In Proceedings of BMVC, Edimburgh, UK, September 2006

Efficiently estimating facial expression and illumination in appearance-based tracking

José M. Buenaposada†, Enrique Muñoz‡, Luis Baumela‡

†ESCET, U. Rey Juan Carlos
C/ Tulipán, s/n
28933 Móstoles, Spain

‡Facultad Informática, UPM
Campus de Montegancedo s/n
28660 Boadilla del Monte, Spain

http://www.dia.fi.upm.es/~pcr

Abstract

We introduce a subspace representation of face appearance which separates facial expressions from illumination variations. The appearance of a face is represented by the addition of two approximately independent linear subspaces modelling facial expressions and illumination respectively. The independence assumption notably simplifies the training of the system. We only require two image sequences. One in which one facial expression is subject to all possible illuminations and another in which the face, under one illumination, performs all facial expressions. This simple model enables us to train the system with no manual intervention. We also introduce an efficient procedure for fitting this model, which can be used for tracking a human face in real-time.

1 Introduction

Facial expression analysis plays an important role in many computer vision applications such as advanced human computer interfaces, lip reading, graphical animation or video-based face recognition. Tracking is generally posed as a minimisation problem. The tracker tries to minimise the discrepancies between a model and the actual configuration of the face in each image of the sequence. Appearance-based tracking approaches represent the face with a linear model of texture (appearance) variation [2, 10]. Changes in facial expressions can also be modelled by using linear subspace representations of facial appearance [6], or linear models of shape+texture such as the 2D *Active Appearance Models* (AAMs) [5] or the 3D *Morphable Models* (MMs) [3]. The main drawback of shape+texture approaches is that they have complex training procedures which often require manual intervention [4]. On the other hand, appearance-based representations are again gaining popularity, since there are various procedures for automatically learning linear [6, 11] subspace models and for probabilistically representing the dynamics of appearance variation [17, 7].

Factoring out some of the sources of appearance variation is also a key issue in many applications. For example, an automated graphical animation system would require the

tracker to separately estimate changes in appearance due to facial expressions and illumination, so that these changes could be re-targeted in a graphical model. Unfortunately, automated procedures for learning appearance-based models [6, 13, 11] cannot automatically factor the various sources of appearance variations represented in the model. In this paper we will introduce a subspace representation of face appearance which can be automatically trained and which separates facial expressions from illumination variations.

Separating illumination changes from other sources of variations in the appearance of the face has traditionally been studied for the construction of face recognition systems, either using subspace [1], or geometrical [8] approaches. Subspace approaches have also been used to separate multiple orthogonal factors using bilinear [16, 9] or multi-linear [18] models. These approaches cannot be used in a real-time tracker, either because they were conceived to analyse a single image [8], to be used in batch processing [16, 18], or because of the computational requirements of the minimisation procedure [9]. In the appearance model introduced in section 2 of this paper, a face is represented by the addition of two independent linear subspaces, one modelling the deformations of the face (facial expressions) and the second one the illumination. By using this model we will be able to train the system with no manual intervention (see subsection 4.1) and to build a real-time tracker.

Most applications not only require visual tracking algorithms to be robust to changes in the target appearance, but also to work in real-time. In section 3 we introduce a minimisation procedure which can efficiently fit the previous appearance model to a target image. It is directly related to the work of Hager and Belhumeur [10], whose tracking procedure is robust to changes in illumination, but assumes a rigid face. We have extended their approach to the case in which the target face deforms. In the experiments described in section 4 we show that, for the model introduced in section 2, our procedure performs better than the original factorisation approach of Hager and Belhumeur [10] and the more recent compositional approach of Matthews and Baker [14].

In summary, the main contributions of this paper are: a) we present an appearance-based model of the face which separates facial expressions from illumination and which can be automatically trained; b) we introduce an efficient procedure for fitting this model, which can be used for tracking a face in real-time.

2 The model

In this section we introduce an appearance-based model representing the variations in the appearance of a face caused by changes in the facial expressions and the illumination of the scene.

