Robust parameterized component analysis: theory and applications to 2D facial appearance models

Fernando De la Torre[a],[*] and Michael J. Black[b]

[a]*Department of Communications and Signal Theory, La Salle School of Engineering, Universitat Ramon LLull, Barcelona 08022 Spain*
[b]*Department of Computer Science, Brown University, Box 1910, Providence, RI 02912, USA*

Abstract

Principal component analysis (PCA) has been successfully applied to construct linear models of shape, graylevel, and motion in images. In particular, PCA has been widely used to model the variation in the appearance of people's faces. We extend previous work on facial modeling for tracking faces in video sequences as they undergo significant changes due to facial expressions. Here we consider person-specific facial appearance models (PSFAM), which use modular PCA to model complex intra-person appearance changes. Such models require aligned visual training data; in previous work, this has involved a time consuming and error-prone hand alignment and cropping process. Instead, the main contribution of this paper is to introduce *parameterized component analysis* to learn a subspace that is invariant to affine (or higher order) geometric transformations. The automatic learning of a PSFAM given a training image sequence is posed as a continuous optimization problem and is solved with a mixture of stochastic and deterministic techniques achieving sub-pixel accuracy. We illustrate the use of the 2D PSFAM model with preliminary experiments relevant to applications including video-conferencing and avatar animation.
© 2003 Elsevier Inc. All rights reserved.

Keywords: Facial appearance models; Principal component analysis; Robust statistics; Eigen-registration; Facial analysis

---

[*] Corresponding author.
E-mail addresses: ftorre@salleURL.edu (F. De la Torre), black@cs.brown.edu (M.J. Black).
URLs: http://www.salleURL.edu/~ftorre/, http://www.cs.brown.edu/~black/.

1. Introduction

This paper addresses the problem of learning a linear subspace representation of a training set in which the data (e.g., images) may have undergone some unknown parametric transformation (e.g., affine). The key idea is to simultaneously solve for the optimal linear subspace representing the data while aligning the training data with that subspace. To illustrate the method we develop it in the context of face modeling. In particular, we adopt the idea of modular eigenspaces (ME) [31,40,44] and apply our *parameterized component analysis* technique to the problem of developing person-specific facial appearance models (PSFAM).

Consider the problem of learning a linear subspace representing the variation of the subject's right eye in Fig. 1. The images were captured by asking the user to change the configuration of the eyes (open, close, look right, etc.) while holding the head still. However, it is not reasonable to assume that the person is absolutely still during the training time, and in practical situations there are always small motions between frames. Observe that in this kind of sequence it is difficult to gather aligned data due to person's motion and the lack of labeled points for solving the correspondence problem between frames.

Although many computer vision researchers have used principal component analysis (PCA) to model the face [11,17–19,26,40,41,52] the major drawback of this traditional technique is that it requires normalized (aligned) samples in the training data. While, in the recognition process, alignment of the data with respect to the face model is a common step as noted by Martínez [37], little work has addressed problems posed by misalignment at the learning stage. Previous methods for constructing appearance models [11,18,19,26,40,41] have cropped the region of interest by hand, or have used a hand-labeled, pre-defined, feature points to compute the translation, scaling and rotation that brought each image into alignment with a prototype. However, this way of collecting data is likely to introduce errors due to inaccuracies which arise from labeling the points by hand, even with the use of landmarks, since it is difficult to achieve sub-pixel accuracy. In addition, manual cropping is a tedious, unpleasant, and time consuming task.

The aim of the paper is illustrated in Fig. 2, where Fig. 2a shows some original images used for training. From these (non-aligned images) we compute a set of linear bases using PCA in the standard way. Fig. 2b shows the original images reconstructed using the non-aligned bases. Fig. 2c shows the reconstructed images obtained using the *parameterized component analysis* technique presented here.

Fig. 1. Some frames from a training sequence of images.

Fig. 2. Reconstruction of image data using an eigenspace representation: (a) example frames from the training data; (b) reconstruction of the right eye without any alignment; (c) reconstruction of the right eye with the proposed method (eigen-registration).

This "eigen-registration" technique iteratively computes the subspace while aligning the training images w.r.t. this subspace. That is, the algorithm that we propose in this paper will simultaneously learn the local appearance basis, creating an eigenspace while computing the motion to align the images w.r.t. the eigenspace. In the case of modular eigenspaces (ME) [31,40,44] considered here, masks which define the spatial domain of the ME are defined by hand in the first frame (no appearance model is previously learned) and after that the method is fully automatic. Preliminary results were presented in [13].

2. Previous work

It is beyond the scope of this paper to review all possible applications of PCA and subspace methods, therefore we just briefly describe the theory and point to related work for further information.

