J Real-Time Image Proc (2007) 2:35–44
DOI 10.1007/s11554-007-0032-2

ORIGINAL RESEARCH PAPER

# Real time 3D face and facial feature tracking

**Fadi Dornaika · Javier Orozco**

Received: 23 November 2006 / Accepted: 26 June 2007 / Published online: 28 July 2007


**Abstract** Detecting and tracking human faces in video sequences is useful in a number of applications such as gesture recognition and human-machine interaction. In this paper, we show that online appearance models (holistic approaches) can be used for simultaneously tracking the head, the lips, the eyebrows, and the eyelids in monocular video sequences. Unlike previous approaches to eyelid tracking, we show that the online appearance models can be used for this purpose. Neither color information nor intensity edges are used by our proposed approach. More precisely, we show how the classical appearance-based trackers can be upgraded in order to deal with fast eyelid movements. The proposed eyelid tracking is made robust by avoiding eye feature extraction. Experiments on real videos show the usefulness of the proposed tracking schemes as well as their enhancement to our previous approach.

**Keywords** Real-time tracking · 3D face tracking · Facial feature tracking · Eyelid tracking · Online appearance models

## 1 Introduction

Detecting and tracking human faces in video sequences is useful in a number of applications such as gesture recognition and human-machine interaction. Faces play a major role in any human–computer interaction system, because they represent a rich source of information. Faces are the main cue humans use for person detection/identification. Besides this, faces are the main gateway to express our feelings and emotional states. Being able to estimate 3D face pose in real-time, we can get a clue about user's intentions.

Vision-based tracker systems provide an attractive alternative since vision sensors are not invasive. Of particular interest are vision-based markerless head and/or face trackers. Since these trackers do not require any artificial markers to be placed on the face, comfortable and natural motions can be achieved. On the other hand, building robust and real-time markerless trackers for head and facial features is a difficult task due to the high variability of the face and the facial features in videos.

To overcome the problem of appearance changes recent works on faces adopted statistical facial textures. For example, the Active Appearance Models have been proposed as a powerful tool for analyzing facial images [5]. Deterministic and statistical appearance-based tracking methods have been proposed [2, 4, 12]. These methods can successfully tackle the image variability and drift problems by using deterministic or statistical models for the global appearance of a special object class: the face. A few algorithms exist which attempt to track both the head and the facial features in real time, e.g., [2, 12]. These works have addressed the combined head and facial feature tracking using the Active Appearance Model principles. However, [2, 12] require tedious learning stages that should be performed beforehand and should be repeated whenever the imaging conditions change. Recently, we have developed a head and facial feature tracking method based on online appearance models (OAMs) [6]. Unlike the active

F. Dornaika (✉)
Institut Géographique National Laboratoire MATIS,
2 Avenue Pasteur, 94165 Saint Mandé, France
e-mail: fadi.dornaika@ign.fr

J. Orozco
Computer Vision Center, Campus UAB,
08193 Bellaterra, Barcelona, Spain
e-mail: orozco@cvc.uab.es

appearance models, the OAMs offer a lot of flexibility and efficiency since they do not require any facial texture model that should be computed beforehand. Instead the texture model is built online from the tracked sequence.

This paper extends our previous work [6] in two directions. First, we show that by adopting a non-occluded shape-free facial texture that excludes the eye region, more accurate and stable 3D head pose parameters can be obtained. Second, unlike feature-based eyelid trackers, we show that the OAMs can be used to track the eyelids. Thus, we can infer the eye state without detecting the eye features such as the irises and the eye corners. A short version of this paper has appeared in [7]. In this paper, we additionally provide (i) extended experiments, (ii) a description about how to build an appearance-based tracker able to deal with fast eyelid movements, and (iii) a performance evaluation aiming at quantifying the eye blink detection in real conditions.

Tracking the eyelids and the irises can be used in many applications such as drowsiness detection and interfaces for handicapped individuals. For applications such as driver awareness systems, one needs to do more than tracking the locations of the person's eyes but obtain their detailed description. Detecting and tracking the eye and its features (eye corners, irises, and eyelids) have been addressed by many researchers [11, 14, 16, 19]. However, most of the proposed approaches rely on intensity edges and are time consuming. In [16], detecting the state of the eye is based on the iris detection in the sense that the iris detection results will directly decide the state of the eye. In [14], the eyelid state is inferred from the relative distance between the eyelid apex and the iris center. For each frame in the video, the eyelid contour is detected using edge pixels and normal flow. The authors reported that when the eyes were fully or partially open, the eyelids were successfully located and tracked 90% of the time. Their proposed approach depends heavily on the extracted intensity edges. Moreover, it assumes high-resolution images depicting an essentially frontal face. In our study, we do not use any edges and there is no assumption on the head pose. In our work, the eyelid motion is inferred at the same time with the 3D head pose and other facial actions, that is, the eyelid state does not rely on the detection results of other features such as the eye corners and irises. Tracking the rapid eyelid motion is not a straightforward task. In our case, we like to track the eyelid motion using the principles of OAMs. The challenges are as follows. First, the upper eyelid is a highly deformable facial feature since it has a great freedom of motion. Second, the eyelid can completely occlude the iris and sclera, that is, a facial texture model will have two different appearances at the same locations. Third, the eyelid motion is very fast.

The remainder of this paper proceeds as follows. Section 2 introduces our deformable 3D facial model as well as the concept of shape-free facial patches. Section 3 describes the online adaptive appearance model. Section 4 presents a generic tracking algorithm that tracks in real-time the 3D head pose and some facial actions. It describes how the eyelids are tracked using this generic algorithm. Section 5 gives some comparisons obtained with different facial patches. In Sect. 6, we present some tracking results and one performance evaluation for eye-blinking detection.

