficult case of non-rigid object tracking.

In [8], the authors propose a tracking-by-detection method, using a key point matching procedure, to recover the 3D pose of a rigid human face under illumination changes and partial occlusions. These methods are fast but are not proposed to track non-rigid facial movements. Local parametric models [2] or optical flow [9] are more adapted to tracking non-rigid motion. Flexible shape and appearance models, later developed as Active Appearance Models (AAM) [3, 7], have been proposed as powerful tools for face analysis. AAM can be used to track simultaneously the shape, the head pose (often in a 2D space) as well as non-rigid facial gestures. As these models are learned on a face database, tracking is not accurate for new faces, when capture conditions have changed, or when a part of the head is occulted. In [10], robust statistics are introduced into the appearance model to improve robustness to occlusion. In [6], an appearance model based on mixture models is proposed to track natural objects with occlusion management. It uses an on-line estimation of the model by a recursive EM algorithm. These two methods have been applied in 2D spaces.

In this paper, we develop a baseline framework to 3D head pose and facial action tracking based on an adaptive appearance model [4] with an iterative registration. This registration is written in a new matricial expression to take into account occlusion managements: we extend robust statistics [5, 10] and mixture models [6] to simultaneous head and facial action tracking. We then present a comparison of each method on tracking disturbed by occlusions.

2. Baseline Framework

In this section, we present the baseline framework of a head and facial action tracking method [4], using a new matricial expression of the registration technique, in order to take into account occlusion management.

2.1. Face appearance model

The face appearance model consists of two components: the shape model, made up of a parametric 3D model, and a face texture model. From these two components, a face appearance can be generated.

2.1.1. Shape model. In our study, we use the generic 3D model *Candide* [1]. The matrix $\mathbf{g}$, aggregation of the 3D coordinates of all the 200 *Candide* vertices, represents the structure of the model. This matrix is obtained by modification of the matrix $\bar{\mathbf{g}}$ which stands for a reference face without expression:

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{S}\tau_S + \mathbf{A}\tau_A \quad (1)$$

In this decomposition, $\mathbf{S}$ model the inter-person variations and $\mathbf{A}$ model the intra-person variation. The vector τ_S is manually initialized to adapt the shape model to the subject physiognomy. These shape parameters τ_S are constant for a given person.

The matrix $\mathbf{A}$ is made up of action units: an action unit, according to the corresponding action parameter in τ_A, represents local facial movements. We track 6 action units: Jaw drop, Upper lip raiser, Lip stretcher, Lip corner depressor, Brow lowerer, Outer brow raiser. The 6-dimension action parameter vector τ_A is initialized to zero, and updated, frame by frame, by the tracker. The state of the shape model is composed of the action unit vector τ_A, three rotations $[\theta_x\, \theta_y\, \theta_z]$, two translations $[t_x\, t_y]$ and a global scale s which are summed up into a 12 dimension vector: $\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, s, \tau_A^T]^T$

2.1.2. Texture model. In our study, the texture is a 40 by 40 pixel image of the face appearance with the reference shape $\bar{\mathbf{g}}$. The texture model is a statistical model of the face texture. It consists in a multidimensional normal law with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}^2$. For computanional efficiency, the pixels are considered independent, thus $\boldsymbol{\Sigma}^2$ is a diagonal matrix. The parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ evolve during the process according to the new observations: the model is adaptive. At each new image $\mathbf{Y}$, we extract the face appearance and warp it from the current shape $\mathbf{g}$ to the reference shape $\bar{\mathbf{g}}$. This is done by affine transformations on each triangle of the shape model and results in a warped or shape-free texture $\mathbf{X}$.

We want to estimate model parameters ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$). The warped texture $\mathbf{X}$ is not stationary during the process. For example, changes in the lightning may occur. We assume that it is stationary inside a time window. In order to save memory, we can not store all the observations of the time windows, that is why we will consider observations under an exponential envelope, as in [10]. By this means, new observations can be recursively incorporated into the model. The exponential envelope is defined by:

$$S_t(k) = \alpha e^{\frac{k-t}{\tau}} \quad (2)$$

with $k < t$, $\tau = n_d / \log 2$ where n_d is the half-life of the envelope (in frame count) and $\alpha = 1 - e^{-1/\tau}$. The sum of $S_t(k)$ from $-\infty$ to 1 is equal to 1.

