2

Facial Expression Recognition in the Presence of Head Motion

Fadi Dornaika[1] and Franck Davoine[2]
National Geographical Institute (IGN), 2 avenue Pasteur, 94165 Saint-Mandé [1],
Heudiasyc Mixed Research Unit, CNRS/UTC, 60205 Compiègne [2],
France

1. Introduction

The human face has attracted attention in a number of areas including psychology, computer vision, human-computer interaction (HCI) and computer graphics (Chandrasiri et al, 2004). As facial expressions are the direct means of communicating emotions, computer analysis of facial expressions is an indispensable part of HCI designs. It is crucial for computers to be able to interact with the users, in a way similar to human-to-human interaction. Human-machine interfaces will require an increasingly good understanding of a subject's behavior so that machines can react accordingly. Although humans detect and analyze faces and facial expressions in a scene with little or no effort, development of an automated system that accomplishes this task is rather diffcult.

One challenge is to construct robust, real-time, fully automatic systems to track the facial features and expressions. Many computer vision researchers have been working on tracking and recognition of the whole face or parts of the face. Within the past two decades, much work has been done on automatic recognition of facial expression. The initial 2D methods had limited success mainly because their dependency on the camera viewing angle. One of the main motivations behind 3D methods for face or expression recognition is to enable a broader range of camera viewing angles (Blanz & Vetter, 2003; Gokturk et al., 2002; Lu et al., 2006; Moreno et al., 2002; Wang et al., 2004; Wen & Huang, 2003; Yilmaz et al., 2002).

To classify expressions in static images many techniques have been proposed, such as those based on neural networks (Tian et al., 2001), Gabor wavelets (Bartlett et al., 2004), and Adaboost (Wang et al., 2004). Recently, more attention has been given to modeling facial deformation in dynamic scenarios, since it is argued that information based on dynamics is richer than that provided by static images. Static image classifiers use feature vectors related to a single frame to perform classification (Lyons et al., 1999). Temporal classifiers try to capture the temporal pattern in the sequence of feature vectors related to each frame. These include the Hidden Markov Model (HMM) based methods (Cohen et al., 2003) and Dynamic Bayesian Networks (DBNs) (Zhang & Ji, 2005). In (Cohen et al., 2003), the authors introduce a facial expression recognition from live video input using temporal cues. They propose a new HMM architecture for automatically segmenting and recognizing human facial expression from video sequences. The architecture performs both segmentation and recognition of the facial expressions automatically using a multi-level architecture composed of an HMM layer and a Markov model layer. In (Zhang & Ji, 2005), the authors present a new approach to spontaneous facial expression understanding in image sequences. The facial feature detection and tracking is based on active Infra Red illumination. Modeling dynamic behavior of facial expression in image sequences falls within the framework of information fusion with DBNs. In (Xiang et al., 2008), the authors propose a temporal classifier based on the use of fuzzy C means where the features are given by Fourrier transform.

Surveys of facial expression recognition methods can be found in (Fasel & Luettin, 2003; Pantic & Rothkrantz, 2000). A number of earlier systems were based on facial motion encoded as a dense flow between successive image frames. However, flow estimates are easily disturbed by illumination changes and non-rigid motion. In (Yacoob & Davis, 1996), the authors compute optical flow of regions on the face, then they use a rule-based classifier to recognize the six basic facial expressions. Extracting and tracking facial actions in a video can be done in several ways. In (Bascle & Black, 1998), the authors use active contours for tracking the performer's facial deformations. In (Ahlberg, 2002), the author retrieves facial actions using a variant of Active Appearance Models. In (Liao & Cohen, 2005), the authors used a graphical model for modeling the interdependencies of defined facial regions for characterizing facial gestures under varying pose. The dominant paradigm involves computing a time-varying description of facial actions/features from which the expression can be recognized; that is to say, the tracking process is performed prior to the recognition process (Dornaika & Davoine, 2005; Zhang & Ji, 2005).

However, the results of both processes affect each other in various ways. Since these two problems are interdependent, solving them simultaneously increases reliability and robustness of the results. Such robustness is required when perturbing factors such as partial occlusions, ultra-rapid movements and video streaming discontinuity may affect the input data. Although the idea of merging tracking and recognition is not new, our work addresses two complicated tasks, namely tracking the facial actions and recognizing expression over time in a monocular video sequence.

In the literature, simultaneous tracking and recognition has been used in simple cases. For example, (North et al., 2000) employs a particle-filter-based algorithm for tracking and recognizing the motion class of a juggled ball in 2D. Another example is given in (Zhou et al., 2003); this work proposes a framework allowing the simultaneous tracking and recognizing of human faces using a particle filtering method. The recognition consists in determining a person's identity, which is fixed for the whole probe video. The authors use a mixed state vector formed by the 2D global face motion (affine transform) and an identity variable. However, this work does not address either facial deformation or facial expression recognition.

In this chapter, we describe two frameworks for facial expression recognition given natural head motion. Both frameworks are texture- and view-independent. The first framework exploits the temporal representation of tracked facial action in order to infer the current facial expression in a deterministic way. The second framework proposes a novel paradigm in which facial action tracking and expression recognition are simultaneously performed. The second framework consists of two stages. First, the 3D head pose is estimated using a deterministic approach based on the principles of Online Appearance Models (OAMs). Second, the facial actions and expression are simultaneously estimated using a stochastic approach based on a particle filter adopting mixed states (Isard & Blake, 1998). This proposed framework is simple, efficient and robust with respect to head motion given that (1) the dynamic models directly relate the facial actions to the universal expressions, (2) the learning stage does not deal with facial images but only concerns the estimation of auto-regressive models from sequences of facial actions, which is carried out using closed- from solutions, and (3) facial actions are related to a deformable 3D model and not to entities measured in the image plane.