## 2 Modeling faces

### 2.1 A deformable 3D model

In our study, we use the 3D face model *Candide* [3]. 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

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\boldsymbol{\tau}_a \tag{1}$$

where $\mathbf{g}_s$ is the static shape of the model, $\boldsymbol{\tau}_a$ the facial action control vector, and the columns of $\mathbf{A}$ are the animation units. In this study, we use seven modes for the facial animation units (AUs) matrix $\mathbf{A}$. We have chosen the seven following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer, outer eyebrow raiser and eyelid raiser. These AUs are enough to cover most common facial animations. Moreover, they are essential for conveying emotions. Thus, the lips are controlled by four facial actions, the eyebrows are controlled by two facial actions, and the eyelids by one facial action. Figure 1 illustrates these seven facial actions. In this study,

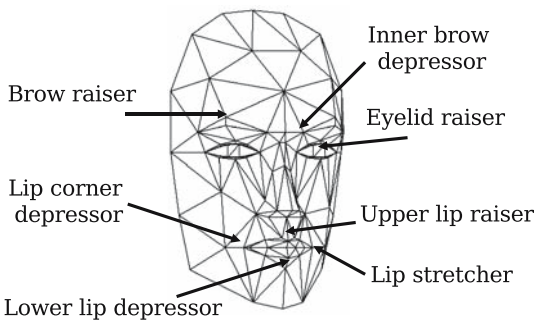


**Fig. 1** The *Candide* model and the seven facial actions

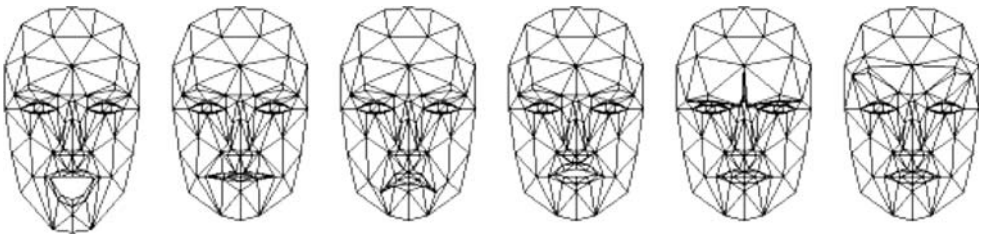

**Fig. 2** The configuration of the 3D face mesh (*frontal view*) when each facial action is set to its "maximum" value. From *left* to *right*: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer, outer eyebrow raiser

the eyelids pair, the inner eyebrows pair, and the outer eyebrows pair are each controlled by one facial action.

In Eq. 1, the 3D shape is expressed in a local coordinate system. However, one should relate the 3D coordinates to the image coordinate system. To this end, we adopt the weak perspective projection model. We neglect the perspective effects since the depth variation of the face can be considered as small compared to its absolute depth. Thus, the state of the 3D wireframe model is given by the 3D head pose parameters (three rotations and three translations) and the internal face animation control vector $\boldsymbol{\tau}_a$.

This is given by the 13-dimensional vector $\mathbf{b}$:

$$\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau}_a^T]^T \quad (2)$$

where:

- $\theta_x$, $\theta_y$, and $\theta_z$ represent the three angles associated with the 3D rotation between the 3D face model coordinate system and the camera coordinate system.
- $t_x$, $t_y$, and $t_z$ represent the three components of the 3D translation vector between the 3D face model coordinate system and the camera coordinate system.
- Each component of the vector $\boldsymbol{\tau}_a$ represents the intensity of one facial action. This belongs to the interval [0, 1] where the zero value corresponds to the neutral configuration (no deformation) and the one value corresponds to the maximum deformation. In the sequel, the word "facial action" will refer to the facial action intensity. Figure 2 illustrates the configuration of the 3D wireframe model when each facial action is set to its maximum value.

Note that if only the aspect ratio of the camera is known, then the component $t_z$ (the in-depth translation) is replaced by a scale factor $s$ having the same mapping role between 3D and 2D. This scale is given by $s = \frac{\alpha}{t_z}$ where $\alpha$ is the focal length of the camera in pixels.

## 2.2 Shape-free facial patches

A facial image is represented as a shape-free texture, as shown in Fig. 3b. In this section, we briefly describe how this shape-free texture is computed from the input image and the geometrical parameters $\mathbf{b}$. More details can be found in [1].

The 2D mesh associated with the shape-free texture is obtained by projecting the static shape $\mathbf{g}_s$ (wireframe), using a centered frontal 3D pose, onto an image with a given resolution. The texture of the shape-free facial image is obtained by texture mapping from the triangular 2D mesh covering the face in the input image (see Fig. 3a) using a piece-wise affine transform, $\mathcal{W}$. Similarly to [1], we have taken advantage of the fact that the barycentric coordinates of the pixels within each triangle are invariant under affine transforms. In other words, since the geometry of the 2D mesh in the shape-free image is fixed, the barycentric coordinates are fixed and can be computed once for all, which considerably reduces the CPU time associated with the texture mapping process—the warping process.