The first and the second order moments of the model are computed recursively according to the envelope:

$$\hat{\mathbf{M}}_{j,t} = \alpha \mathbf{X}_t^j + (1-\alpha)\hat{\mathbf{M}}_{j,t-1} \quad (3)$$

where j is the moment order and $\mathbf{X}^j$ denotes the component-wise exponentiation of $\mathbf{X}$ to the power j. In this notation, we have: $\boldsymbol{\mu}_t = \mathbf{M}_{1,t} \quad \boldsymbol{\sigma}_t^2 = \mathbf{M}_{2,t} - \boldsymbol{\mu}_t^2$ where $\boldsymbol{\sigma}^2$ stands for the diagonal elements of $\boldsymbol{\Sigma}^2$. Thus, We deduce the update equation of the texture parameters:

$$\hat{\boldsymbol{\mu}}_t = \alpha\, \mathbf{X}(\mathbf{b}_t) + (1-\alpha)\, \hat{\boldsymbol{\mu}}_{t-1} \quad (4)$$

$$\hat{\boldsymbol{\sigma}}_t^2 = \frac{\alpha}{1-\alpha}\, (\mathbf{X}(\mathbf{b}_t) - \hat{\boldsymbol{\mu}}_t)^2 + (1-\alpha)\, \hat{\boldsymbol{\sigma}}_{t-1}^2 \quad (5)$$

The matrix $\boldsymbol{\mu}$ is initialized at $t = 0$ by the warped texture $\mathbf{X}(\mathbf{b}_0)$ of the first frame. The diagonal matrix $\hat{\boldsymbol{\Sigma}}^2$ is initialized to a fixed value (here 5% of the interval) and updated at each new frame. In order to prevent any overflow computation, we put a low threshold on σ. We have chosen a fixed value of α into $[0.01, 0.10]$.

2.2. Tracking process

Tracking aims to estimate the state vector $\mathbf{b}_t$ of the shape model from the frame sequence, $\mathbf{Y}_{[t]} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_t\}$. We first approximate the probability of the frame sequence, $\mathbf{Y}_{[t]}$ parameterized by $\mathbf{b}$, by the probability of the last frame, $\mathbf{Y}_t$ paremeterized by $\mathbf{b}$: we assume that the current appearance does not rely on the past appearance. The face appearance does not have the same dimension in the whole process: as the shape is moving, the number of pixels covering the face appearance on the video is varying . Thus, in a second approximation, we will use the probability of the warped texture $\mathbf{X}_t$ and not the probability of $\mathbf{Y}_t$. The likelihood function of the appearance is given by:

$$p(\mathbf{Y}_{[t]}; \mathbf{b}_t) \simeq p(\mathbf{Y}_t; \mathbf{b}) \simeq p(\mathbf{X}_t) = \prod_{i=1}^{d} \mathbf{N}(x_i; \mu_i, \sigma_i) \quad (6)$$

with x_i,μ_i,σ_i respectively the i^{th} element of $\mathbf{X}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$; d is the dimension of the texture and $\mathbf{N}(x_i; \mu_i, \sigma_i)$ a Gaussian distribution:

$$\mathbf{N}(x_i; \mu_i, \sigma_i) = (2\pi\sigma_i^2)^{-1/2} \exp\left[-\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right] \quad (7)$$

The log-likelihood is the *Mahalanobis* distance between the warped texture and the expected texture:

$$\varepsilon(\mathbf{b}_t) = \sum_{i=1}^{d} \phi(u_{i,t}) \quad (8)$$

where $u_{i,t} = \frac{x_i(\mathbf{b}_t)-\mu_i}{\sigma_i}$ and $\phi(u) = \frac{1}{2}u^2$ in order to match the Gaussian density. We will modify this function in section 3.1 to improve robustness.

The dynamic model is given by:

$$\mathbf{b}_t = \mathbf{b}_{t-1} + \Delta\mathbf{b}_t \quad (9)$$

where $\Delta\mathbf{b}_t$ encodes the displacement on the shape model parameters.