1.1 Outline of the chapter

This chapter provides a set of recent deterministic and stochastic (robust) techniques that perform efficient facial expression recognition from video sequences. The chapter organization is as follows. The first part of the chapter (Section 2) briefly describes a real time face tracker adopting a deformable 3D mesh and using the principles of Online Appearance Models. This tracker can provide the 3D head pose parameters and some facial actions. The second part of the chapter (Section 3) focuses on the analysis and recognition of facial expressions in continuous videos using the tracked facial actions. We propose two pose- and texture-independent approaches that exploit the tracked facial action parameters. The first approach adopts a Dynamic Time Warping technique for recognizing expressions where the training data are a set of trajectory examples associated with universal facial expressions. The second approach models trajectories associated with facial actions using Linear Discriminant Analysis. The third part of the chapter (Section 4) addresses the simultaneous tracking and recognition of facial expressions. In contrast to the mainstream approach "tracking then recognition", this framework simultaneously retrieves the facial actions and expression using a particle filter adopting multi-class dynamics that are conditioned on the expression.

2. Face and facial action tracking

2.1 A deformable 3D model

In our study, we use the *Candide* 3D face model (Ahlberg, 2002). This 3D deformable wireframe model was first developed for the purposes of model-based image coding and computer animation. The 3D shape of this wireframe model (triangular mesh) 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:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{S}\,\boldsymbol{\tau}_\mathbf{s} + \mathbf{A}\,\boldsymbol{\tau}_\mathbf{a} \tag{1}$$

where $\overline{g}$ is the standard shape of the model, τ_s and τ_a are shape and animation control vectors, respectively, and the columns of $\mathbf{S}$ and $\mathbf{A}$ are the Shape and Animation Units. A Shape Unit provides a means of deforming the 3D wireframe so as to be able to adapt eye width, head width, eye separation distance, etc. Thus, the term $\mathbf{S}\,\tau_s$ accounts for shape variability (inter-person variability) while the term $\mathbf{A}\,\tau_a$ accounts for the facial animation (intra-person variability). The shape and animation variabilities can be approximated well enough for practical purposes by this linear relation. Also, we assume that the two kinds of variability are independent. With this model, the ideal neutral face configuration is represented by $\tau_a = \mathbf{0}$. The shape modes were created manually to accommodate the

subjectively most important changes in facial shape (face height/width ratio, horizontal and vertical positions of facial features, eye separation distance). Even though a PCA was initially performed on manually adapted models in order to compute the shape modes, we preferred to consider the *Candide* model with manually created shape modes with semantic signification that are easy to use by human operators who need to adapt the 3D mesh to facial images. The animation modes were measured from pictorial examples in the Facial Action Coding System (FACS) (Ekman & Friesen, 1977).
In this study, we use twelve modes for the facial Shape Units matrix **S** and six modes for the facial Animation Units (AUs) matrix **A**. Without loss of generality, we have chosen the six following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer and outer eyebrow raiser. These AUs are enough to cover most common facial animations (mouth and eyebrow movements). Moreover, they are essential for conveying emotions. The effects of the Shape Units and the six Animation Units on the 3D wireframe model are illustrated in Figure 1.

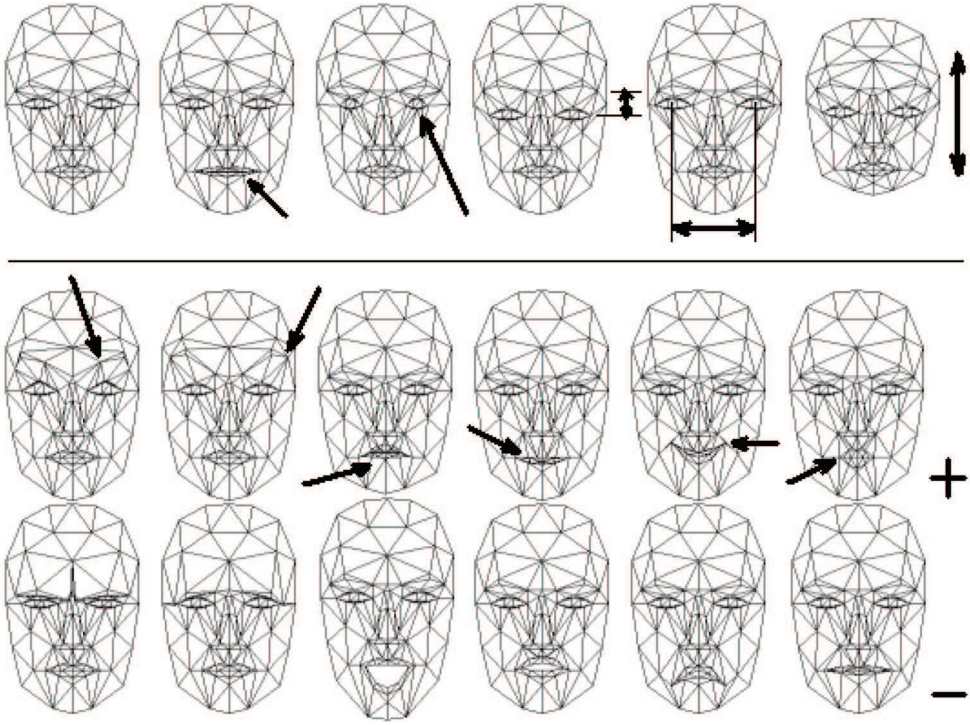


Figure 1: First row: Facial Shape units (neutral shape, mouth width, eyes width, eyes vertical position, eye separation distance, head height). Second and third rows: Positive and negative perturbations of Facial Action Units (Brow lowerer, Outer brow raiser, Jaw drop, Upper lip raiser, Lip corner depressor, Lip stretcher).