Once an instance of the 3D model (encoded by the vector $\mathbf{b}$) is projected onto the input image, the warping process proceeds as follows. The shape-free image bounding the fixed 2D mesh is scanned pixel by pixel. For every scanned pixel in this image, we know its triangle as well as its barycentric coordinates within this triangle. Therefore, the 2D location of the corresponding pixel in the input image can be easily inferred using a linear combination of the coordinates of the triangle vertices where the

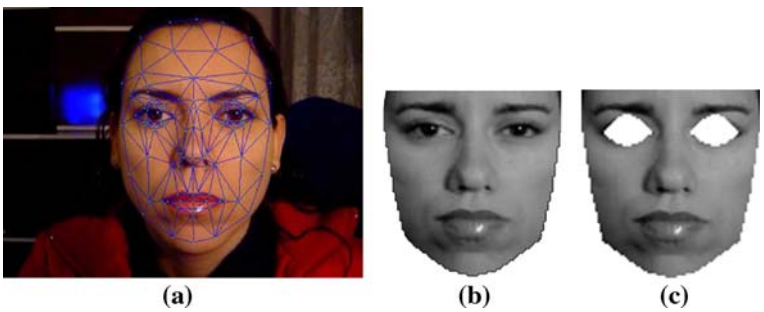


**Fig. 3** **a** An input image with correct adaptation. **b** The corresponding shape-free facial patch. **c** The same patch without the eyes region

coefficients are given by the barycentric coordinates. The greylevel of the scanned pixel is then set by blending the greylevels associated with the four closest pixels to the non-integer coordinates of the returned location—the bilinear interpolation.

Mathematically, the warping process applied to an input image $\mathbf{y}$ is denoted by

$$\mathbf{x}(\mathbf{b}) = \mathcal{W}(\mathbf{y}, \mathbf{b}) \tag{3}$$

where $\mathbf{x}$ denotes the shape-free facial texture and $\mathbf{b}$ denotes the geometrical parameters. Without loss of generality, we have used two resolution levels for the shape-free textures, encoded by 1,310 and 5,392 facial pixels bounded by $40 \times 42$ and $80 \times 84$ rectangular boxes, respectively. Obviously, other levels can be used. Generally speaking as the resolution increases, the tracking accuracy increases. However, by experience we found that the second resolution level is a good trade-off between the accuracy and the computational cost.

To partially compensate for contrast variations, the original shape-free texture $\mathbf{x}$ is transformed into an image having a mean equal to 0 and a variance equal to 1. The complete image transformation is implemented as follows: (i) transfer the texture $\mathbf{y}$ using the piece-wise affine transform associated with the geometric parameters $\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau}_a^T]^T$, and (ii) perform zero-mean-unit-variance normalization on the obtained patch.

Figure 3 illustrates two shape-free patches associated with an input image.

## 3 Problem formulation and adaptive appearance models

Given a video sequence depicting a moving head/face, we would like to recover, for each frame, the 3D head pose and the facial actions encoded by the control vector $\boldsymbol{\tau}_a$. In other words, we would like to estimate the vector $\mathbf{b}_t$ (2) at time $t$ 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.

For each input frame $\mathbf{y}_t$, the observation is the shape-free facial patch associated with the geometric parameters $\mathbf{b}_t$. We use the HAT symbol for the tracked parameters and patches. 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,

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

The estimation of the current parameters $\hat{\mathbf{b}}_t$ from the previous ones $\hat{\mathbf{b}}_{t-1}$ and from the sequence of images will be presented in Sect. 4. In our work, the initial parameters $\hat{\mathbf{b}}_1$ corresponding to the first frame are manually provided. The automatic initialization can be obtained using the statistical technique proposed in [2].

By assuming that the pixels within the shape-free patch are independent, we can model the appearance of the shape-free facial patch using a multivariate Gaussian with a diagonal covariance matrix $\boldsymbol{\Sigma}$. Let $\boldsymbol{\mu}$ be the Gaussian center and $\boldsymbol{\sigma}$ be the vector containing the square root of the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are $d$-vectors ($d$ is the size of $\mathbf{x}$) representing the appearance parameters. In summary, the observation likelihood at time $t$ is written as

$$p(\mathbf{y}_t|\mathbf{b}_t) = p(\mathbf{x}_t|\mathbf{b}_t) = \prod_{i=1}^{d} \mathbf{N}(x_i; \mu_i, \sigma_i)_t \tag{5}$$

where $\mathbf{N}(x_i; \mu_i, \sigma_i)$ is a normal density:

$$\mathbf{N}(x_i; \mu_i, \sigma_i) = (2\pi\sigma_i^2)^{-1/2} \exp\left[-\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right] \tag{6}$$

We assume that the appearance model summarizes the past observations under an exponential envelope, that is, the past observations are exponentially forgotten. When the appearance is tracked for the current input image, i.e. the texture $\hat{\mathbf{x}}_t$ is available, we can update the appearance and use it to track in the next frame. It can be shown that the appearance model parameters, i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ can be updated using the following equations (see [9] for more details on online appearance models):

$$\mu_{i_{(t+1)}} = (1 - \alpha)\mu_{i_{(t)}} + \alpha\hat{x}_{i_{(t)}} \tag{7}$$

$$\sigma^2_{i_{(t+1)}} = (1 - \alpha)\sigma^2_{i_{(t)}} + \alpha(\hat{x}_{i_{(t)}} - \mu_{i_{(t)}})^2 \tag{8}$$

In the above equations, the subscript $i$ denotes a pixel in the patch $\hat{\mathbf{x}}$. This technique, also called recursive filtering, is simple, time-efficient and therefore suitable for real-time applications. The appearance parameters reflect the most recent observations within a roughly $L = 1/\alpha$ window with exponential decay.