The estimation of (9) is based on a matching technique between the texture model and the observed texture. This is done by minimizing the error $\varepsilon(\mathbf{b}_t)$. We look for $\Delta\mathbf{b}_t$ such that the gradient G of $\varepsilon(\mathbf{b}_t)$ with respect to $\mathbf{b}$ is null.

$$G(\mathbf{b}_t + \Delta\mathbf{b}_t) = G(\mathbf{b}_t) + \frac{\partial G}{\partial \mathbf{b}}\Delta\mathbf{b}_t = 0 \quad (10)$$

The approximation of the optimal $\Delta\mathbf{b}$ is then:

$$\Delta\mathbf{b} = -\left(\frac{\partial G}{\partial \mathbf{b}}\right)^{-1} G \quad (11)$$

This update is computed using the robust statistic estimation of Huber [5]:

$$\begin{aligned} G &= \sum_i \phi'(u_{i,t-1}) \frac{\partial u_i}{\partial \mathbf{b}} \\ G &= \sum_i \frac{\phi'(u_{i,t-1})}{u_{i,t-1}} u_{i,t-1} \frac{\partial u_i}{\partial \mathbf{b}} \\ G &= \sum_i v_i \; u_{i,t-1} \; \frac{1}{\sigma_i} \; \frac{\partial x_i(\mathbf{b}_{t-1})}{\partial \mathbf{b}} \end{aligned} \quad (12)$$

with $v_i = \frac{\phi'(u_{i,t-1})}{u_{i,t-1}}$. The algorithm W of [5] considers v_i as a constant; this gives:

$$\begin{aligned} \frac{\partial G}{\partial \mathbf{b}} &= \sum_i \frac{v_i}{\sigma_i^2} \frac{\partial x_i(\mathbf{b}_{t-1})}{\partial \mathbf{b}} \frac{\partial x_i(\mathbf{b}_{t-1})}{\partial \mathbf{b}^\intercal} \\ &+ \sum_i \frac{v_i}{\sigma_i} \phi'(u_{i,t-1}) \frac{\partial^2 x_i(\mathbf{b}_{t-1})}{\partial \mathbf{b} \partial \mathbf{b}^\intercal} \end{aligned} \quad (13)$$

Equations (12) and (13) can be expressed in matrix notation:

$$G = \mathbf{J}^\intercal \mathbf{V} \boldsymbol{\Sigma}^{-2} (\mathbf{X} - \boldsymbol{\mu}) \quad (14)$$

$$\frac{\partial G}{\partial \mathbf{b}} = \mathbf{J}^\intercal \boldsymbol{\Sigma}^{-2} \mathbf{V} \mathbf{J} + \sum_i r_i \quad (15)$$

with $\mathbf{J}$ the Jacobian matrix $\mathbf{J}_{i,k} = \frac{\partial x_i(\mathbf{b})}{\partial b_k}$ where b_k is the k^{th} element of $\mathbf{b}$, $r_i = \frac{1}{\sigma_i}\phi'(u_{i,t-1})\frac{\partial^2 x_i(\mathbf{b}_{t-1})}{\partial \mathbf{b}\partial \mathbf{b}^\intercal}$, and $\mathbf{V}$ diagonal matrix such as: $\mathbf{V}_{i,i} = \frac{\phi'(u_{i,t-1})}{u_{i,t-1}}$.

We can not estimate $\frac{\partial^2 x_i(\mathbf{b}_{t-1})}{\partial \mathbf{b}\partial \mathbf{b}^\intercal}$ so we can not estimate r_i. We assume that the last term of equation (15) can be neglected. $\sum_i r_i$ can be considered small compared to the other terms when $\mathbf{b}_{t-1}$ is near $\mathbf{b}_t$. This hypothesis is used in the Gauss-Newton method. This gives :

$$\frac{\partial G}{\partial \mathbf{b}} = \mathbf{J}^\intercal \boldsymbol{\Sigma}^{-2} \mathbf{V} \mathbf{J} \quad (16)$$