In equation (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. Therefore, the mapping

between the 3D face model and the image is given by a 2×4 matrix, **M**, encapsulating both the 3D head pose and the camera parameters.
Thus, a 3D vertex $\mathbf{P}_i = (X_i, Y_i, Z_i)^T \subset \mathbf{g}$ will be projected onto the image point $\mathbf{p}_i = (u_i, v_i)^T$ given by:

$$(u_i, v_i)^T = \mathbf{M}\,(X_i, Y_i, Z_i, 1)^T \quad (2)$$

For a given subject, τ_s is constant. Estimating τ_s can be carried out using either feature-based (Lu et al., 2001) or featureless approaches (Ahlberg, 2002). In our work, we assume that the control vector τ_s is already known for every subject, and it is set manually using for instance the face in the first frame of the video sequence (the *Candide* model and target face shapes are aligned manually). Therefore, Equation (1) becomes:

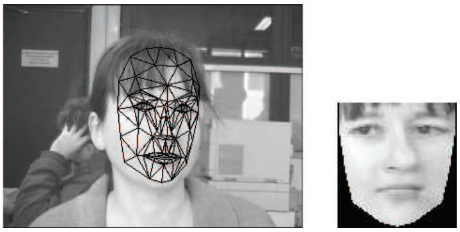
$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\,\boldsymbol{\tau_a} \quad (3)$$

where $\mathbf{g}_s$ represents the static shape of the face-the neutral face configuration. Thus, the state of the 3D wireframe model is given by the 3D head pose parameters (three rotations and three translations) and the animation control vector τ_a. This is given by the 12-dimensional vector **b**:

$$\mathbf{b} = [\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, \boldsymbol{\tau}_\mathbf{a}^T]^T \quad (4)$$

$$= [\mathbf{h}^T, \boldsymbol{\tau}_\mathbf{a}^T]^T \quad (5)$$

where the vector **h** represents the six degrees of freedom associated with the 3D head pose.



(a) (b)

Figure 2: (a) an input image with correct adaptation of the 3D model. (b) the corresponding shape-free facial image.

2.2 Shape-free facial patches

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

$$\mathbf{x}(\mathbf{b}) = \mathcal{W}(\mathbf{y}, \mathbf{b}) \quad (6)$$

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

2.3 Adaptive facial texture model

In this work, the facial texture model (appearance model) is built online using the tracked shape-free patches. We use the HAT symbol for the tracked parameters and patches. For a given frame t, $\hat{b}_t$ represents the computed geometric parameters and $\hat{x}_t$ the corresponding shape-free patch, that is,

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

The estimation of $\hat{b}_t$ from the sequence of images will be presented in Section 2.4. $\hat{b}_0$ is set manually, according to the face in the first video frame. The facial texture model (appearance model) associated with the shape-free facial patch at time t is time-varying in that it models the appearances present in all observations $\hat{x}$ up to time t - 1. This may be required as a result, for instance, of illumination changes or out-of-plane rotated faces.

By assuming that the pixels within the shape-free patch are independent, we can model the appearance using a multivariate Gaussian with a diagonal covariance matrix Σ. In other words, this multivariate Gaussian is the distribution of the facial patches $\hat{x}_t$. Let μ be the Gaussian center and σ the vector containing the square root of the diagonal elements of the covariance matrix Σ. μ and σ are d-vectors (d is the size of x).

In summary, the observation likelihood is written as:

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

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

$$\mathbf{N}(x_i; \mu_i, \sigma_i) = (2\pi\sigma_i^2)^{-1/2} \exp\left[-\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right] \tag{9}$$

We assume that the appearance model summarizes the past observations under an exponential envelope with a forgetting factor $\alpha = 1 - \exp\left(-\frac{\log 2}{n_h}\right)$, where n_h represents the half-life of the envelope in frames (Jepson et al., 2003).

When the patch $\hat{x}_t$ is available at time t, the appearance is updated and used to track in the next frame. It can be shown that the appearance model parameters, i.e., the μ_i's and σ_i's can be updated from time t to time (t + 1) using the following equations (see (Jepson et al., 2003) for more details on OAMs):

$$\mu_{i_{(t+1)}} = (1 - \alpha)\, \mu_{i_{(t)}} + \alpha\, \hat{x}_{i_{(t)}} \tag{10}$$

$$\sigma^2_{i_{(t+1)}} = (1-\alpha)\,\sigma^2_{i_{(t)}} + \alpha\,(\hat{x}_{i_{(t)}} - \mu_{i_{(t)}})^2 \tag{11}$$

This technique 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 μ is initialized with the first patch $\hat{x}_0$. However, equation (11) is not used with α being a constant 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, equation (11) is used with α being set to *1/t* .

Here we used a single Gaussian to model the appearance of each pixel in the shape-free template. However, modeling the appearance with Gaussian mixtures can also be used at the expense of an additional computational load (e.g., see (Lee, 2005; Zhou et al., 2004)).

2.4 Face and facial action tracking

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 state vector $\mathbf{b}_t$ (equation 5).