2.1. Subspace learning

Let $\mathbf{D} = [\mathbf{d}_1 \ \mathbf{d}_2 \ \cdots \ \mathbf{d}_T] = [\mathbf{d}^1 \ \mathbf{d}^2 \ \cdots \ \mathbf{d}^d]^{\mathrm{T}}$ be a matrix $\mathbf{D} \in \Re^{d \times T}$, where each column $\mathbf{d}_i$ is a data sample, T is the number of training images, and d is the number of pixels in each image. If the effective rank of $\mathbf{D}$ is much less than d, we can approximate the column space of $\mathbf{D}$ with $k \ll d$ principal components. Let the first k principal

components of $\mathbf{D}$ be $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \Re^{d \times k}$. The columns of $\mathbf{B}$ span the subspace of maximum variation of $\mathbf{D}$.[1]

Although a closed form solution for computing the principal components ($\mathbf{B}$) can be achieved by finding the k largest eigenvectors of the covariance matrix $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ [20], here it is useful to exploit work that formulates PCA/subspace learning as the minimization of an energy function [16,20,21]

$$E_{\mathrm{pca}}(\mathbf{B}, \mathbf{C}) = \|\mathbf{D} - \mathbf{B}\mathbf{C}\|_{\mathrm{F}}^2 = \sum_{i=1}^{T} \|\mathbf{d}_i - \mathbf{B}\mathbf{c}_i\|_2^2 = \sum_{t=1}^{T} \sum_{p=1}^{d} \left(d_{pt}. - \sum_{j=1}^{k} b_{pj} c_{jt} \right)^2,$$

where $\mathbf{C} = [\mathbf{c}_1 \; \mathbf{c}_2 \; \cdots \; \mathbf{c}_n]$ and each $\mathbf{c}_i$ is a vector of coefficients used to reconstruct the data vector $\mathbf{d}_i$. Observe that subspace learning involves approximately factoring the data, $\mathbf{D}$, into the product of the bases, $\mathbf{B}$, and the coefficients, $\mathbf{C}$, therefore it can be posed as a bilinear estimation problem. There exist many methods for minimizing this equation including alternated least squares (ALS), criss-cross regression, variants of expectation-maximization (EM), etc., but in the case of PCA, they share the same basic philosophy. These algorithms alternate between solving for the coefficients $\mathbf{C}$ with the appearance bases $\mathbf{B}$ fixed and then solving for the bases $\mathbf{B}$ with $\mathbf{C}$ fixed. Typically, both updates are computed by solving a linear system of equations.

2.2. Adding motion into the subspace formulation

Since the preliminary work of Sirovich and Kirby [48] and the successful eigenface application of Turk and Pentland [49], PCA has been widely applied to the construction of a face subspace. Since then, there has been a lot of work and interest in trying to construct more accurate models of the high dimensional manifold of faces. During the last few years there has been a growing trend to apply new machine learning or multivariate statistical techniques to construct more accurate face models. Many 2D/3D linear/non-linear models [26,39,46,52] have been proposed based on support vector machines, mixture of factor analyzers, Independent Component Analysis, Kernel PCA, etc. See [12,26,52] for an extended review in the context of recognition and modeling.

Mis-registration or variations in scale introduce significant non-linearities in the manifold of faces and can reduce the accuracy of tracking and recognition algorithms. While previous approaches have dealt with these issues as a separate, off-line registration processes (often manual), here it is integrated into the learning procedure.

---

[1] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the jth column of $\mathbf{D}$ and $\mathbf{d}^j$ is a column vector representing the jth *row* of $\mathbf{D}$. d_{ij} denotes the scalar in row i and column j of $\mathbf{D}$ and the scalar ith element of a column vector $\mathbf{d}_j$. All non-bold letters represent scalar variables. d_{ji} is the ith scalar element of the vector $\mathbf{d}^j$. diag is an operator that transforms a vector to a diagonal matrix, or a matrix into a column vector by taking each of its diagonal components. $\mathrm{tr}(\mathbf{D})$ is the trace operator. $\|\mathbf{d}\|_2^2 = \mathbf{d}^{\mathrm{T}}\mathbf{d}$ denotes the L_2 norm and $\|\mathbf{d}\|_{\mathrm{W}}^2 = \mathbf{d}^{\mathrm{T}}\mathbf{W}\mathbf{d}$ is the weighted L_2 norm. $\|\mathbf{D}\|_{\mathrm{F}}^2 = \mathrm{tr}(\mathbf{D}^{\mathrm{T}}\mathbf{D}) = \mathrm{tr}(\mathbf{D}\mathbf{D}^{\mathrm{T}})$ is the Frobenius norm of $\mathbf{D}$. $\mathbf{D}_1 \circ \mathbf{D}_2$ denotes the Hadamard (point wise) product.

Recently, there has been an interest in the simultaneous computation of appearance bases and the motion that aligns the training images. This is a classic chicken-and-egg problem. Once the correspondence of *interesting* points through an image sequence is known, learning the appearance model is straightforward, and if the appearance is known solving for the correspondence is easy. De la Torre et al. [17] proposed a method for face tracking which recovers affine parameters using subspace methods. This method dynamically updates the eigenspace by utilizing the most recent history. The updating algorithm estimates the parametric transformation, which aligns the actual image w.r.t. the eigenspace and recalculates a local eigenspace. Because the new images usually contain information not available in the eigenspace, the motion parameters are calculated in a robust manner. However, the method assumes that an initial eigenspace is learned from a training set aligned by hand.

Schweitzer [47] proposed a deterministic method which registers a set of images with respect to their eigenfeatures, applying it to the *flower garden* sequence for indexing purposes. However, the assumption of affine or quadratic motion models [47] is only valid when the scene is planar. The extension to the general case of arbitrary 3D scenes and camera motions remains unclear. As Schweitzer notices [47] the algorithm is likely to get stuck in local minima, since it comes from a linearization and uses gradient descent methods. Alternatively, Rao [45] proposed a neural-network which can learn a translation-invariant code for natural images. Although he suggests updating the appearance basis, the experiments show only translation-invariant recognition, as proposed by Black and Jepson [4]. Frey and Jojic [24] took a different approach and they introduce an expectation maximization (EM) algorithm for factor analysis (similar to PCA) that is invariant to geometric transformations. The proposed method is problematic because the computational cost grows exponentially with the number of possible spatial transformations, and can be too computationally intensive when working with realistic high dimensional (greater than two) motion models.