Let $I(\mathbf{x},t)$ be the image acquired at time t, where $\mathbf{x}$ is a vector representing the coordinates of a point in the image, and let $\mathbf{I}(\mathbf{x},t)$ be a vector storing the brightness values of $I(\mathbf{x},t)$. Let us assume that the target moves rigidly (with no deformation) between time instants t_0 and t, and that this motion can be described by the motion model $f(\mathbf{x},\mu)$, being μ the vector of rigid motion parameters. If there are no changes in the target appearance caused by the scene illumination, the brightness constancy equation $\mathbf{I}(f(\mathbf{x},\mu_t),t) = \mathbf{I}(\mathbf{x},t_0)$ holds. If the face is now allowed to deform non-rigidly, then we may write a new brightness constancy equation $\mathbf{I}(f(\mathbf{x},\mu_t),t) - [\mathtt{B}_d\mathbf{c}_{d,t}](\mathbf{x}) = \bar{\mathbf{I}}_d(\mathbf{x})$, where the non-rigid deformations have been modelled by a linear subspace with basis $\mathtt{B}_d$, mean value $\bar{\mathbf{I}}_d(\mathbf{x})$ and linear deformation parameters $\mathbf{c}_{d,t}$. By $[\mathtt{B}_d\mathbf{c}_{d,t}](\mathbf{x})$ we denote the value of

$\mathtt{B}_d\mathbf{c}_{d,t}$ for the pixel with position $\mathbf{x}$. Finally, for a given rigid motion μ_t and deformation $\mathbf{c}_{d,t}$, we could also model the illumination of the face by including a new subspace with basis $\mathtt{B}_i$ and linear illumination parameters $\mathbf{c}_i$, which represents all the possible illuminations of the mean face $\bar{\mathbf{I}}_d(\mathbf{x})$. So, the final brightness constancy equation is

$$\mathbf{I}(f(\mathbf{x},\mu_t),t) = \bar{\mathbf{I}}_d(\mathbf{x}) + [\mathtt{B}_i\mathbf{c}_{i,t}](\mathbf{x}) + [\mathtt{B}_d\mathbf{c}_{d,t}](\mathbf{x}) = \bar{\mathbf{I}}_d(\mathbf{x}) + [\mathtt{B}\mathbf{c}_t](\mathbf{x}) \;\; \forall \mathbf{x} \in \mathscr{F}, \qquad (1)$$

where $\mathtt{B} = [\mathtt{B}_i|\mathtt{B}_d]$, $\mathbf{c}_t^\top = (\mathbf{c}_{i,t}^\top, \mathbf{c}_{d,t}^\top)^\top$, $k = \dim(\mathbf{c}_t)$, and $\mathscr{F}$ represents the set of pixels of the face used for tracking. Vectors $\mathbf{c}_i$ and $\mathbf{c}_d$ are respectively the illumination and the deformation appearance parameters. The assumption that illumination and deformation subspaces are independent will simplify the training of the model: instead of having to use image sequences in which all combinations of illuminations and facial expressions are present, we will only have to process two image sequences, one with one facial expression and all illuminations and another with one illumination and all facial expressions (see section 4.1).

In order to validate the previous model we made the following experiment. First we trained it according to the procedure described in section 4.1. Then we manually selected the parameters of two facial expressions and two illuminations, and generated a set of intermediate illuminations and expressions by uniformly sampling the parameter space between those locations. We have repeated this process three times. The results are shown in Fig. 1. In spite of the linearity of the model, it correctly generates the appearance of the faces.

(a) (b) (c)

Figure 1: Images generated using our appearance model; (a) From left to right images generated falling eyebrows, and from top to down images generated varying illumination; (b) Now rotating eyes with a different illumination; (c) Now closing the mouth using an illumination different from the previous ones.

3 Efficient tracking

Tracking a face consists of estimating, for each image in the sequence, the values of the motion, μ, and appearance, $\mathbf{c}$, parameters which minimise the error function

$$E(\mu,\mathbf{c}) = ||\mathbf{I}(f(\mathbf{x},\mu_t),t) - \bar{\mathbf{I}}_d - [\mathtt{B}\mathbf{c}_t](\mathbf{x})||^2. \qquad (2)$$

In order to robustly estimate the minimum value of (2), the quadratic error norm can be replaced by a robust one (e.g. see [10]).