The purpose of the tracking is to estimate the state vector $\mathbf{b}_t$ by using the current appearance model encoded by μ_t and σ_t. To this end, the current input image $\mathbf{y}_t$ is registered with the current appearance model. The state vector $\mathbf{b}_t$ is estimated by minimizing the *Mahalanobis* distance between the warped image patch and the current appearance mean - the current Gaussian center

$$\min_{\mathbf{b}} e(\mathbf{b}_t) = \min d\left[\mathbf{x}(\mathbf{b}_t), \boldsymbol{\mu}_t\right] = \min \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^2_{(t)} \tag{12}$$

The above criterion can be minimized using an iterative gradient descent method where the starting solution is set to the previous solution $\hat{b}_{t-1}$. Handling outlier pixels (caused for instance by occlusions) is performed by replacing the quadratic function by the Huber's cost function (Huber, 1981). The gradient matrix is computed for each input frame. It is approximated by numerical differences. More details about this tracking method can be found in (Dornaika & Davoine, 2006).

3. Tracking then recognition

In this section, we show how the time series representation of the estimated facial actions, τ_a, can be utilized for inferring the facial expression in continuous videos. We propose two different approaches. The first one is a non-parametric approach and relies on Dynamic Time Warping. The second one is a parametric approach and is based on Linear Discriminant Analysis.

In order to learn the spatio-temporal structure of the facial actions associated with the universal expressions, we have used the following. Video sequences have been picked up from the CMU database (Kanade et al., 2000). These sequences depict five frontal view universal expressions (surprise, sadness, joy, disgust and anger). Each expression is performed by 7 different subjects, starting from the neutral one. Altogether we select 35 video sequences composed of around 15 to 20 frames each, that is, the average duration of each sequence is about half a second. The learning phase consists in estimating the facial action parameters τ_a (a 6-vector) associated with each training sequence, that is, the temporal trajectories of the action parameters.

Figure 3 shows six videos belonging to the CMU database. The first five images depict the estimated deformable model associated with the high magnitude of the five basic expressions. Figure 4 shows the computed facial action parameters associated with three training sequences: surprise, joy and anger. The training video sequences have an interesting property: all performed expressions go from the neutral expression to a high magnitude expression by going through a moderate magnitude around the middle of the sequence.

Surprise

Sadness

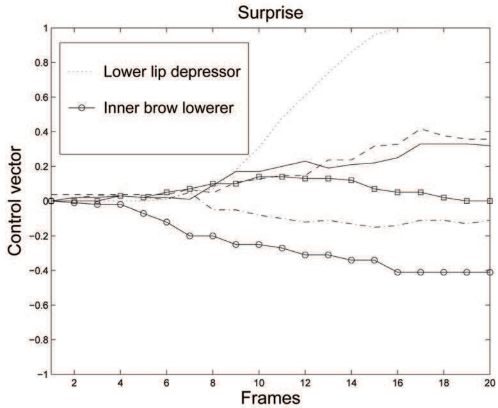
Joy

Disgust

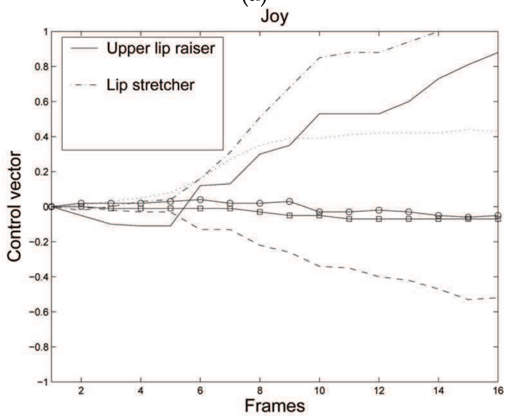
Anger

Neutral

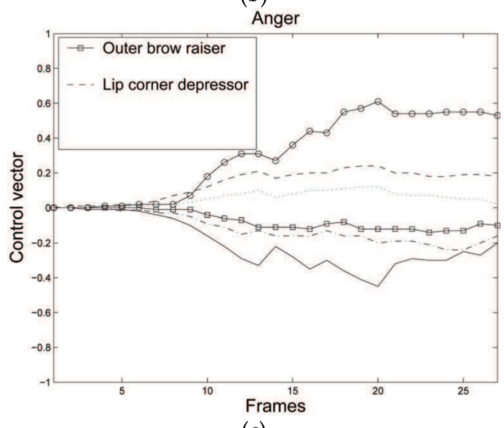
Figure 3: Six video examples associated with the CMU database. The first five images depict the high magnitude of the five basic expressions.



(a)



(b)



(c)

Figure 4: Three examples (sequences) of learned facial action parameters as a function of time. (a) Surprise expression. (b) Joy expression. (c) Anger expression.

3.1 Dynamic time warping

In the recognition phase, the head and facial actions are recovered from the video sequence using our developed appearance-based tracker (Dornaika & Davoine, 2006). The current facial expression is then recognized by computing a similarity measure between the tracked facial actions $\tau_{a(t)}$ associated with the test sequence and those associated with each universal expression. This recognition scheme can be carried out either online or off-line. One can notice that a direct comparison between the tracked trajectories and the stored ones is not feasible since there is no frame-to-frame correspondence between the tracked facial actions and the stored ones. To overcome this problem, we use dynamic programming which allows temporal deformation of time series as they are matched against each other.