Thus, equation (11) can be written :

$$\Delta\mathbf{b}_t = -\left(\mathbf{J}^\intercal \boldsymbol{\Sigma}_t^{-2} \mathbf{V} \mathbf{J}\right)^{-1} \mathbf{J}^\intercal \mathbf{V} \boldsymbol{\Sigma}_t^{-2} (\mathbf{X}_t(\mathbf{b}_{t-1}) - \boldsymbol{\mu}_t) \quad (17)$$

with the *Mahalanobis* distance, $\phi(u) = \frac{1}{2}u^2$, we have $\phi'(u) = u$. Thus, the $\mathbf{V}$ matrix is the identity matrix. This fact leads to a simpler expression of $\Delta\mathbf{b}_t$:

$$\Delta\mathbf{b}_t = -\left(\mathbf{J}^\intercal \boldsymbol{\Sigma}_t^{-2} \mathbf{J}\right)^{-1} \mathbf{J}^\intercal \boldsymbol{\Sigma}_t^{-2} (\mathbf{X}_t(\mathbf{b}_{t-1}) - \boldsymbol{\mu}_t) \quad (18)$$

Due to the truncated approximation of $\varepsilon(\mathbf{b}_t)$, the minimum is not reached by solution to equation (18) but $\Delta\mathbf{b}_t$ provides a descent direction:

$$\mathbf{b}_t = \mathbf{b}_{t-1} + \rho\Delta\mathbf{b}_t \quad (19)$$

The step ρ is estimated recursively by golden search.

The value of the Jacobian $\mathbf{J}$, needed to compute $\Delta\mathbf{b}$, can not be accessed directly because no analytic expression of $\mathbf{X}(\mathbf{b})$ exists. This Jacobian is estimated by numerical differentiation smoothed by a uniform distribution. The j^{th} column of $\mathbf{J}$ can be expressed by:

$$\mathbf{J}_{j,t+1} = \frac{1}{K} \sum_{k=-K/2, k\neq 0}^{K/2} \frac{\mathbf{X}_t(\mathbf{b}_t) - \mathbf{X}_t(\mathbf{b}_t + k\,\delta_j\,\mathbf{e}_j)}{k\,\delta_j} \quad (20)$$

where δ_j is the smallest step of the $\mathbf{b}_j$ parameter, $\mathbf{e}_j$ is an null vector except on the j^{th} item and K the number of samples used in the uniform distribution.

3. Improving Robustness

3.1. Robust statistics

The impact of occlusions on tracking can be reduced thanks to the use of robust statistics [10]. An influence function is introduced in the likelihood formula [5]. When the value of warped texture pixel is too far from the expected texture (in our study, 3 times the standard deviation), the pixel is declared as an outlier. Its influence on the likelihood function is then linear instead of quadratic. In the *Mahanalobis* distance (equation 8) the function ϕ is replaced by a new function ψ where $c = 3$:

$$\psi(u) = \begin{cases} \frac{1}{2}u^2 & \text{if} \quad |u| \leq c \\ c|u| - \frac{1}{2}c^2 & \text{if} \quad |u| > c \end{cases} \quad (21)$$

This influence function affects $\Delta\mathbf{b}$: the $\mathbf{V}$ matrix is not identity matrix, we have to keep the whole definition of $\Delta\mathbf{b}$ (equation 17) with:

$$\mathbf{V}_{i,i} = \begin{cases} 1 & \text{if} \quad |u_{i,t-1}| \leq c \\ \frac{c}{|u_{i,t-1}|} & \text{if} \quad |u_{i,t-1}| > c \end{cases} \tag{22}$$

Occlusions must not influence the texture model: when a pixel is an outlier, its value does not affect the texture model, the update equations (4)-(5) are not applied. A global occlusion is declared when the percentage of outlier pixels exceeds some threshold.

3.2. Mixture model

Jepson et al. [6] have proposed the use of mixture models in the block matching framework. Our second method follows the same idea.