Note that $\boldsymbol{\mu}$ is initialized with the first patch $\hat{\mathbf{x}}_1$ corresponding to the geometrical parameters $\hat{\mathbf{b}}_1$. However, Eq. 8 is not used until the number of frames reaches a given value (e.g., the first 40 frames). For these frames, the classical variance is used, that is, Eq. 8 is used with $\alpha$ being set to $1/t$.

Here, we used a single Gaussian to model the appearance of each pixel in the shape-free patch. However, modeling the appearance with Gaussian mixtures can also be used on the expense of some additional computational load (e.g., see [10, 18]).

## 4 Tracking using adaptive appearance registration

### 4.1 A generic tracking algorithm

In this section, we describe the tracking algorithm that is used for tracking the head and the facial features in a monocular video sequence. The tracked facial actions as well as the shape-free facial patch can be arbitrary. We consider the state vector $\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau}_a^T]^T$ encapsulating the 3D head pose parameters and the facial actions. If we only track the lips and the eyebrows then the vector $\boldsymbol{\tau}_a$ will encode six facial actions, i.e., the vector $\mathbf{b}$ will have 12 components. If we track the lips, the eyebrows, and the eyelids then $\boldsymbol{\tau}_a$ will encode seven facial actions, i.e., the vector $\mathbf{b}$ will have 13 components. We will show how this state vector can be recovered for time $t$ using (i) the previous known state $\hat{\mathbf{b}}_{t-1}$, (ii) the current input image $\mathbf{y}_t$, and (iii) the current appearance parameters encoded by the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$.

The sought parameters $\mathbf{b}_t$ at time $t$ are estimated by registering the input image (warped version) to the current appearance model.

For this purpose, we minimize the *Mahalanobis* distance between the warped texture and the current appearance mean,

$$\min_{\mathbf{b}_t} e(\mathbf{b}_t) = \min_{\mathbf{b}_t} D(\mathbf{x}(\mathbf{b}_t), \boldsymbol{\mu}_t) = \sum_{i=1}^{d} \left( \frac{x_i - \mu_i}{\sigma_i} \right)^2 \tag{9}$$

The above criterion can be minimized using iterative first-order linear approximation which is equivalent to a Gauss-Newton method. It is worthwhile noting that minimizing the above criterion is equivalent to maximizing the likelihood measure given by (5). Moreover, the above optimization is made robust by using robust statistics [6]. For every frame in the video sequence, the corresponding state vector $\mathbf{b}_t$ is estimated using the following. The whole tracking algorithm is outlined in Fig. 4.

#### 4.1.1 Tracking algorithm

Starting from $\mathbf{b} = \hat{\mathbf{b}}_{t-1}$, we compute the error vector $\mathcal{W}(\mathbf{y}_t, \hat{\mathbf{b}}_{t-1}) - \boldsymbol{\mu}_t = \mathbf{x}_t(\hat{\mathbf{b}}_{t-1}) - \boldsymbol{\mu}_t$ and the corresponding *Mahalanobis* distance $e(\mathbf{b})$ (given by Eq. 9). We find a shift $\Delta\mathbf{b}$ by multiplying the error vector with the negative pseudo-inverse of the gradient matrix $\mathbf{G}_t = \frac{\partial \mathbf{x}_t}{\partial \mathbf{b}}$ using (10).

$$\Delta\mathbf{b} = -\mathbf{G}_t^{\dagger}(\mathcal{W}(\mathbf{y}_t, \hat{\mathbf{b}}_{t-1}) - \boldsymbol{\mu}_t) \tag{10}$$

where $\mathbf{G}_t^{\dagger} = (\mathbf{G}_t^T \mathbf{G}_t)^{-1} \mathbf{G}_t^T$ is the pseudo-inverse of $\mathbf{G}_t$.

The vector $\Delta\mathbf{b}$ gives a displacement in the search space for which the error, $e$, can be minimized. We compute a new parameter vector and a new error:

$$\mathbf{b}' = \mathbf{b} + \rho\Delta\mathbf{b} \tag{11}$$

$$e' = e(\mathbf{b}')$$

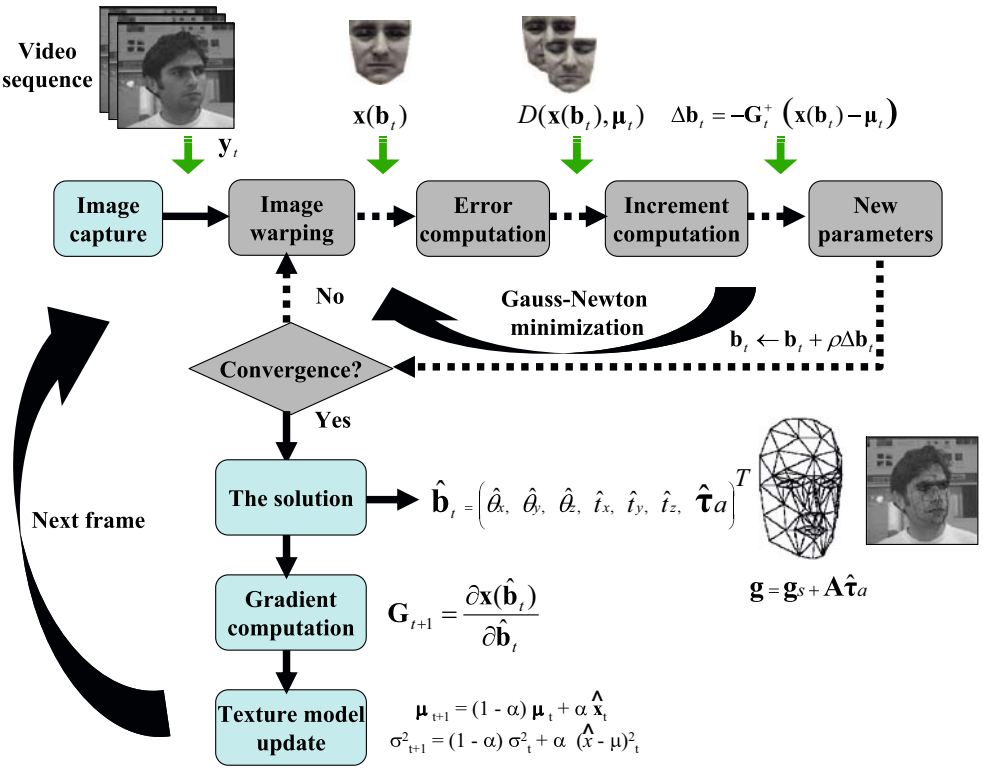


**Fig. 4** The generic appearance-based tracking algorithm. For every image in the monocular video sequence, the 3D head pose parameters as well as the facial actions are simultaneously estimated by registering the current texture with the current facial texture model—the current appearance model

where $\rho$ is a positive real.

If $e' < e$, we update $\mathbf{b}$ according to (11) and the process is iterated until convergence. If $e' \geq e$, we try smaller update steps in the same direction (i.e., a smaller $\rho$ is used). Convergence is declared when the error cannot be improved anymore or the number of iterations reaches a maximum.

In the above optimization, the gradient matrix $\mathbf{G} = \frac{\partial \mathcal{W}(\mathbf{y}_t, \mathbf{b}_t)}{\partial \mathbf{b}} = \frac{\partial \mathbf{x}_t}{\partial \mathbf{b}}$ is approximated by numerical differences similar to the work of Cootes [5]. Notice that the gradient matrix is computed for each time step. The advantage is twofold. First, a varying gradient matrix is able to accommodate appearance changes. Second, it will be closer to the exact gradient matrix since it is computed for the current geometric configuration (3D head pose and facial actions) whereas a fixed gradient matrix can be a source of errors. More details about this optimization technique can be found in [6].

### 4.2 Eyelids tracking

As we have mentioned earlier, tracking the eyelid motion is a challenging task, and most of the proposed approaches for locating and tracking the eyelids rely on the extracted intensity edges. In our case, the generic tracking algorithm (Sect. 4.1) is used for both cases: (i) tracking the lips and the eyebrows, and (ii) tracking the lips, the eyebrows, and the eyelids. However, in the second case, there are two main differences.

First, we adopt a shape-free facial patch whose eye region corresponds to closed eyes configuration (see Fig. 5), which excludes the iris and sclera regions. Note that the 2D shape of the eyelids in the shape-free patch is fixed like any other facial feature, that is, the eyelids appear closed in the facial patch regardless of the state of the eyelids in the input image. This is illustrated in Fig. 5. The maximum value of the eyelid facial action corresponds to wide open eyes while the zero value corresponds to closed eyes. Note that when the eyes are open in the input image, the texture of the eyelid region in the shape-free patch (associated with a correct eyelid facial action) will be a distorted version of a very small area in the input image. However, the global appearance of the eyelid is still preserved since the eyelids have the skin appearance.

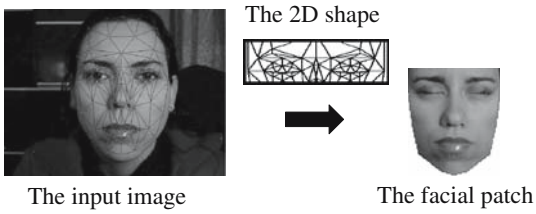


**Fig. 5** The shape-free facial patch used to track 13 degrees of freedom including the eyelid motion

Second, since the eyelid motion is very fast a good estimation of its gradient (a column in the global gradient matrix $\mathbf{G}_t = \partial \mathbf{x}_t / \partial \mathbf{b}$) is computed with a large number of perturbation steps that cover almost all the variation interval.

## 5 Tracking comparisons

In this section, we compare the 3D head pose estimates obtained with two different shape-free facial patches using the same tracking algorithm described above (Sect. 4.1) and the same state vector $\mathbf{b}$ given by the six head pose parameters and the six facial actions associated with the lips and eyebrows, that is, the eyelid facial action is not used.

To this end, we use the two shape-free patches depicted in Fig. 3. The first patch includes a region for the eye features namely the iris and the sclera. The second patch is obtained from the first one by only removing the eyes region.

We have used a 1,000-frame sequence featuring a talking subject[1] as a test video. Note that talking is a spontaneous activity. Figure 6 illustrates the estimated 3D head pose parameters associated with a 150-frame segment using the two different shape-free facial patches. The displayed parameters are (from top to bottom): the pitch angle $\theta_x$, the yaw angle $\theta_y$, the scale $s$, and the vertical translation $t_y$. The video segment starts at frame 500 and contains three eye blinks at frames 10, 104, and 145. The solid curves correspond to the first facial patch (with eye region) while the dotted curves correspond to the second patch (without eye region).

One can notice that (i) the solid curves and the dotted curves coincide for almost all frames, and (ii) the most significant discrepancies occur at those frames associated with an eye blink (e.g., see the scale plot). Although there is no ground-truth data for the head motions, we found that these discrepancies correspond well to some introduced errors since actually the head has not suddenly moved at these time instants (see frames 104 and 145). Thus, to some extent these discrepancies can be considered as errors associated with the estimated parameters.