We infer the facial expression associated with the current frame t by considering the estimated trajectory, i.e. the sequence of vectors $\tau_{a(t)}$, within a temporal window of size T centered at the current frame t. In our tests, T is set to 9 frames. This trajectory is matched against the 35 training trajectories using the Dynamic Time Warping (DTW) technique (Rabiner & Juang, 1993; Berndt & Clifford, 1994). For each training trajectory, the DTW technique returns a dissimilarity measure between the tested trajectory and the training trajectory (known universal expression). The classification rule stipulates that the smallest average dissimilarity decides the expression classifcation where the dissimilarity measures associated with a given universal expression are averaged over the 7 subjects.

The proposed scheme accounts for the variability in duration since the DTW technique allows non-linear time scaling. The segmentation of the video is obtained by repeating the whole recognition scheme for every frame in the test video.

In order to evaluate the performance, we have created test videos featuring the universal facial expressions. To this end, we have asked a volunteer student to perform each universal expression several times in a relatively long sequence. The subject was instructed to display the expression in a natural way, i.e. the displayed expressions were independent of any database. Each video sequence contains several cycles depicting a particular universal facial expression.

The performance of the developed recognition scheme is evaluated by utilizing five test videos. Table 1 shows the confusion matrix for the dynamical facial expression classifier using the DTW technique. We point out that the learned trajectories were inferred from the CMU database while the used test videos were created at our laboratory. The recognition rate of dynamical expressions was 100% for all universal expressions except for the disgust expression for which the recognition rate was 44%. The reason is that the disgust expression performed by our subject was very different from that performed by most of the CMU database subjects. Therefore, for the above experiment, the overall recognition rate is 90.4%.

	Surp.	Sad.	Joy	Disg.	Ang.
Surp.	14	0	0	0	0
Sad.	0	9	0	0	0
Joy	0	0	10	5	0
Disg.	0	0	0	4	0
Ang.	0	0	0	0	10

Table 1: Confusion matrix for the dynamical facial expression classifier using the DTW technique (the smallest average similarity). The learned trajectories were inferred from the CMU database while the used test videos were created at our laboratory. The recognition rate of dynamical expressions was 100% for all basic expressions except for the disgust expression for which the recognition rate was 44%.

3.2 Linear discriminant analysis

As can be seen from the previous section, the CPU time of the recognition scheme based on the DTW technique is proportional to the number of the subjects present in the database. Whenever this number is very large, the recognition scheme becomes computationally expensive. In this section, we propose a parametric recognition scheme by which the training trajectories can be represented in a more compact form. The computational cost of the recognition scheme does not depend on the number of examples.

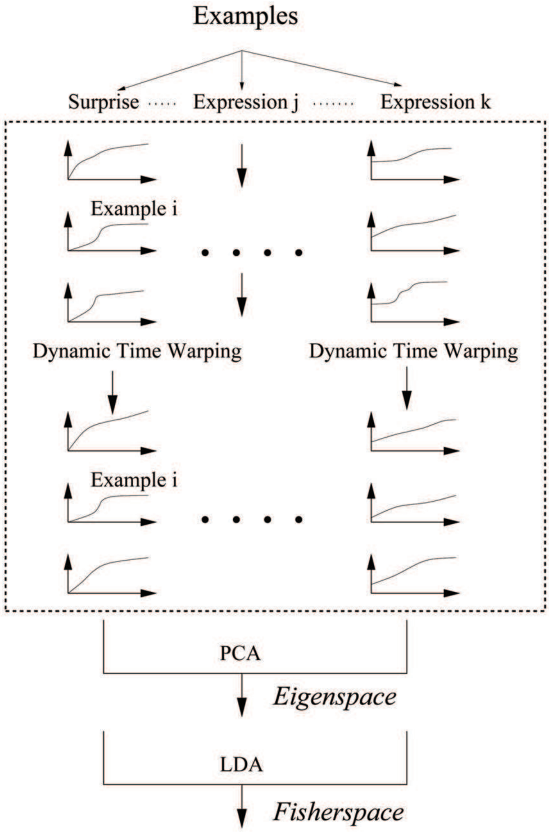


Figure 5: The parameterized modeling of facial expressions using Eigenspace and Fisherspace.

Learning. The learning phase is depicted in Figure 5. Again, we use the training videos associated with the CMU database. In order to obtain trajectories with the same number of frames (duration) the trajectories belonging to the same expression class are aligned using the DTW technique. Recall that this technique allows a frame-to-frame correspondence between two time series.

Let e_i^j be the aligned trajectory i belonging to the expression class j. The example e_i^j is represented by a column vector of dimension $1 \times 6T$ and is obtained by concatenating the facial action 6-vectors $\tau_{a(t)}$:

$$\mathbf{e}_i^j = [\boldsymbol{\tau}_{\mathbf{a}(1)}; \boldsymbol{\tau}_{\mathbf{a}(2)}; \ldots; \boldsymbol{\tau}_{\mathbf{a}(T)}]$$

Note that T represents the duration of the aligned trajectories which will be fixed for all examples. For example, a nominal duration of 18 frames for the aligned trajectories makes the dimension of all examples e_i^j (all i and j) equal to 108.