In general, minimising (2) can be a difficult task as it defines a non-convex cost function. Black and Jepson [2] presented an iterative solution by using a gradient descent procedure and a robust metric with increasing resolution levels. Their algorithm is not suitable for real-time performance, since, for example, the Jacobian of each incoming image has to be computed once on every frame for each level in the multi-resolution pyramid. Similar problems have been solved efficiently using Gauss-Newton minimisation [10, 14]. Hager and Belhumeur [10], in the context of invariance to illumination changes, introduced an efficient procedure for minimising (3) by assuming $\nabla_{\mathbf{x}}[\mathtt{B}\mathbf{c}](\mathbf{x}) \approx 0$. This assumption is valid approximation when modelling the illumination of a rigid head, but it cannot be reliably used for tracking faces whose appearance changes due to causes other than illumination (see section 4). In this section we will introduce an efficient procedure for minimising (2) without such restriction.

In order to make Gauss-Newton iterations, a Taylor series expansion of $\mathbf{I}$ at $(\mu_t, \mathbf{c}_t, t)$ is performed, producing a new error function

$$E(\delta\mu, \delta\mathbf{c}) = ||\mathtt{M}\delta\mu + \mathbf{I}(f(\mathbf{x}, \mu_t), t+\delta t) - \bar{\mathbf{I}}_d - \mathtt{B}(\mathbf{c}_t + \delta\mathbf{c})||^2, \tag{3}$$

where $\mathtt{M} = \left[\frac{\partial \mathbf{I}(f(\mathbf{x},\mu),t)}{\partial \mu}\Big|_{\mu=\mu_t} \right]$ is the $N \times n$ ($n = \dim(\mu)$) Jacobian matrix of $\mathbf{I}$.

3.1 Jacobian matrix factorisation

One of the obstacles for minimising (3) online, while tracking, is the computational cost of estimating $\mathtt{M}$ for each frame. In this subsection we will show that $\mathtt{M}$ can be factored into the product of two matrices, $\mathtt{M}_0\Sigma(\mu, \mathbf{c})$, where $\mathtt{M}_0$ is a constant matrix, which can be computed off-line.

Each row $m_i(\mu_t, \mathbf{c}_t)$ of $\mathtt{M}(\mu_t, \mathbf{c}_t)$ can be written as the product,

$$m_i(\mu_t, \mathbf{c}_t) = \nabla_{\mathbf{f}}\mathbf{I}(f(\mathbf{x}_i, \mu_t), t)^\top f_\mu(\mathbf{x}_i, \mu_t). \tag{4}$$

where $\nabla_{\mathbf{f}}\mathbf{I}(f(\mathbf{x}_i, \mu_t), t)^\top = \left[\frac{\partial \mathbf{I}(\mathbf{y},t)}{\partial \mathbf{y}}\Big|_{\mathbf{y}=f(\mathbf{x}_i,\mu_t)} \right]$ and $f_\mu(\mathbf{x}_i, \mu_t) = \left[\frac{\partial f(\mathbf{x}_i,\mu)}{\partial \mu}\Big|_{\mu=\mu_t} \right]$. Taking derivatives w.r.t. $\mathbf{x}$ on both sides of (1) we get

$$\nabla_{\mathbf{f}}\mathbf{I}(f(\mathbf{x}_i, \mu_t), t)^\top f_{\mathbf{x}}(\mathbf{x}_i, \mu_t) = \nabla_{\mathbf{x}}\bar{\mathbf{I}}_d(\mathbf{x}) + \nabla_{\mathbf{x}}[\mathtt{B}\mathbf{c}_t](\mathbf{x}), \tag{5}$$

where $f_{\mathbf{x}}(\mathbf{x}_i, \mu_t) = \left[\frac{\partial f(\mathbf{x},\mu_t)}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_i} \right]$ and $\nabla_{\mathbf{x}}$ denotes the image gradient. Finally, from (4) and (5) we get a new expression for $\mathtt{M}$,

$$\mathtt{M}(\mu, \mathbf{c}) = \begin{bmatrix} \mathtt{B}_\nabla(\mathbf{x}_1)\mathtt{C} f_{\mathbf{x}}(\mathbf{x}_1, \mu)^{-1} f_\mu(\mathbf{x}_1, \mu) \\ \vdots \\ \mathtt{B}_\nabla(\mathbf{x}_N)\mathtt{C} f_{\mathbf{x}}(\mathbf{x}_N, \mu)^{-1} f_\mu(\mathbf{x}_N, \mu) \end{bmatrix}, \tag{6}$$