Whenever an eye blink occurs the patch without the eyes region has provided more accurate and stable parameters than the patch with the eye region. This is explained by the fact that the patch containing the eye region (sclera and iris) does not model the shape of the eyelids. Therefore, despite the use of robust statistics the estimation

[1] http://www.prima.inrialpes.fr/FGnet/html/benchmarks.html.

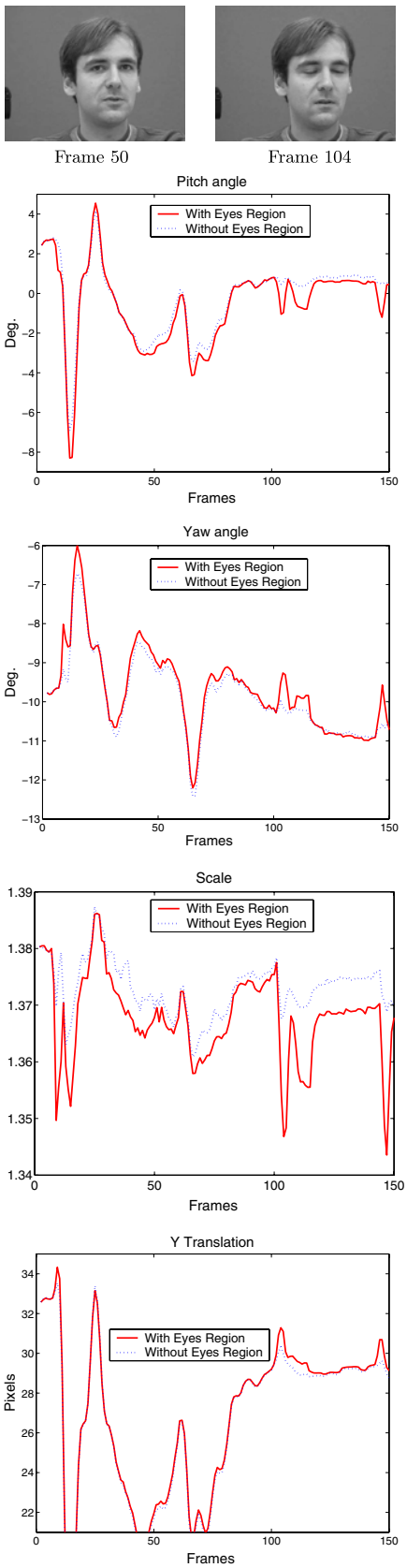


**Fig. 6** 3D head pose parameters obtained with two different facial patches that differ by the eyes region

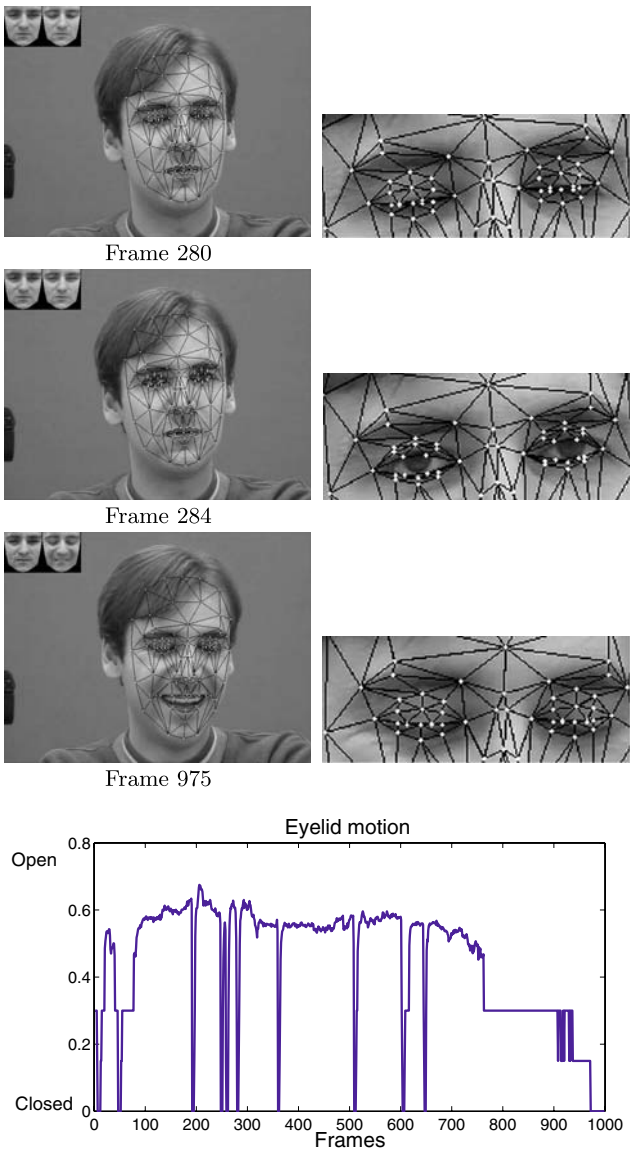


**Fig. 7** Tracking the 3D head pose, the lips, the eyebrows and the eyelids associated with a 1,000-frame sequence. Only frames 280, 284, 975 are shown. The plot depicts the estimated eyelid facial action as a function of the sequence frames

of the 3D head pose parameters with a patch containing the eye region (sclera and iris) is affected by the eyelid motion. One can notice that the rotational discrepancies seem to be small (about one degree). However, the vertical and in-depth translation errors can be large. For example, at frame 145 the obtained scale discrepancy is about 0.025, which corresponds to an in-depth error of about 3 cm.[2]

[2] The exact value depends on the camera-intrinsic parameters and the absolute depth.