Using a different approach Mandel and Penev [36] report the interesting observation that non-properly aligned data lie on curved manifolds. This observation forms the basis of an algorithm to align visual data. Results where reported on image sequences of faces to compensate for translational motion. However, it is not clear how to extend the method to more complex high dimensional motion models without considerably increasing the computational cost. In this paper, unlike previous methods we use stochastic and multi-resolution techniques to avoid local minima in the minimization process. Also, we extend previous approaches to multiple regions within a robust (to outliers) and continuous optimization framework.

In a different direction, there has been intensive research on automatically or semi-automatically building facial shape models using extracted landmarks. Most of the previous work in this area assumes that the object has already been segmented from the image sequence and in some cases the features or curves are placed by hand. If this is the case, the problem is how to put the features in correspondence using rigid or non-rigid transformations [9,29]. In the other direction, Walker et al. [32] have proposed a method for automatically placing landmarks to define correspondence between images and hence automatically constructing appearance models.

See the report of Cootes and Taylor [12] for a good review in automatic 2D/3D land-mark placement. In contrast to previous automatic landmark methods, we use pa-rameterized matching with a low dimensional model (e.g. affine) and generalize the matching by incorporating a subspace for the appearance variation.

2.3. Person specific models

While most work on face tracking focuses on generic trackers which are indepen-dent of the identity of the person being tracked [5,8,10,11,26,27,34], here we focus on PSFAM [17,26,22,50] for tracking a single individual and use PCA to model the variations due to changes in expression. Although PSFAM are only valid for one person, they remain useful in many vision related applications such as vision-based human–computer interaction [5,8,10,11,17–19,26,27,31], driver fatigue detec-tion, facial animation, face detection/recognition, video-conferencing, text to speech, etc., which usually involve tracking or modeling a particular user.

We build these PSFAMs using modular eigenspaces (ME) [31,40] which have benefits over global eigenspace methods (e.g., more accurate reconstruction of the regions of interest, lower computational cost, robustness to occlusions [37], etc.). However, it is worth pointing out that representations other than ME have been ex-plored successfully for face recognition and tracking; for instance, Local Feature Analysis [42,43] or Gabor jets with elastic graph matching [51]. Although these tech-niques have shown good performance in recognition and tracking domains, they do not address the issue of learning a model invariant to geometric transformations.

3. Generative model for 2D faces

The generative model we propose for image formation takes into account the mo-tion and appearance of the face. Adopting the ME approach we use predefined masks for the various image features and learn the appearance bases within these re-gions. Fig. 3 shows some frames of a training set for learning a 2D PSFAM. Given this training data as input, the algorithm that we propose in this paper is able to fac-tor the training data into appearance and motion of the predefined face regions. In principle the regions of support (masks) could be computed as an eigenspace-based segmentation problem (finding independent regions). However, in the case of the face, these regions are quite clear, and a rough approximation is sufficient. There-fore, we define the masks in the first image and they will remain the same for the entire training image sequence.

Let $\mathbf{d}_t \in \Re^{d \times 1}$ be the region of d pixels belonging to the face, defined by hand in the first image. $\boldsymbol{\pi}^l = [\pi_1^l \ \pi_2^l \ \cdots \ \pi_d^l]^{\mathrm{T}} \in \Re^{d \times 1}$ denotes the binary mask for the region l and it has the same size as the face region (d pixels). Each of the mask's pixels take a binary value, $\pi_p^l \in \{0, 1\}$ and there is no overlap between masks, that is, $\sum_{l=1}^{L} \pi_p^l = 1 \ \forall p$. $\boldsymbol{\pi}^l$ will contain d_l pixels with value 1, which define the spatial domain of the mask l (see Fig. 3) and $\sum_{l=1}^{L} d_l = d$.
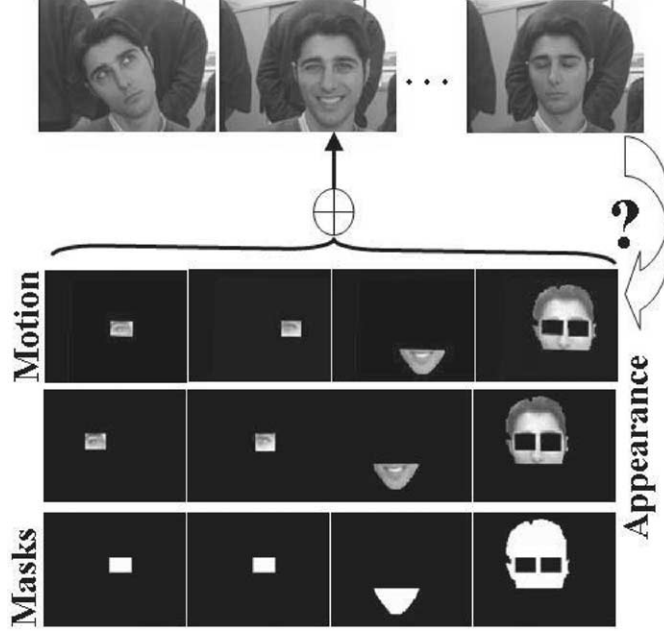
Fig. 3. The generative model for an image sequence. Face images are decomposed using appearance models within regions corresponding to the eyes, mouth, and remainder of the face. The appearance within regions varies independently. In the current implementation the regions move together according to a single affine (or other parameterized) model.