where $\mathtt{B}_\nabla$ is the gradient of the subspace basis vector and $\mathtt{C}$ is a matrix storing $\mathbf{c}$. Therefore $\mathtt{M}$ can be expressed in terms of the gradient of the subspace basis vectors, $\mathtt{B}_\nabla$, which are constant, and the motion and appearance parameters $(\mu, \mathbf{c})$, which vary over time. If we

choose a motion model f such that $\mathtt{C} f_{\mathbf{x}}(\mathbf{x}_i,\mu)^{-1} f_{\mu}(\mathbf{x}_i,\mu) = \Gamma(\mathbf{x}_i)\Sigma(\mu,\mathbf{c})$, then $\mathtt{M}$ can be factored into

$$\mathtt{M}(\mu,\mathbf{c}) = \begin{bmatrix} \mathtt{B}_{\nabla}(\mathbf{x}_1)\Gamma(\mathbf{x}_1) \\ \vdots \\ \mathtt{B}_{\nabla}(\mathbf{x}_N)\Gamma(\mathbf{x}_N) \end{bmatrix} \Sigma(\mu,\mathbf{c}) = \mathtt{M}_0\Sigma(\mu,\mathbf{c}), \tag{7}$$

where $\mathtt{M}_0$ is constant matrix and Σ depends on $\mathbf{c}$ and μ.

3.2 Minimising $E(\mu,\mathbf{c})$

The minimum of (3) can be estimated by least-squares $[\delta\mu\ \delta\mathbf{c}]^{\top} = -(\mathtt{M}_J^{\top}\mathtt{M}_J)^{-1}\mathtt{M}_J\mathscr{E}$, where $\mathtt{M}_J = (\mathtt{M}|-\mathtt{B})$ and $\mathscr{E} = \mathbf{I}(f(\mathbf{x},\mu_t),t+\delta t) - \bar{\mathbf{I}}_d - \mathtt{B}\mathbf{c}_t$. Then, $\delta\mu = -(\mathtt{M}^{\top}\mathtt{N}_B\mathtt{M})^{-1}\mathtt{M}^{\top}\mathtt{N}_B\mathscr{E}$ and $\delta\mathbf{c} = (\mathtt{B}^{\top}\mathtt{N}_M\mathtt{B})^{-1}\mathtt{B}^{\top}\mathtt{N}_M\mathscr{E}$, where $\mathtt{N}_B = \mathtt{I} - \mathtt{B}(\mathtt{B}^{\top}\mathtt{B})^{-1}\mathtt{B}^{\top}$ and $\mathtt{N}_M = \mathtt{I} - \mathtt{M}(\mathtt{M}^{\top}\mathtt{M})^{-1}\mathtt{M}^{\top}$. Since $\mathtt{N}_B$ is a constant matrix, we get an efficient solution for $\delta\mu$ factoring $\mathtt{M}$ according to (7)

$$\delta\mu = -(\Sigma^{\top}\Lambda_{M1}\Sigma)^{-1}\Sigma^{\top}\Lambda_{M2}\mathscr{E}, \tag{8}$$

where $\Lambda_{M1} = \mathtt{M}_0^{\top}\mathtt{N}_B\mathtt{M}_0$ and $\Lambda_{M2} = \mathtt{M}_0^{\top}\mathtt{N}_B$ are constant and can be precomputed off-line. A similar solution for $\delta\mathbf{c}$ would not be efficient, since $\mathtt{N}_M$ depends on $(\mu,\mathbf{c})$ and would have to be recomputed for each frame in the sequence. Nevertheless, an efficient solution can be obtained from (3) by least-squares, considering that $\delta\mu$ is known

$$\delta\mathbf{c} = \Lambda_B[\mathtt{M}\delta\mu + \mathscr{E}], \tag{9}$$

where $\Lambda_B = (\mathtt{B}^{\top}\mathtt{B})^{-1}\mathtt{B}^{\top}$ is also constant and can be precomputed off-line.