The observation $\mathbf{X}_t$ varies smoothly under modification of facial expression or out-of-plane rotation, while abrupt change may occur in case of occlusions. This observation suggests building a two component mixture distribution: one component models stable observations, one component models noise and unpredictive phenomena like occlusions. The stable component is modeled by a normal density: $p_s(\mathbf{X}_t;\mathbf{Q}_t)$. The model parameters, $\mathbf{Q}_t = (\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t^2)$, vary slowly during the process. This component corresponds to the texture model of the baseline framework. The noise component is modelled by a uniform density: $p_b(\mathbf{X}_t)$. These two components are combined in a mixture model. For the observation $\mathbf{X}_t$:

$$p(\mathbf{X}_t;\mathbf{m}_t,\mathbf{Q}_t) = \mathbf{m}_t p_s(\mathbf{X}_t;\mathbf{Q}_t) + (1-\mathbf{m}_t) p_b(\mathbf{X}_t) \tag{23}$$

where $\mathbf{m}_t$ is the mixing parameters vector.

For each new frame, we update the $\mathbf{m}_t$ and $\mathbf{Q}_t$ vectors of equation (23). This is done by an Expectation-Maximization algorithm:

The E-step computes the ownership of both components. The ownership to the stable component $o_{s,t}$ is computed by:

$$o_{s,t}(\mathbf{X}_t) = \frac{\mathbf{m}_t p_s(\mathbf{X}_t;\mathbf{Q}_t)}{p(\mathbf{X}_t;\mathbf{m}_t,\mathbf{Q}_t)} \tag{24}$$

The noise ownership being $o_{b,t}(\mathbf{X}_t) = 1 - o_{s,t}(\mathbf{X}_t)$. All operations are componentwise.

The M-step needs to access the whole set of observations. Jepson et al. [6] proposed an on-line estimation based on a recursive approximation of the EM Algorithm which allows forgetting the observations. The ownership at the current time of past observation are approximated by the ownership at the corresponding observation time: that is, for all $k < t$, $o_{s,k}(.)$ is approximated by $o_{s,t}(.)$. This leads to a recursive expression of the moments:

$$\hat{\mathbf{M}}_{j,t} = \alpha \mathbf{X}_t^j o_{s,t}(\mathbf{X}_t) + (1-\alpha)\hat{\mathbf{M}}_{j,t-1} \tag{25}$$

As $\mathbf{m}_{s,t}$ equals $\mathbf{M}_{0,t}$, we also have:

$$\hat{\mathbf{m}}_t = \alpha o_{s,t}(\mathbf{X}_t) + (1-\alpha)\hat{\mathbf{m}}_{t-1} \tag{26}$$

Thanks to ownerships, the EM algorithm reduces the influence of occlusions on the texture model update. Regarding the tracking: the influence of outlier pixels is reduced due to ownership weights on the pixel error to the stable component. The state update $\Delta\mathbf{b}$ is then obtained by :

$$\Delta\mathbf{b}_t = -\left(\mathbf{J}^\intercal \boldsymbol{\Sigma}_t^{-2} \mathbf{V} \mathbf{J}\right)^{-1} \mathbf{J}^\intercal \mathbf{V} \boldsymbol{\Sigma}_t^{-2} (\mathbf{X}_t(\mathbf{b}_{t-1}) - \boldsymbol{\mu}_t) \tag{27}$$

with $\mathbf{V}_{i,i} = [o_{s,t}(\mathbf{X}_t)]_i$ the ownership of the i^{th} pixel.

4. Experimental Results

4.1. Tracking process

The tracker is robust to large head or facial action movements. As it does not rely on a learned database, it can be applied immediately to an unknown subject; but the frame rate is slower than rates of trackers learned on a database like Active Appearance Models. Figure 4 shows the tracking of the 3D head pose and facial actions of a subject in a 591-frame sequence, each frame is 512 by 512 pixels; the frames 93, 204 and 441 are displayed. The curves in figure 1 show the variations of 2 rotations of the head pose as well as the variations of the all 6 action parameters involved in the tracking, the mixture models method is used.

4.2. Comparison of the robust models

We have produced occlusions with two kinds of objects: a hand, whose skin can easily be mistaken for face texture; a black rod, whose black plastic texture is dissimilar to the face texture.