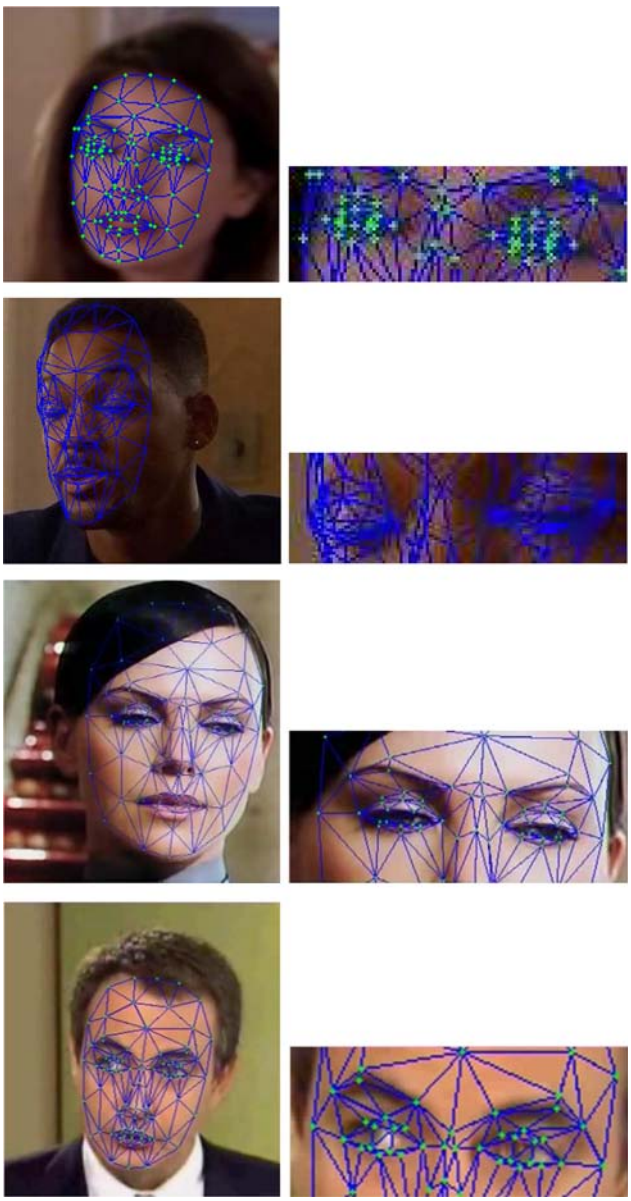

**Fig. 8** Tracking results associated with four video sequences. The first one has a low resolution

## 6 Head, lips, eyebrows, and eyelid tracking

In the previous section, we have shown that the accuracy of the tracked 3D head pose parameters can be affected by the eyelid motion (eye blinking) if the sclera and iris region is included in the texture model. This is not surprising since eye blinking corresponds to a sudden occlusion of a small part of the face. Thus, if the eyelid motion is tracked one can expect that the 3D head pose parameters can be more stable.

We have tracked the head, lips, eyebrows, and eyelids using the 1,000-frame sequence. Figure 7 displays the tracking results (13 degrees of freedom) associated with frames 280, 284, and 975. The left column displays


**Fig. 9** A frame belonging to the low resolution video depicted at the *top* of Fig. 8

zoomed views of those frames. Notice how the eyelids are correctly tracked in the input images. The estimated eyelid facial action reflects the degree of the eye openness.

The upper left corner of each image shows the current appearance ($\boldsymbol{\mu}_t$) and the current shape-free texture ($\hat{\mathbf{x}}_t$). The bottom of this figure displays the estimated eyelid facial action as a function of the sequence frames where zero value corresponds to a closed eye and one value to a wide open eye.

Figure 8 displays some tracking results obtained with four video sequences. The first video has a low resolution. Figure 9 displays a snapshot of the low resolution video.

### 6.1 Eye blink detection

Eye blinking is a discrete and important facial action [8, 13, 15]. The rate of blinking varies, but on average the eye blinks once every 5 s.[3] In our case, the eye-blinks can be directly detected and segmented by thresholding the tracked eyelid facial action. As can be seen, the dual state of the eye can easily be inferred from the continuous curve associated with the eyelid facial action. For the tracked sequences, all eye blinks are correctly detected and segmented.

Figure 10 illustrates the tracking results associated with a 594-frame sequence. This sequence depicts a subject performing head motions and facial animations. This sequence depicts 18 eye blinks. The proposed algorithm was able to correctly detect 17 blinks. The non-detected blink happened at the same time when the subject put on his glasses, that is, the eyelids are suddenly occluded by the frame of the eyeglasses, which corresponds to a sudden appearance variation. However, once the glasses are put on all subsequent eye blinks are correctly detected.

[3] http://www.science.enotes.com/science-fact-finder/human-body/how-often-does-human-eye-blink.