At first glance this result may seem similar to the one presented in [14], section 4.1, and in [10]. There are nevertheless three major differences: a) here model parameters are additively updated, whereas in [14] the update procedure is compositional; b) here subspace appearance parameters are incrementally estimated and additively updated ($\mathbf{c}_{t+1} = \delta\mathbf{c} + \mathbf{c}_t$) and, in consequence, $\mathscr{E}$ includes a $-\mathtt{B}\mathbf{c}_t$ term, whereas in [14], as well as in [10], there is no such term; c) here the derivatives of the subspace basis are part of the Jacobian, whereas in [14] and in [10] they are not. As described in [10], this implies that assumption $\nabla_{\mathbf{x}}[\mathtt{B}\mathbf{c}](\mathbf{x}) \approx 0$. This assumption is approximately true for a rigid face, but not for a face whose appearance changes. In the experiments conducted in section 4 we show that for our problem the procedure introduced in this section performs better than those in [10] and [14].

3.3 The algorithm

In the implementation of our algorithm we use a modular eigenspace [15]. This allows a more flexible, compact, accurate and better conditioned model of the regions of interest. We will consider that all the regions are part of the same object and hence that they share the same $\delta\mu$ but could have different appearance variations.

Let $\{\mathtt{B}_1,\cdots,\mathtt{B}_r\}$ be the set of subspace basis for all modules. Given the reconstructed image for all pixels in region j, $[\mathtt{B}_j\mathbf{c}_j](\mathbf{x})$, the Jacobian matrix of the modular appearance tracker can be written as:

$$\mathtt{M} = \begin{bmatrix} \mathtt{M}_{0,1}\Sigma_1(\mu_t,\mathbf{c}_1) \\ \vdots \\ \mathtt{M}_{0,r}\Sigma_r(\mu_t,\mathbf{c}_r) \end{bmatrix},$$

where $\mathtt{M}_{0,j}$ and Σ_j are the factorisation result for the Jacobian matrix corresponding to region j. Finally, the factored modular tracking algorithm is as follows:

- **Off-line:**
  1. For each region j do:
     a) Compute and store $\mathtt{M}_{0,j}$ using $\mathtt{B}_j$.
     b) Compute and store $\Lambda_{M2,j} = \mathtt{M}_{0,j}^{\top} \mathtt{N}_{B_j}$.
     c) Compute and store $\Lambda_{M1,j} = \Lambda_{M2,j} \mathtt{M}_{0,j}$.
     d) Compute and store $\Lambda_{B,j} = (\mathtt{B}_j^{\top} \mathtt{B}_j)^{-1} \mathtt{B}_j^{\top}$.

- **Online (one iteration):**
  1. For each region j do:
     a) Warp $\mathbf{I}(z, t+\delta t)$ to $\mathbf{I}(f(x,\mu_t), t+\delta t)$.
     b) Compute $\mathscr{E}_j = [\mathbf{I}(f(\mathbf{x},\mu_t), t+\delta t) - \mathbf{I}_d - \mathtt{B}_j \mathbf{c}_{j,t}]$.
     c) Compute $\Sigma_j(\mu_t, \mathbf{c}_{t,j})$.
     d) Compute $\mathtt{H}_j = \Sigma(\mu_t, \mathbf{c}_{t,j})^{\top} \Lambda_{M1,j} \Sigma(\mu_t, \mathbf{c}_{t,j})$.
     e) Compute $\mathtt{A}_j = \Sigma(\mu_t, \mathbf{c}_{t,j})^{\top} \Lambda_{M2,j} \mathscr{E}_j$.
  2. Compute $\mathtt{H} = \sum_{j=1}^{r} \mathtt{H}_j$.
  3. Compute $\mathtt{A} = \sum_{j=1}^{r} \mathtt{A}_j$.
  4. Compute $\delta\mu = -\mathtt{H}^{-1}\mathtt{A}$.
  5. Update $\mu_{t+\delta t} = \mu_t + \delta\mu$.
  6. For each region j do:
     a) Compute $\delta \mathbf{c}_{j,t+\delta t} = \Lambda_{B,j} [\mathtt{M}_{0,j} \Sigma(\mu_t, \mathbf{c}_{t,j}) \delta\mu + \mathscr{E}_j]$.
     b) Update $\mathbf{c}_{j,t+\delta t} = \mathbf{c}_{j,t} + \delta \mathbf{c}_{j,t+\delta t}$.