4.2.1. Occlusion by a hand. When the hand hides the face, tracking is not disturbed. When the hand has stayed at length and moves, the shape model drifts to follow the hand until it leaves completely off the face. The model returns then to the correct position. Figure 5 shows the face appearance overdubbed by the shape model during a hand occlusion: the drift is noticeable with the baseline framework (a) reduced with robust statistics (b) and nearly inexistent with mixture models (c). This drift can be highlighted on action parameters: figure 2 shows the artefact induced by the

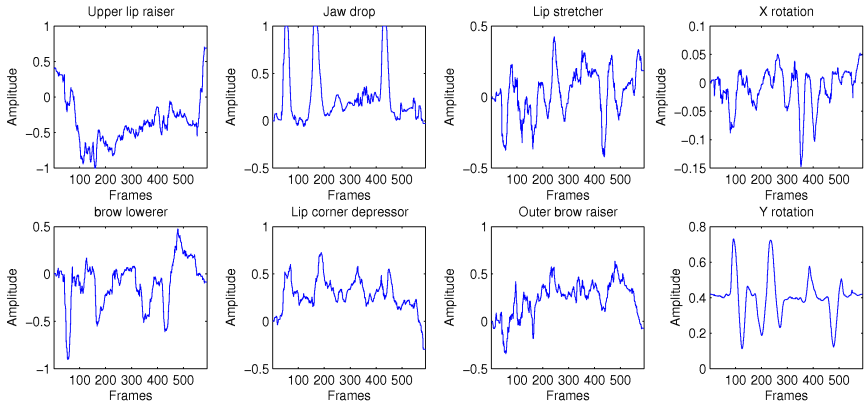


Figure 1. Tracking of the 6 action parameters and 2 head pose parameters

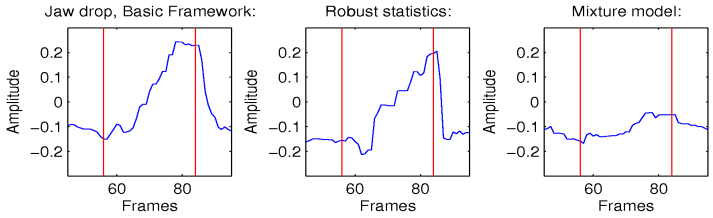


Figure 2. Artefacts induced by an occlusion of a similar texture object

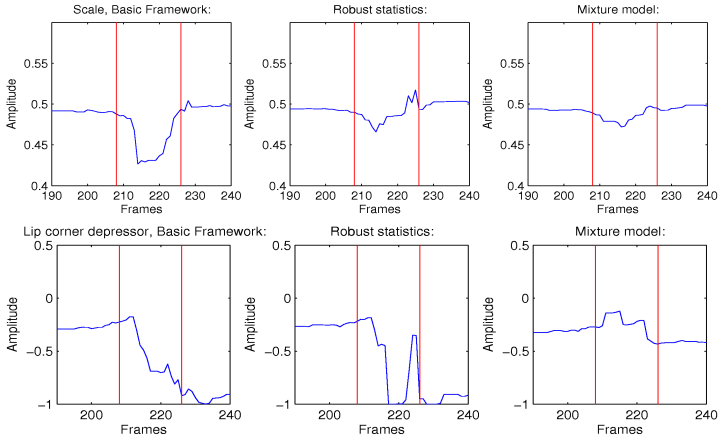


Figure 3. Artefacts induced by an occlusion of a dissimilar texture object

occlusion on the *Jaw drop* parameter for each method. Central bars stand for the beginning and the end of the occlusion.

The tracking process is not disturbed by the hand because its texture is similar to the face texture. But when the hand has stayed at length on the face, it is incorporated into the model texture. Then when the hand moves, the tracker mistakes this movement as a face movement. Both robust methods prevent this as they reject the hand when updating the texture model.

4.2.2. Occlusion by a black rod. When the rod hides the face, tracking is disturbed (figure 6): there is a drift in the head pose and facial actions are abnormal within the baseline framework (a), the drift on the head pose is less important with robust statistics (b) and only a little artefact on mouth facial actions can be seen with mixture models (c). The rod is correctly detected as occlusion by both robust methods (figure 7): the rod pixels on the warped texture (a) are correctly declared as outlier with the first method (b) or rejected on the noise component with the second method (c). The induced artefacts can lead to a target loss: the parameters keep being in a wrong position after the occlusion stops. Figure 3 shows state parameters during the rod occlusion:

Scale (a) and *Lip corner depressor* (b). For (a), the artefact is reduced by both robust methods. For (b), a target loss occurs with the baseline framework and robust statistics.