Each of these masks will have an associated eigenspace. Let $\mathbf{d}_t^l \in \Re^{d_l \times 1}$ be the image patch of the region l and let $\mathbf{c}_t^l$ be the appearance coefficients of the region l at time t. $\mathbf{B}^l = [\mathbf{b}_1^l\ \mathbf{b}_2^l\ \cdots\ \mathbf{b}_{k_l}^l] \in \Re^{d_l \times k_l}$ are the k_l appearance bases for the lth region. $\tilde{\mathbf{B}}^l \in \Re^{d \times k_l}$ (which is introduced for notational convenience) will be equal to $\mathbf{B}^l$ for all pixels where $\pi_p^l = 1$ (i.e., belongs to the l mask) and otherwise can take an arbitrary value. The graylevel of the patch, or region l, will be reconstructed by a linear combination of an appearance basis $\tilde{\mathbf{B}}^l$, as

$$\mathbf{d}_t = \begin{bmatrix} \mathbf{d}_t^1 \\ \vdots \\ \mathbf{d}_t^L \end{bmatrix} = \begin{bmatrix} \mathbf{B}^1 \mathbf{c}_t^1 \\ \vdots \\ \mathbf{B}^L \mathbf{c}_t^L \end{bmatrix} = \sum_{i=1}^{L} (\pi^l \circ \tilde{\mathbf{B}}^l \mathbf{c}_t^l). \tag{1}$$

3.1. Motion

If the face to be modeled can be considered to be far away from the camera, it can be approximated by a plane. The motion of planar surfaces, under orthographic or perspective projection, can be recovered with a parametric model of 6 or 8 parameters [5]. For simplicity, the rigid motion of the face will be parameterized by an affine model: $\mathbf{f}_1(\mathbf{x}_p, \mathbf{a}_t^l) = [a_{1t}^l\ a_{4t}^l]^{\mathrm{T}} + \mathbf{A}_f^l [x_p - x_c^l\ y_p - y_c^l]^{\mathrm{T}}$, where $\mathbf{A}_f^l$ is a

matrix containing the affine parameters $(a_{2t}^l \ a_{3t}^l \ a_{5t}^l \ a_{6t}^l)$. Let $\mathbf{a}_t^l = [a_{1t}^l \ a_{2t}^l \ \cdots \ a_{6t}^l]^{\mathrm{T}}$ denote the vector of affine motion parameters of the mask l at time t and let $\mathbf{x}_p = [x_p \ y_p]^{\mathrm{T}}$ denote the Cartesian coordinates of the image at the pixel p and $\mathbf{x}_c^l = [x_c^l \ y_c^l]^{\mathrm{T}}$ denote the center of the lth region. Throughout the paper, we will assume that the rigid motion of all the modular eigenspaces (w.r.t. the center) is the same (i.e., $\mathbf{a}_t^1 = \mathbf{a}_t^2 \cdots = \mathbf{a}_t^L$).

Once the appearance and motion models have been defined, the graylevel of each pixel of the image $\mathbf{d}_t$ is explained as a superposition of a region-subspace plus a warping, see Fig. (3); that is, $\mathbf{d}_t = \sum_{l=1}^{L} (\boldsymbol{\pi}^l \circ \tilde{\mathbf{B}}^l \mathbf{c}_t^l)(\mathbf{f}_1(\mathbf{x}, \mathbf{a}_t^l))$ where $\mathbf{x} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_d]^{\mathrm{T}}$ and the notation $(\boldsymbol{\pi}^l \circ \tilde{\mathbf{B}}^l \mathbf{c}_t^l)(\mathbf{f}_1(\mathbf{x}, \mathbf{a}_t^l))$ means that the reconstructed image region $(\boldsymbol{\pi}^l \circ \tilde{\mathbf{B}}^l \mathbf{c}_t^l)$ is warped by the motion $(\mathbf{f}_1(\mathbf{x}, \mathbf{a}_t^l))$. Observe that this image model is essentially the same as previous appearance representations [4,11,18] but with the addition of modular eigenspaces and we now treat the basis as parameters to be estimated.

4. Learning the model parameters

Once the model has been defined, in order to automatically learn the PSFAM, it is necessary to learn the model parameters. In this section, we describe the learning procedure; that is, given an observed image sequence $(\mathbf{D} \in \mathfrak{R}^{d \times T})$ and L masks in the first image $(\boldsymbol{\pi} = \{\boldsymbol{\pi}^1, \ldots, \boldsymbol{\pi}^L\})$, we find the parameters $\mathcal{B}, \mathcal{C}, \mathcal{A}$, and $\boldsymbol{\sigma}$, that best reconstruct the sequence (in a robust statistical sense). Where $\mathcal{A} = \{\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^L\}$ is the set of motion parameters of all the face regions in all the image frames. $\mathbf{A}^i = [\mathbf{a}_1^i \ \mathbf{a}_2^i \ \cdots \ \mathbf{a}_T^i]$ is the matrix which contains the motion parameters for each image in the ith region. Analogously, $\mathcal{C} = \{\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^L\}$, where $\mathbf{C}^i = [\mathbf{c}_1^i \ \mathbf{c}_2^i \ \cdots \ \mathbf{c}_T^i]$ and $\mathcal{B} = \{\mathbf{B}^1, \mathbf{B}^2, \ldots, \mathbf{B}^L\}$.