4 Experiments

In this section we will show some experiments that validate the model and the fitting algorithm introduced in the paper. We will use an RTS (rotation, translation and scale) motion model, so $\mu = (\theta, t_u, t_v, s)$, and $f(\mathbf{x},\mu) = s\mathtt{R}(\theta)\mathbf{x} + \mathbf{t}$, where $\mathbf{x} = (u,v)^{\top}$, $\mathbf{t} = (t_u, t_v)^{\top}$ and $\mathtt{R}(\theta)$ is a 2D rotation matrix. In this case the factorisation in (7) results in

$$\Gamma(\mathbf{x}_i) = \begin{bmatrix} \mathtt{I}_{2l\times 2l}, & \begin{bmatrix} -v_i \mathtt{I}_{l\times l} & u_i \mathtt{I}_{l\times l} \\ u_i \mathtt{I}_{l\times l} & v_i \mathtt{I}_{l\times l} \end{bmatrix} \end{bmatrix}, \quad \Sigma(\mathbf{c},\mu) = \begin{bmatrix} \mathtt{C}\frac{1}{s}\mathtt{R}(-\theta) & 0 \\ 0 & \mathtt{C}\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{s} \end{bmatrix} \end{bmatrix},$$

where $\mathtt{I}_{d\times d}$ is the $d \times d$ identity matrix, $\mathtt{C}$ is a matrix storing $\mathbf{c}$ and $l = k+1$, being k the dimension of the linear subspace. For this model $\mathtt{M}_0$ and Σ have dimensions $N \times 4l$ and $4l \times 4$ respectively.

4.1 Model building.

One of the advantages of the appearance model introduced in section 2 is that deformation and illumination subspaces are independent, and so, they can be independently trained. This allows us to simplify the training process. We do not need image sequences with all facial expressions under all possible illumination conditions. Now, each subspace is trained with one video sequence. For the illumination subspace we use a sequence in which a light orbits in front of the target face with a neutral expression. For the deformation subspace we use a sequence captured with a non-saturating frontal illumination

in which the target face performs different facial expressions. The face is located and aligned in the first frame of both sequences, then, with a procedure similar to the one described in [11], both sequences are independently tracked and both linear subspace models independently built (see Fig. 2).

Figure 2: Some images used to build the deformation (first four images) and illumination (last four images) subspaces.

4.2 Tracking experiments.

In the first experiment we compare the performance of our model fitting procedure with the algorithms introduced in [10] and [14]. To this end we generate synthetic image sequences with a graphical face model and use the RMS of the frame reconstruction error, $I(f(\mathbf{x},\mu_{t+\delta t}),t+\delta t)-[\mathrm{B}\mathbf{c}_{t+\delta t}](\mathbf{x})$, as performance index. We have rendered three test sequences, SM (sequence with rigid head motion), SME (sequence with the same motion as in SM and some facial expressions) and SMIE (sequence with the same rigid motion and facial expressions as in SME but with a light orbiting around the face). For model training we have rendered two additional sequences with different expressions and similar illumination variations to the testing ones. The resultant illumination subspace has dimension 5 for the mouth region (45×45 pixels), 5 for the left eye (30×35 pixels) and 5 for the right eye area (30×35 pixels). The deformation subspace has dimension 7 for the mouth area and 10 for each eye region. When tracking the SM sequence all three approaches have equal performance (see Fig.3 left). When facial expressions are strong enough, the assumption $\nabla_{\mathbf{x}}[\mathrm{B}\mathbf{c}](\mathbf{x})\approx 0$ becomes invalid and the performance of our tracker is better. This is shown in the SME sequence (Fig.3 centre) between frames 300 and 400 in which strong eye and eyelid expressions are performed. When motion, facial expressions and strong illumination changes are combined, the only tracker that performs correctly is the one introduced in this paper (see Fig. 3 right).

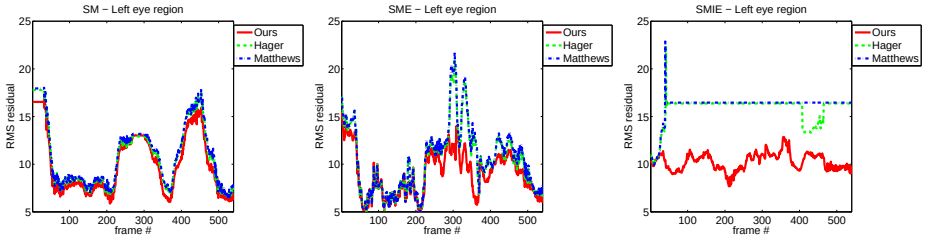


Figure 3: Comparison of Hager & Belhumeur [10], Matthews & Baker [14] and our model fitting procedure.