Applying a Principal Component Analysis on the set of all training trajectories yields the mean trajectory **e** as well as the principal modes of variation. Any training trajectory **e** can be approximated by the principal modes using the q largest eigenvalues:

$$\begin{aligned} \mathbf{e} &\cong \overline{\mathbf{e}} + \mathbf{U}\,\mathbf{c} \\ &= \overline{\mathbf{e}} + \sum_{l=1}^{q} c_l\,\mathbf{U}_l \end{aligned}$$

In our work, the number of principal modes is chosen such that the variability of the retained modes corresponds to 99% of the total variability. The vector **c** can be seen as a parametrization of any input trajectory, $\hat{e}$, in the space spanned by the q basis vectors $\mathbf{U}_l$. The vector **c** is given by:

$$\mathbf{c} = \mathbf{U}^T\,(\hat{\mathbf{e}} - \overline{\mathbf{e}}) \tag{13}$$

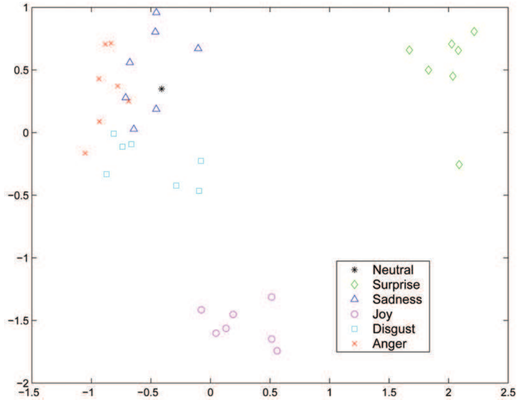
Thus, all training trajectories e_i^j can now be represented by the vectors c_i^j (using (13)) on which a Linear Discriminant Analysis can be applied. This gives a new space (the Fisherspace) in which each training video sequence is represented by a vector of dimension l -1 where l is the number of expression classes. Figure 6 illustrates the learning results associated with the CMU data. In this space, each trajectory example is represented by a 5-vector. Here, we use six facial expression classes: Surprise, Sadness, Joy, Disgust, Anger, and Neutral. (a) displays the second component versus the first one, and (b) displays the fourth component versus the third one. In this space, the neutral trajectory (a sequence of zero vectors) is represented by a star.

Recognition. The recognition scheme follows the main steps of the learning stage. We infer the facial expression by considering the estimated facial actions provided by our face tracker (Dornaika & Davoine, 2006). We consider the one-dimensional vector **e'** (the concatenation of the facial actions $\tau_{a(t)}$) within a temporal window of size T centered at the current frame t. Note that the value of T should be the same as in the learning stage. This vector is projected onto the PCA space, then the obtained vector is projected onto Fisherspace in which the classification occurs. The expression class whose mean is the closest to the current trajectory is then assigned to this trajectory (current frame).

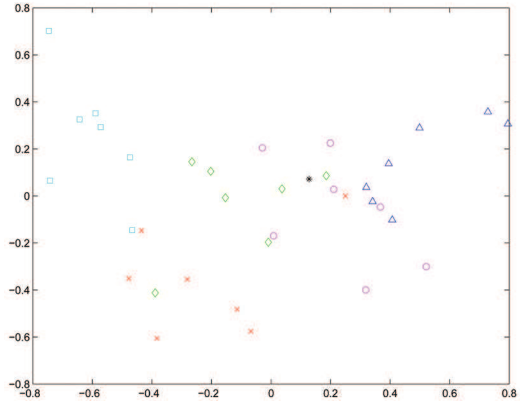
Preformance evaluation. Table 2 shows the confusion matrix for the dynamical facial expression classifier using Eigenspace and Fisherspace. The learned trajectories were inferred from the CMU database while the used test videos were created at our laboratory. The recognition rate of dynamical expressions was 100% for all basic expressions except for the disgust expression for which the recognition rate was 55%. Therefore, for the above experiment, the overall recognition rate is 92.3%. One can notice the slight improvement in the recognition rate over the classical recognition scheme based on the DTW.

	Surp.	Sad.	Joy	Disg.	Ang.
Surp.	14	0	0	0	0
Sad.	0	9	0	0	0
Joy	0	0	10	4	0
Disg.	0	0	0	5	0
Ang.	0	0	0	0	10

Table 2: Confusion matrix for the dynamical facial expression classifier using Eigenspace and Fisherspace. The learned trajectories were inferred from the CMU database while the used test videos were created at our laboratory. The recognition rate of dynamical expressions was 100% for all basic expressions except for the disgust expression for which the recognition rate was 55%.



(a)



(b)

Figure 6: The 35 trajectory examples associated with five universal facial expressions depicted in Fisherspace. In this space, each trajectory example is represented by a 5-vector. Here, we use six facial expression classes: Surprise, Sadness, Joy, Disgust, Anger, and Neutral. (a) displays the second component versus the first one, and (b) displays the fourth component versus the third one. In this space, the neutral trajectory (a sequence of zero vectors) is represented by a star.

4. Tracking and recognition

In Section 3, the facial expression was inferred from the time series representation of the tracked facial actions. In this section, we propose to simultaneously estimate the facial actions and the expression from the video sequence.
Since the facial expression can be considered as a random discrete variable, we need to append to the continuous state vector $\mathbf{b}_t$ a discrete state component γ_t in order to create a mixed state:

$$\begin{pmatrix} \mathbf{b}_t \\ \gamma_t \end{pmatrix} \tag{14}$$

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

$$\begin{pmatrix} \mathbf{h}_t \\ \mathbf{a}_t \end{pmatrix} \tag{15}$$

where $\mathbf{h}_t$ denotes the 3D head pose parameters, and $\mathbf{a}_t$ the facial actions appended with the expression label γ_t, i.e. $\mathbf{a}_t = [\tau_{a(t)}^T, \gamma_t]^T$.
This separation is consistent with the fact that the facial expression is highly correlated with the facial actions, while the 3D head pose is independent of the facial actions and expressions. The remainder of this section is organized as follows. Section 4.1 provides some backgrounds. Section 4.2 describes the proposed approach for the simultaneous tracking and recognition. Section 4.3 describes experiments and provides evaluations of performance to show the feasibility and robustness of the proposed approach.