At this point, learning the model parameters can be posed as a minimization problem. In this case the residual will be the difference between the image at time t and the reconstruction using the model. In order to take into account outlying data, we introduce a robust objective function, minimizing E_{rereg}:

$$E_{\mathrm{rereg}}(\mathcal{B}, \mathcal{C}, \mathcal{A}, \boldsymbol{\sigma}) = \sum_{t=1}^{T} \sum_{p=1}^{d} \rho \left(d_{pt} - \sum_{l=1}^{L} \left(\pi_p^l \sum_{j=1}^{k} b_{pj}^l c_{jt}^l \right) (\mathbf{f}_1(\mathbf{x}_p, \mathbf{a}_t^l)), \sigma_p \right), \qquad (2)$$

where b_{pj}^l is the pth pixel of the jth basis of $\mathbf{B}^l$ for the region l. Observe that the pixel residual is *filtered* by the Geman–McClure robust error function [25] given by $\rho(x, \sigma_p) = x^2/(x^2 + \sigma_p^2)$, in order to reduce the influence of outlying data. σ_p is a parameter that controls the convexity of the robust function and is used for deterministic annealing [4,7]. Benefits of the robust formulation for subspace related problems are explained elsewhere [15,16]. Observe that the previous equation is a *patched* version of *Eigentracking* [4], and similar to AAM [11] or *Flexible Eigentracking* [18] without shape constraints. However, in contrast to these approaches [4,11,18], in E_{rereg} the appearance bases $\mathcal{B}$ are now treated as a set of parameters to be estimated.

4.1. Stochastic state initialization

The error function E_{rereg}, Eq. (2), is a non-convex function, thus, without a good starting point, any gradient descent method may get trapped in local minima. When computing the motion parameters, as in the case of optical flow, a coarse-to-fine strategy [4,12], in which the input images are represented by a Gaussian pyramid, can help avoid local minima. Although a coarse-to-fine strategy is helpful, this technique is insufficient in our case, since in real image sequences the size of the face can be small in comparison to the number of pixels in the background, and large motions can be performed (e.g., in the sequences that we tried, the face can move more than 20 pixels from frame to frame). In order to cope with such real conditions, we explore the use of stochastic methods such as simulated annealing (SA), genetic algorithms (GA) [38] or Condensation (particle filtering) [6] for motion estimation. Betke and Makris [3] have used a fast version of SA to match traffic signals over rigid parameters, Lanitis et al. [33] made use of GA to fit an active shape model (ASM). De la Torre et al. [19] applied particle filtering [6] for appearance based tracking of rigid and non-rigid motion. The use of particle filtering allows switching between models (e.g., models with different spatial support [19]), coping with large motion changes and avoiding local minima in the parameter estimation process. Although the techniques are very similar computationally speaking, here we make use of GA [38] within a coarse-to-fine strategy.

Given the first image of the sequence we manually initialize the masks at the highest resolution level and assign the graylevel image values to the first basis for each region $\mathcal{B} = \{\mathbf{b}_1^1, \ldots, \mathbf{b}_1^L\}$. Afterwards, we take the subset of the m frames closest in time (typically $m = 15$), and use a GA for a first estimation of the motion parameters which minimize Eq. (2). For the initial estimation of the motion parameters with the GA, we use a least squares version of Eq. (2); that is, $\rho(x) = x^2$. Given the genetic estimation of these parameters, we recompute the bases $\mathcal{B}$ which preserve 60% of the energy. This initialization procedure is repeated until all the frames in the image sequence are initialized. The procedure is summarized as

- Manual initialization in the first frame.
  - Initialize the mask in the image $\mathbf{d}_1$.
  - Initialize the bases $\mathcal{B} = \{\mathbf{b}_1^1, \ldots, \mathbf{b}_1^L\}$ with the graylevel values of $\mathbf{d}_1$.
- Stochastic initialization of the motion and appearance parameters for $\mathbf{D}$.
  - for $i = 2\!:\!m\!:\!T$ (Matlab notation)

    Run the GA for computing the motion and appearance parameters in $\{\mathbf{d}_i, \ldots, \mathbf{d}_{i+m}\}$.

    Perform SVD on the registered set of images from 1 to m and keep the number of bases which preserve 60% of the energy.

    Update the set of bases $\mathcal{B}$.
  - end

The GA uses 300 individuals over 13 generations for each frame. The selection function we use is the normalized geometric ranking, which defines the probability of one individual as $P_i = (q/(1 - (1 - q)^P))(1 - q)^{(r-1)}$, where q is the probability of selecting the best individual, r is the rank of the individual, and P the population

size. See [38] for a more detailed explanation of the GA. At the beginning, q has a low value, and it is successively increased over generations acting as a temperature parameter in the deterministic annealing [4,7] for improving the local search. The crossover process is a convex combination between two samples, i.e., $\alpha * cromosome_1 + (1 - \alpha) * cromosome_2$ where $1 \geqslant \alpha \geqslant 0$. The genetic operator is a simple Gaussian random perturbation, which also depends on the temperature parameter. In our experiments we take $q = 0.04$ and $\alpha = 0.5$.