In the second experiment we use real image sequences acquired with an Apple iSight camera. This time the illumination subspace has dimension 5 for the mouth region (23×23 pixels), 5 for the left eye (15×18 pixels) and 5 for the right eye area (15×18

pixels). The deformation subspace has dimension 18 for the mouth area and 8 and 9 for the eye regions. With this model our tracker runs at 40 fps (including the reading of images from disk and displaying the results on screen) with an unoptimised C++ implementation on a Pentium-M Sonoma 1.83GHz. In Fig. 4 we show the tracking results for a test sequence in which the face performs non-rigid motion with drastic illumination changes. This test sequence is different from the one used for training (in Fig. 2 are shown some sample images from the training sequence). The illumination conditions are modified by moving a light source in front of the user, with fluorescent roof lights on. The estimated position of the face is overlayed in red (a rectangle for each module tracked). To the right side of each result image we show four smaller images: the rectified images of the three regions used in tracking ($I(f(\mathbf{x},\mu_t),t+\delta t)$) on the left-upper side, the reconstructed images ($I_{d,j}(\mathbf{x})+[\mathtt{B}_{i,j}\mathbf{c}_{i,t,j}](\mathbf{x})+[\mathtt{B}_{d,j}\mathbf{c}_{d,t,j}](\mathbf{x})$) on the right-upper side, the illumination reconstructed images ($I_{d,j}(\mathbf{x})+[\mathtt{B}_{i,j}\mathbf{c}_{i,t,j}](\mathbf{x})$) on the left-lower side and the deformation reconstructed images ($I_{d,j}(\mathbf{x})+[\mathtt{B}_{d,j}\mathbf{c}_{d,t,j}](\mathbf{x})$) on the right-lower side. The tracker is locked on the face trough all the 966 frames of the experiment.

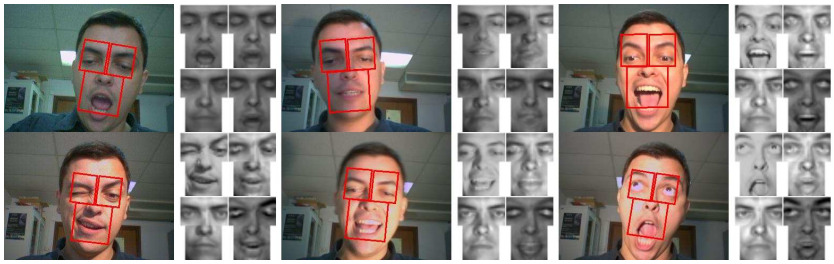
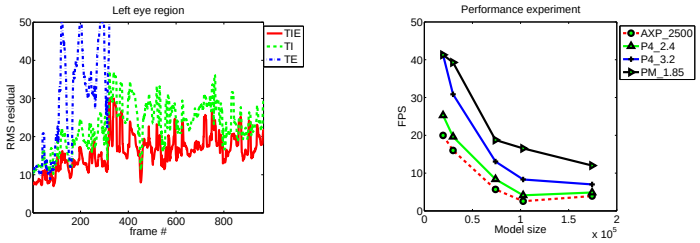


Figure 4: Real tracking experiment .

The rectified images give us an idea of how robust the tracker is to the changes in the appearance throughout the sequence. The performance in terms of robustness is almost perfect. The images reconstructed with the illumination and deformation models inform us about how well each source of appearance is separated during tracking. Here again the performance is remarkable, given that the illumination subspace accurately estimates the changes in the illumination of the scene and the deformation subspace represents the facial expression. Occasionally, the images reconstructed with the deformation model show "ghost" expressions. These are caused by facial expressions not present in the training sequence, because we are approximating with a linear subspace the manifold of facial expressions which is non-linear. Finally, the image reconstructed with both models gives us information on how good our model reconstructs the target image. Here, again, the reconstruction is good, except for those expressions not present in the training sequence.