4.1 Backgrounds

4.1.1 Facial action dynamic models

Corresponding to each basic expression class, γ, there is a stochastic dynamic model describing the temporal evolution of the facial actions $\tau_{a(t)}$, given the expression. It is assumed to be a Markov model of order *K*. For each basic expression γ, we associate a Gaussian Auto-Regressive Process defined by:

$$\boldsymbol{\tau}_{\mathbf{a}(t)} = \sum_{k=1}^{K} \mathbf{A}_k^\gamma \boldsymbol{\tau}_{\mathbf{a}(t-k)} + \mathbf{d}^\gamma + \mathbf{B}^\gamma \mathbf{w}_t \tag{16}$$

in which $\mathbf{w}_t$ is a vector of 6 independent random $N(0, 1)$ variables. The parameters of the dynamic model are: (i) deterministic parameters $A_1^\gamma, A_2^\gamma, \ldots, A_K^\gamma$ and $\mathbf{d}^\gamma$, and stochastic parameters $\mathbf{B}^\gamma$ which are multipliers for the stochastic process $\mathbf{w}_t$. It is worth noting that the above model can be used in predicting the process from the previous *K* values. The predicted value at time t obeys a multivariate Gaussian centered at the deterministic value of (16), with $\mathbf{B}^{\gamma}\mathbf{B}^{\gamma T}$ being its covariance matrix. In our study, we are interested in second-order models, i.e. K = 2. The reason is twofold. First, these models are easy to estimate. Second, they are able to model complex dynamics. For example, these models have been used in (Blake & Isard, 2000) for learning the 2D motion of talking lips (profile contours), beating heart, and writing fingers.

4.1.2 Learning the second-order auto-regressive models

Given a training sequence $\tau_{a(1)},\ldots,\tau_{a(T)}$, with $T > 2$, belonging to the same expression class, it is well known that a Maximum Likelihood Estimator provides a closed-form solution for the model parameters (Blake & Isard, 2000). For a second-order model, these parameters reduce to two 6×6 matrices A_1^{γ}, A_2^{γ}, a 6-vector $\mathbf{d}^{\gamma}$, and a 6 × 6 covariance matrix $\mathbf{C}^{\gamma} = \mathbf{B}^{\gamma}\,\mathbf{B}^{\gamma\,T}$. Therefore, Eq. (16) reduces to:

$$\boldsymbol{\tau}_{\mathbf{a}(t)} = \mathbf{A}_2^{\gamma}\,\boldsymbol{\tau}_{\mathbf{a}(t-2)} + \mathbf{A}_1^{\gamma}\,\boldsymbol{\tau}_{\mathbf{a}(t-1)} + \mathbf{d}^{\gamma} + \mathbf{B}^{\gamma}\,\mathbf{w}_t \quad (17)$$

The parameters of each auto-regressive model can be computed from temporal facial action sequences. Ideally, the temporal sequence should contain several instances of the corresponding expression.

More details about auto-regressive models and their computation can be found in (Blake & Isard, 2000; Ljung, 1987; North et al., 2000). Each universal expression has its own second-order auto-regressive model given by Eq.(17). However, the dynamics of facial actions associated with the neutral expression can be simpler and are given by:

$$\tau_{a(t)} = \tau_{a(t-1)} + \mathrm{D}\mathrm{w}_t$$

where **D** is a diagonal matrix whose elements represent the variances around the ideal neutral configuration $\tau_a = \mathbf{0}$. The right-hand side of the above equation is constrained to belong to a predefined interval, since a neutral configuration and expression is characterized by both the lack of motion and the closeness to the ideal static configuration. In our study, the auto-regressive models are learned using a supervised learning scheme. First, we asked volunteer students to perform each basic expression several times in approximately 30-second sequences. Each video sequence contains several cycles depicting a particular facial expression: Surprise, Sadness, Joy, Disgust, Anger, and Fear. Second, for each training video, the 3D head pose and the facial actions $\tau_{a(t)}$ are tracked using our deterministic appearance-based tracker (Dornaika & Davoine, 2006) (outlined in Section 2). Third, the parameters of each auto-regressive model are estimated using the Maximum Likelihood Estimator.

Figure 7 illustrates the value of the facial actions, $\tau_{a(t)}$, associated with six training video sequences. For clarity purposes, only two components are shown for a given plot. For a given training video, the neutral frames are skipped from the original training sequence used in the computation of the auto- regressive models.

4.1.3 The transition matrix

In our study, the facial actions as well as the expression are simultaneously retrieved using a stochastic framework, namely the particle filtering method. This framework requires a transition matrix **T** whose entries $T_{\gamma',\gamma}$ describe the probability of transition between two expression labels γ' and γ. The transition probabilities need to be learned from training video sequences. In the literature, the transition probabilities associated with states (not necessarily facial expressions) are inferred using supervised and unsupervised learning techniques. However, since we are dealing with high level states (the universal facial expressions), we have found that a realistic *a priori* setting works very well. We adopt a 7 ×7 symmetric matrix whose diagonal elements are close to one (e.g. $T_{\gamma,\gamma} = 0.8$, that is, 80% of the transitions occur within the same expression class). The rest of the percentage is distributed equally among the expressions. In this model, transitions from one expression to another expression without going through the neutral one are allowed. Furthermore, this model adopts the most general case where all universal expressions have the same probability. However, according to the context of the application, one can adopt other transition matrices in which some expressions are more likely to happen than others.