4.2. Robust deterministic learning

The previous section describes a method for computing an initial estimate of the parameters $\mathcal{B}, \mathcal{C}, \mathcal{A}$. In order to improve the solution and achieve sub-pixel accuracy, a normalized gradient descent algorithm for minimizing Eq. (2) has been employed in [13]. Alternatively (and conveniently) we can reformulate the minimization problem as one of iteratively reweighted least-squares (IRLS), which provides an approximate, iterative, solution to the robust M-estimation problem [30,35]. For a given $\boldsymbol{\sigma}$, a matrix $\mathbf{W} \in \Re^{d \times T}$, which contains the positive weights for each pixel and each image, is calculated for each iteration as a function of the previous residuals $e_{pi} = d_{pt} - (\pi_p^l \sum_{j=1}^{k} b_{pj}^l c_{jt}^l)(\mathbf{f}_1(\mathbf{x}_p, \mathbf{a}_t^l))$. Each element, w_{pi} (pth pixel of the ith image) of $\mathbf{W}$ will be equal to $w_{pi} = \psi(e_{pi}, \sigma_p)/e_{pi}$, where $\psi(e_{pi}, \sigma_p) = \partial \rho(e_{pi}, \sigma_p)/\partial e_{pi} = 2 e_{pi}\sigma_p^2/(e_{pi}^2 + \sigma_p^2)^2$, [28]. Given an initial error, the weight matrix $\mathbf{W}$ is computed and Eq. (2) becomes

$$E_{\text{wereg}}(\mathcal{B}, \mathcal{C}, \mathcal{A}, \boldsymbol{\sigma}) = \sum_{t=1}^{T} \left\| \mathbf{d}_t - \sum_{l=1}^{L} (\boldsymbol{\pi}^l \circ \tilde{\mathbf{B}}^l \mathbf{c}_t^l)(\mathbf{f}_1(\mathbf{x}, \mathbf{a}_t^l)) \right\|_{\mathbf{W}_t}^2 , \tag{3}$$

$$= \sum_{t=1}^{T} \sum_{l=1}^{L} \| \mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^l)) - \mathbf{B}^l \mathbf{c}_t^l \|_{\mathbf{W}_t^l}^2, \tag{4}$$

where $\mathbf{f}$ will warp the images towards the eigenspace, whereas $\mathbf{f}_1$ warps the bases towards the images. Observe that $\mathbf{f}$ will be approximately the inverse of $\mathbf{f}_1$. Recall that $\|\mathbf{d}\|_{\mathbf{W}}^2 = \mathbf{d}^T\mathbf{W}\mathbf{d}$ is a weighted norm. $\mathbf{W}_t \in \Re^{d \times d}$ is a diagonal matrix, such that the diagonal elements are the tth column of $\mathbf{W}$. $\mathbf{W}_t^l \in \Re^{d_l \times d_l}$ is a diagonal matrix, where the diagonal is created by the elements of the tth column of $\mathbf{W}$ which belong to the lth region. Observe that if $\mathbf{W}$ is a matrix with all ones we have the least-squares solution.

Eq. (4) provides the formulation for robust parameterized component analysis. Minimizing (4) with respect to the parameters gives a subspace that is invariant to the allowed geometric transformations and robust to outliers on a pixel level. Clearly, finding the minimum is a challenge and the process for doing so is described below.

Notice that, if the motion parameters are known, computing the basis and the coefficients translates into a weighted bilinear problem (computing basis $\mathcal{B}$ and coefficients $\mathcal{C}$). In order to compute the updates of the bases and coefficients in closed form in the simplest way, we use the following observation:

$$E_{\text{wereg}} = \sum_{t=1}^{T} \sum_{l=1}^{L} \|(\mathbf{d}_w^l)_t - \mathbf{B}^l \mathbf{c}_t^l\|_{\mathbf{W}_t^l}^2 = \sum_{p=1}^{d_l} \sum_{l=1}^{L} \|(\mathbf{d}_w^l)^p - (\mathbf{C}^l)^{\text{T}} (\mathbf{b}^l)^p\|_{(\mathbf{W}^l)^p}^2, \tag{5}$$

where $(\mathbf{d}_w^l)_t$ is the warped image $\mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t))$ and it is the tth column of the matrix $\mathbf{D}_w$ (just the d_l pixels of the lth region). Recall that $(\mathbf{d}_w^l)^p$ is a column vector which corresponds to the pth row of the matrix $\mathbf{D}_w$ and that $(\mathbf{W}^l)^p$ is a diagonal matrix which contains the pth row of the matrix $\mathbf{W}$ of the region l.

Minimizing Eq. (4) is a non-linear optimization problem w.r.t. the motion parameters. Following previous work on motion estimation [2,4,27], we linearize the variation of the function, using a first-order Taylor series approximation. Without loss of generality, rather than linearizing the transformation which warps the eigenspace towards the image $\mathbf{f}_1(\mathbf{x}, \mathbf{a}_t)$, we linearize the transformation which aligns the incoming image w.r.t. the eigenspace $\mathbf{f}(\mathbf{x}, \mathbf{a}_t)$ (see Eq. 4). Expanding, $\mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^{l0} + \Delta \mathbf{a}_t^l))$ in the Taylor series about the initial estimation of the motion parameters $\mathbf{a}_t^{l0}$ (which are given by the GA):

$$\mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^{l0} + \Delta \mathbf{a}_t^l)) = \mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^{l0})) + \mathbf{J}_t^l \Delta \mathbf{a}_t^l + \text{h.o.t.}, \tag{6}$$

where $\mathbf{J}_t^l$ is the Jacobian at time t of the lth region and h.o.t. denotes the higher order terms.