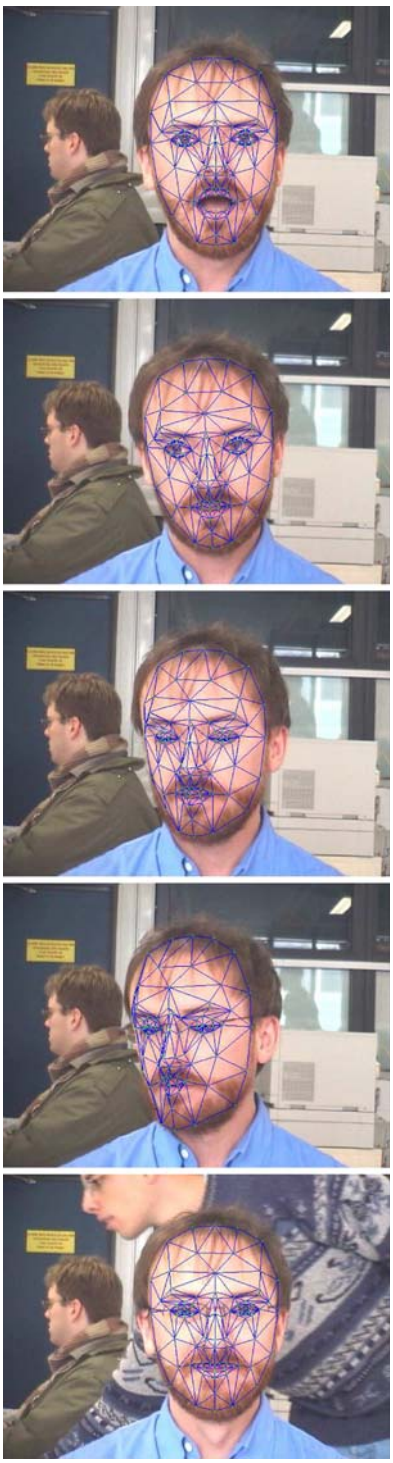

**Fig. 10** Tracking results associated with one video sequence depicting 18 eye blinks

### *6.1.1 Processing time*

On a 3.2-GHz PC, a non-optimized C code of our proposed approach computes the 3D head pose and the seven facial actions in 70 ms. The edge-based method presented in [16] takes 333 ms on a Pentium II 400 MHz PC. This method only tracks the iris location as well as the eye state. In [17], the average running time for tracking the eyelids and eye corners in one frame is about 100 ms. However, this method only tracks the eye corners and the eyelids.

## 7 Conclusion

In this paper, we have extended our appearance-based 3D head and facial action tracker to deal with eyelid motions. The 3D head pose and the facial actions associated with the lips, eyebrows, and eyelids are simultaneously estimated in real-time using OAMs. Compared with other eyelid tracking techniques our proposed approach has several advantages. First, computing and segmenting intensity edges has been avoided. Second, the eyelid tracking does not depend on the detection of other eye features. Third, the eyelid motion is tracked using a continuous facial action. Experiments on real video sequences including low-resolution videos indicate that the eye state can be detected using the eyelid tracking results.

**Acknowledgments** The authors thank Dr. Franck Davoine from CNRS, Compiegne, France, for providing the video sequence shown in Figure 10.

## References

1. Ahlberg, J.: Real-time facial feature tracking using an active model with fast image warping. In: International Workshop on Very Low Bitrate Video (VLBV). Athens, Greece (2001)
2. Ahlberg, J.: An active model for facial feature tracking. EURASIP J. Appl. Signal Process. **2002**(6), 566–571 (2002)
3. Ahlberg, J: Model-based coding: extraction, coding, and evaluation of face model parameters. Ph.D. thesis, No. 761, Linköping University, Sweden (2002)
4. Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models. IEEE Trans. Pattern Anal. Mach. Intell. **22**(4), 322–336 (2000)
5. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6): 681–684 (2001)
6. Dornaika, F., Davoine, F.: On appearance based face and facial action tracking. IEEE Trans. Circuits Syst. Video Technol. **16**(9):1107–1124 (2006)
7. Dornaika, F., Orozco, J., Gonzalez, J.: Combined head, lips, eyebrows, and eyelids tracking using adaptive appearance models. In: LNCS 4069. IV Conference on Articulated Motion and Deformable Objects, pp. 110–119 (2006)
8. Grauman, K., Betke, M., Gips, J., Bradski, G.R.: Communication via eye blinks: detection and duration analysis in real time. In: International Conference on Computer Vision and Pattern Recognition (2001)
9. Jepson, A., Fleet, D., El-Maraghi, T. Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(10):1296–1311 (2003)
10. Lee, D.: Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5):827–832 (2005)

11. Liu, H., Wu, Y., Zha, H.: Eye states detection from color facial image sequence. In: SPIE international conference on image and graphics, vol. 4875, pp. 693–698 (2002)
12. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Comput. Vis. **60**(2):135–164 (2004)
13. Moriyama, T., Kanade, T., Cohn, J., Xiao, J., Ambadar, Z., Gao, J., Imamura, H.: Automatic recognition of eye blinking in spontaneously occuring behavior. In: International Conference on Pattern Recognition (2002)
14. Sirohey, S., Rosenfeld, A., Duric, Z.: A method of detecting and tracking irises and eyelids in video. Pattern Recognit. **35**(6):1389–1401 (2002)
15. Tan, H., Zhang, Y.J.: Detecting eye blink states by tracking iris and eyelids. Pattern Recognit. Lett. **27**(6):667–675 (2006)
16. Tian, Y., Kanade, T., Cohn, J.F.: Dual-state parametric eye tracking. In: International Conference on Automatic Face and Gesture Recognition (2000)
17. Uzunova, V.I.: An eyelids and eye corners detection and tracking method for rapid iris tracking. Master's thesis, University of Magdeburg, 2005
18. Zhou, S., Chellappa, R., Mogghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Trans. Image Process. **13**(11):1473–1490 (2004)
19. Zhu, J., Yang, J.: Subpixel eye gaze tracking. In: International Conference on Automatic Face and Gesture Recognition (2002)