In the third experiment we compare the accuracy (RMS residual) when tracking the sequence shown in Fig. 4 with and without each of the subspaces in the appearance model. With this experiment we will establish the contribution of each subspace to the tracking process. In Fig. 5 (a) we show the RMS residual for the left eye region. The tracker using deformations and illumination subspaces (TIE) performs consistently better than those including only illumination (TI) or deformations (TE). Since the appearance changes caused by the illumination are more significant than those due to facial expressions the performance of the TE tracker is much worse than the others and eventually looses track. Al-

though the illumination subspace alone can successfully track the sequence, it can not explain all the appearance variations. That is why its residual is higher than the one obtained with the whole appearance model (TIE).



(a) (b)

Figure 5: (a) Contribution of each subspace to the tracking process. (b) Algorithm performance.

In the last experiment we evaluate the performance of the tracker in a real sequence with models of different size and in various personal computers. Best performances are obtained when all data structures used in the algorithm fit in the cache memory of the processor. As shown in Fig. 5 (b), the performance quickly degrades as the model size increases. The *model size* is the number of pixels per region times the number of basis per region times the number of regions. We have tested our algorithm on an Athlon XP 2500+ (AXP_2005), Pentium 4 2.4 GHz (P4_2.4), Pentium 4 3.2 GHz (P4_3.2) and on a Pentium-M Sonoma 1.86 GHz (PM_1.85).

5 Conclusions

An important issue in facial expression analysis is developing easy-to-train, efficient and robust tracking algorithms, which can factor some of the various sources of appearance variation in the face. In this paper we have introduced a linear subspace representation of facial appearance which separates facial expressions from illumination variations. The appearance of a face is represented by the addition of two independent linear subspaces, one modelling the facial expressions and the other modelling the illumination. To our knowledge, this model is new. In the context of 3D and 2D rigid face tracking invariant to illumination changes, LaCascia [12] (3D face model) and Hager [10] (2D face model) had also used a single linear subspace to model changes in illumination. In this paper we have shown that an independent illumination and deformation subspace may also be used for fitting an eigenface (i.e. a non-rigid 2D appearance-based model of a face). Vasilescu and Terzopoulos [18] also used a linear model to represent variations in the illumination and appearance of an eigenface, but in their multi-linear tensor model illumination and appearance are not independent. In this paper we have shown that they can be assumed to be approximately independent. This assumption notably simplifies the training of the model, which can be made with no manual intervention. We have also introduced an efficient model fitting algorithm, which is able to track a deforming face in real-time and which performs better than the two other most prominent efficient tracking algorithms.

The ideas presented in this paper could also be applied to other areas of interest in computer vision. For example, if in our training data we exchange facial expression for

identity, the tracker could also be used for video-based recognition.

Acknowledgements

The authors gratefully acknowledge funding from the Spanish *Ministerio de Educación y Ciencia*, under contract TRA2005-08529-C02-02.

References

[1] R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *Trans. on PAMI*, 25(2):218–233, February 2003.

[2] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *IJCV*, 26(1):63–84, 1998.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proc. of SIGGRAPH*, pages 187–194, 1999.

[4] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *Trans. on PAMI*, 25(9):1–12, 2003.

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *Proc. of ECCV*. 1998.

[6] F. de la Torre and M.J. Black. Robust parameterized component analysis: Applications to 2d facial modeling. In *Proc. of ECCV (4)*, pages 653–669. 2002.

[7] H. Fei and I. Reid. Joint bayes filter: A hybrid tracker for non-rigid hand motion recognition. In *Proc. of ECCV*, pages 497–508, 2004.

[8] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Trans. on PAMI*, 23(6):643–660, 2001.

[9] D.B. Grimes, A.P. Shon, and R.P.N. Rao. Probabilistic bilinear models for appearance-based vision. In *Proc. of ICCV*. 2003.

[10] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *Trans. on PAMI*, 20(10):1025–1039, 1998.

[11] L. Jongwoo, D. Ross, L. Ruei-Sung, and Y. Ming-Hsuan. Incremental learning for visual tracking. In *Advances in Neural Information Processing Systems*, 2004.

[12] M. LaCascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *Trans. on PAMI*, 22(4), 2000.

[13] K. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In *Proc. of CVPR*, 2005.

[14] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[15] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces. In *Proc. of CVPR*, pages 84–91, 1994.

[16] J.B. Tenenbaum and W.T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12:1247–1283, 2000.

[17] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. of ICCV*, 2001.

[18] M. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proc. of ECCV*, 2002.