$$\mathbf{J}_t^l = \left[\frac{\partial \mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^{l0}))}{\partial a_{t1}^l} \quad \frac{\partial \mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^{l0}))}{\partial a_{t2}^l} \quad \cdots \quad \frac{\partial \mathbf{d}_t^l(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^{l0}))}{\partial a_{tm}^l} \right]$$

is computed as

$$\mathbf{J}_t^l = \begin{bmatrix} \nabla d_{1t}^{\text{T}}(\mathbf{f}(\mathbf{x}_1, \mathbf{a}_t^{l0})) \frac{\partial \mathbf{f}(\mathbf{x}_1, \mathbf{a}_t^{l0})}{\partial \mathbf{a}_t^l} \\ \vdots \\ \nabla d_{d_l t}^{\text{T}}(\mathbf{f}(\mathbf{x}_{d_l}, \mathbf{a}_t^{l0})) \frac{\partial \mathbf{f}(\mathbf{x}_{d_l}, \mathbf{a}_t^{l0})}{\partial \mathbf{a}_t^l} \end{bmatrix},$$

where

$$\nabla d_{it}(\mathbf{f}(\mathbf{x}_i, \mathbf{a}_t^{l0})) = \left[\frac{\partial d_{it}(\mathbf{f}(\mathbf{x}_i, \mathbf{a}_t^{l0}))}{\partial x} \quad \frac{\partial d_{it}(\mathbf{f}(\mathbf{x}_i, \mathbf{a}_t^{l0}))}{\partial y} \right]^{\text{T}} \in \Re^{2 \times 1}$$

is the spatial gradient of the image $\mathbf{d}_t$ warped with $\mathbf{a}_t^{l0}$ at the position $\mathbf{x}_i$. $(\partial \mathbf{f}(\mathbf{x}_i, \mathbf{a}_t^{l0}) / \partial \mathbf{a}_t^l) \in \Re^{2 \times 6}$ is the derivative of the parametric motion w.r.t. the motion parameters evaluated at the pixel $\mathbf{x}_i$ and warped with the initial motion parameters $\mathbf{a}_t^{l0}$. In the case that $\mathbf{f}(\mathbf{x}_p, \mathbf{a}_t^l)$ is an affine model, $\partial \mathbf{f}(\mathbf{x}_p, \mathbf{a}_t^{l0}) / \partial \mathbf{a}_t^l$ would be equal to

$$\frac{\partial \mathbf{f}(\mathbf{x}_p, \mathbf{a}_t^{l0})}{\partial \mathbf{a}_t^l} = \begin{bmatrix} 1 & x_p - x_c & y_p - y_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_p - x_c & y_p - y_c \end{bmatrix}. \tag{7}$$

Observe that after the linearization the objective function E_{wereg}, Eq. (4), is convex in each of the parameters. For instance, $\Delta \mathbf{a}_t$ can be computed in closed form by solving a linear system of equations:

$$\begin{bmatrix} ((\mathbf{J}_t^1)^\mathrm{T}\mathbf{W}_t^1\mathbf{J}_t^1) \\ \vdots \\ ((\mathbf{J}_t^L)^\mathrm{T}\mathbf{W}_t^L\mathbf{J}_t^L) \end{bmatrix} \begin{bmatrix} \mathbf{\Delta a}_t \end{bmatrix} = \begin{bmatrix} (\mathbf{J}_t^1)^\mathrm{T}\mathbf{W}_t^1(\mathbf{d}_t^1(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^0)) - \mathbf{B}^1\mathbf{c}_t^1) \\ \vdots \\ (\mathbf{J}_t^L)^\mathrm{T}\mathbf{W}_t^L(\mathbf{d}_t^L(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^0)) - \mathbf{B}^L\mathbf{c}_t^L) \end{bmatrix},$$

where recall that $\mathbf{W}_t^l$ is a matrix containing the weights for the region l at time t. In this case, we have assumed that $\mathbf{\Delta a}_t^l = \mathbf{\Delta a}_t \; \forall l$ and drop the superscript l since all regions in the ME are assumed to have the same motion.

However, E_{wereg} is no longer convex as a joint function of these variables. In order to learn the parameters, we break the estimation problem into two sub-problems. We alternate between estimating $\mathcal{C}$ and $\mathcal{A}$ with a Gauss–Newton scheme [2,4] and learning for the basis $\mathcal{B}$ and scale parameters $\boldsymbol{\sigma}$ until convergence (see [16,15] for more detailed information). Each of the updates for $\mathcal{C}, \mathcal{A}$ and $\mathcal{B}$ are computed in closed form. This multi-linear fitting algorithm monotonically reduces the cost function, although it is not guaranteed to converge to the global minimum. We also use a coarse-to-fine strategy [2,4,12] to cope with large motions and to improve the efficiency of the algorithm. Towards that end, a Gaussian image pyramid is constructed. Each level of the pyramid is constructed by taking the image at the previous resolution level, convolving it with a Gaussian filter and subsampling. Details are given below.