4.2 Approach

Since at any given time, the 3D head pose parameters can be considered as independent of the facial actions and expression, our basic idea is to split the estimation of the unknown parameters into two main stages. For each input video frame $\mathbf{y}_t$, these two stages are invoked in sequence in order to recover the mixed state $[\boldsymbol{h}_t^T, \boldsymbol{a}_t^T]^T$. Our proposed approach is illustrated in Figure 8. In the first stage, the six degrees of freedom associated with the 3D head pose (encoded by the vector $\mathbf{h}_t$) are obtained using a deterministic registration technique similar to that proposed in (Dornaika & Davoine, 2006). In the second stage, the facial actions and the facial expression (encoded by the vector $\mathbf{a}_t = [\tau_{a(t)}^T, \gamma_t]^T$) are simultaneously estimated using a stochastic framework based on a particle filter. Such models have been used to track objects when different types of dynamics can occur (Isard & Blake, 1998). Other examples of auxiliary discrete variables beside the main hidden state of interest are given in (Perez & Vermaak, 2005). Since $\tau_{a(t)}$ and γ_t are highly correlated their simultaneous estimation will give results that are more robust and accurate than results obtained with methods estimating them in sequence. In the following, we present the parameter estimation process associated with the current frame $\mathbf{y}_t$. Recall that the head pose is computed using a deterministic approach, while the facial actions and expressions are estimated using a probabilistic framework.

4.2.1 3D head pose

The purpose of this stage is to estimate the six degrees of freedom associated with the 3D head pose at frame t, that is, the vector $\mathbf{h}_t$. Our basic idea is to recover the current 3D head pose parameters from the previous 12-vector $\hat{b}_{t-1} = [\hat{\theta}_{x(t-1)}, \hat{\theta}_{y(t-1)}, \hat{\theta}_{z(t-1)}, \hat{t}_{x(t-1)}, \hat{t}_{y(t-1)}, \hat{t}_{z(t-1)}, \hat{\tau}_{a(t-1)}^T]^T = [\hat{h}_{t-1}^T, \hat{\tau}_{a(t-1)}^T]^T$ using the same region-based registration technique outlined in Section 2.4. However, this time the unknown parameters are only given by the 3D head pose parameters:

$$\min_{\mathbf{h}} e(\mathbf{h}_t) = \min d\left[\mathbf{x}(\mathbf{b}_t), \boldsymbol{\mu}_t\right] = \min \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i}\right)_{(t)}^2 \tag{18}$$

4.2.2 Simultaneous facial actions and expression

In this stage, our goal is to simultaneously infer the facial actions as well as the expression label associated with the current frame t given (i) the observation model (Eq.(8)), (ii) the dynamics associated with each expression (Eq.(17)), and (iii) the 3D head pose for the current frame computed by the deterministic approach (see Section 4.2.1). This will be performed using a particle filter paradigm. Thus, the statistical inference of such paradigm will provide a posterior distribution for the facial actions $\tau_{a(t)}$ as well as a Probability Mass function for the facial expression γ_t.

Since the 3D head pose $\mathbf{h}_t$ is already computed, we are left with the mixed state $\mathbf{a}_t = [\tau_{a(t)}^T, \gamma_t]^T$. The dimension of the vector $\mathbf{a}_t$ is 7. Here we will employ a particle filter algorithm allowing the recursive estimation of the posterior distribution $p(\mathbf{a}_t | x_{1:(t)})$ using a particle set. This is approximated by a set of J particles $\{(\mathbf{a}_t^{(0)}, w_t^{(0)}), \ldots, (\mathbf{a}_t^{(J)}, w_t^{(J)})\}$. Once this distribution is known the facial actions as well as the expression can be inferred using some loss function such as the MAP or the mean. Figure 9 illustrates the proposed two-stage approach. It shows how the current posterior $p(\mathbf{a}_t | x_{1:(t)})$ can be inferred from the previous posterior $p(\mathbf{a}_{t-1} | x_{1:(t-1)})$ using a particle filter algorithm.

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

4.3 Experimental results

In this section, we first report results on simultaneous facial action tracking and expression recognition. Then we present performance studies, considering different perturbing factors such as robustness to rapid facial movements or to imprecise 3D head pose estimation.

4.3.1 Simultaneous tracking and recognition

Figure 10 shows the application of the proposed approach to a 748-frame test video sequence. The upper part of this figure shows 9 frames of this sequence: 50, 130, 221, 300, 371, 450, 500, 620, and 740. The two plots illustrate the probability of each expression as a function of time (frames). The lower part of this figure shows the tracking results associated with frames 130, 371, and 450. The upper left corner of these frames depicts the appearance mean and the current shape-free facial patch. Figure 11.a illustrates the weighted average of the tracked facial actions, $\hat{\tau}_{a(t)}$. For the sake of clarity, only three out of six components are shown. For this sequence, the maximum probability was correctly indicating the displayed expression. We noticed that some displayed expressions can, during a short initial phase (very few frames), be considered as a mixture of two expressions (the displayed one and another one). This is due to the fact that face postures and dynamics at some transition phases can be shared by more than one expression. This is not a problem since the frame-wise expression probabilities can be merged and averaged over a temporal patch including contiguous non-neutral frames. Figure 11.b illustrates this scheme and shows the resulting segmentation of the used test video. One remarks that this holds true for a human observer, who may fail to recognize a gesture from only one single frame.