The tracking process is disturbed by the rod as this occlusion lowers significantly the likelihood of the warped texture. The robust statistics do not reject sufficiently outliers, leading to artefacts on facial parameters.

5. Conclusion and future works

In this work, we track simultaneously head and facial actions like lip and eyebrow movements in a video sequence. Our baseline framework is based on an adaptive appearance model.

We have considered and compared two improved appearance models aiming to increase robustness to occlusions. The first method, robust statistics, is efficient when the face is occluded by an object with similar texture. Confronted with an dissimilar object in its texture, it does not prevent target loss. The second approach, mixture models, performs better in both cases: Temporary occlusion of the face leads to smaller drifts and the tracked object is recovered in more extreme situations.

Our ultimate goal is to perform classification in high-level categories, such as expression recognition. In the adopted appearance model, the shape parameters τ_S and the action parameters τ_A are fitted independently. As a result, tracking produces identity-independent action units, which may be processed by the classifier that will be designed for the high-level recognition task. Tuning such a classifier usually requires an extensive database. We expect that the modeling efforts at the tracking stage will considerably reduce this burden, opening the possibility to extrapolate the classification rule to previously unseen subjects.

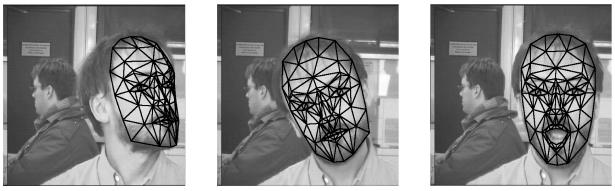
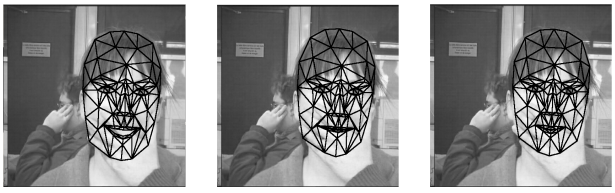
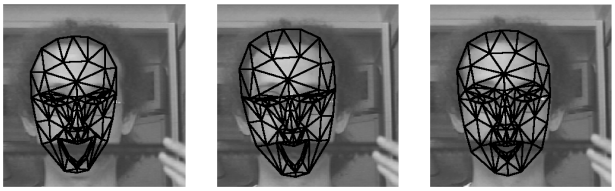


Figure 4. Sample images of tracking



(a) Basic framework (b) Robust statistics (c) Mixture model

Figure 5. Occlusion with a similar texture objet



(a) Basic framework (b) Robust statistics (c) Mixture model

Figure 6. Occlusion with a dissimilar texture objet

(a) Warped texture (b) Outliers (c) Stable ownership

Figure 7. Warped texture with the rod occlusion

References

[1] J. Ahlberg. CANDIDE-3 - an updated parametrized face. Technical Report LiTH-ISY-R-2326, Linköping University, Sweden, 2001.

[2] M. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parametrized models of image motion. *International Journal of Computer Vision*, 25(1):23–48, 1997.

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

[4] F. Davoine and F. Dornaika. Head and facial animation tracking using appearance-adaptive models and particle filters. In *Real-Time Vision for Human-Computer Interaction*, pages 121–140. Springer Verlag, 2005.

[5] P. Huber. *Robust Statistics*. Wiley, 2003.

[6] 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.

[7] I. Matthews and S. Baker. Active appearance models revisited. Technical Report CMU-RI-TR-03-02, The Robotics Institute, Carnegie Mellon University, 2002.

[8] J. P. V. Lepetit and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *CVPR*, Washington D.C., June 2004.

[9] Y. Yacoob and L. Davis. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):636–642, 1996.

[10] S. Zhou, R. Chellappa, and B. Mogghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13:1491–1506, Nov. 2004.