- For each resolution level (coarse to fine) until convergence of $\mathcal{C}, \mathcal{A}$, and $\mathcal{B}$.
  - Until convergence of $\mathcal{C}, \mathcal{A}$.
    Until convergence of $\mathcal{A}$, rewarp $\mathbf{D}$ to $\mathbf{D}_w$ and update the motion parameters for each region by computing: $(\mathbf{a}_t^l) = (\mathbf{a}_t^l) + \mathbf{\Delta a}_t \; \forall l = 1 \dots L$.
    Update the appearance coefficients for each region and each image $((\mathbf{B}^l)^\mathrm{T}\mathbf{W}_t^l\mathbf{B}^l)\mathbf{c}_t^l = (\mathbf{B}^l)^\mathrm{T}\mathbf{W}_t^l\mathbf{d}_t(\mathbf{f}(\mathbf{x}, \mathbf{a}_t^l)) \; \forall l = 1 \dots L, \; \forall t = 1 \dots T$.
  - Update $\mathcal{B}$ preserving 85% of the energy, solving:
    $(\mathbf{C}^l(\mathbf{W}^l)^p(\mathbf{C}^l)^\mathrm{T})(\mathbf{b}^l)^p = \mathbf{C}^l(\mathbf{W}^l)^p(\mathbf{d}_w^l)^p \; \forall l = 1 \dots L, \; \forall p = 1 \dots d_l$.
  - Recompute the error, weights ($\mathbf{W}$) and the scale statistics $\boldsymbol{\sigma}$ [16].
- Propagate the motion parameters to the next finer resolution level [2,4,12] (the translation parameters are multiplied by a factor of 2). Once the motion parameters are propagated the bases are recomputed.

Since the face usually performs smooth changes in motion and appearance over time, the previous model can be improved by incorporating dynamical information as additional regularization terms into the energy function framework, minimizing:

$$E_{\mathrm{dwereg}} = E_{\mathrm{wereg}} + \sum_{t=2}^{T} \sum_{l=1}^{L} (\lambda_1 \|\mathbf{c}_t^l - \mathbf{\Gamma}_c^l\mathbf{c}_{t-1}^l\|_{\ddot{\mathbf{w}}_t} + \lambda_2 \|\mathbf{a}_t^l - \mathbf{\Gamma}_a^l\mathbf{a}_{t-1}^l\|_{\ddot{\mathbf{w}}_t}).$$

Here we have introduced the linear dynamics $\mathbf{\Gamma}_c^l$ of the appearance coefficients, and the the linear dynamics $\mathbf{\Gamma}_a^l$ of the motion parameters. The first term E_{wereg} expresses a data conservation term, while the second term introduces a temporal smoothness constraint into the model. The addition of this dynamical information will act as a regularization term to prefer smooth solutions of the appearance and motion parameters. However, due to the coupling, a more efficient technique than IRLS will be a normalize gradient descent [13].

5. Experiments and applications

5.1. Automatic learning of eigeneyes

Eyes are one of the key elements in Vision Based Human–Computer Interaction. Tracking the eye becomes a difficult task because the image changes are not solely due to motion but also to appearance change [4,18,19,26,40]. In this experiment, we automatically learn a person-specific eigeneye model without any manual cropping, except in the first image. We assume that during the training process the person is not moving far from the first frame (around 5–8 pixels). However, it is not reasonable to assume that the person is absolutely still during the training session.

Recall that Fig. 1 illustrates the eigen-registration method and shows a few images from a training set. In the first frame, we manually select the mask for the eyes, face, and background (in Fig. 3 the regions are represented). In this case, because we are assuming a small motion, the GA has not been applied for initializing the algorithm, and we minimize Eq. (4) with the robust deterministic learning method proposed, with a coarse-to-fine strategy (2 levels) over the entire training set (around 300 frames). We have presupposed that the data had few outliers, so we give σ a high value. The results are shown in Fig. 2; see Section 1 for more details.

Fig. 4 shows the normalized reconstruction error of the eye for the original training set $\mathbf{D}$ and the aligned training set $\mathbf{D}_w$ with the same number of bases. The normalized reconstruction error for the image i is $r_i = \|\mathbf{d}_i - \mathbf{Bc}_i\| / \|\mathbf{d}_i\|$. The reconstruction error resulting from our method (solid line) compared with standard
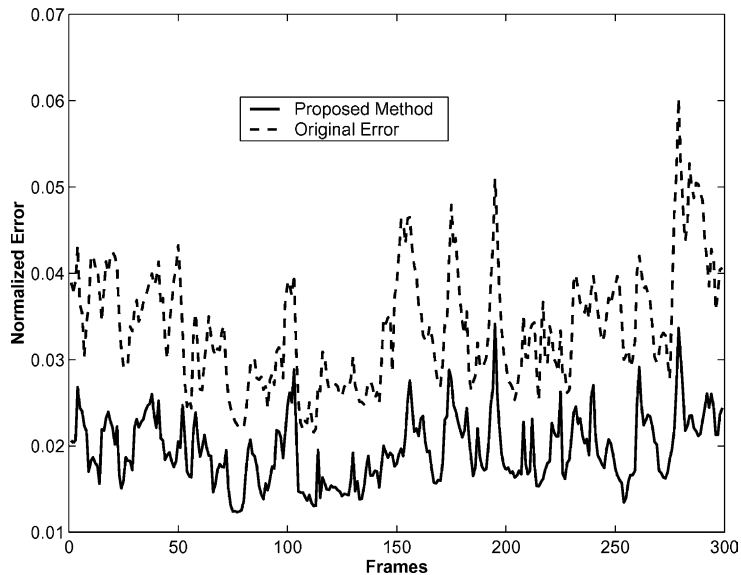

Fig. 4. Normalized reconstruction error of the right eye for the experiment 5.2 versus the number of frames. See text for details about the normalized error.

PCA (dotted line). Once the eigeneyes have been learned, tracking can be performed with deterministic techniques [4,11] or stochastic ones [19]. Applications to driver fatigue detection are being explored [19].

5.2. Automatic face learning

In this experiment, we explore the possibility of learning the entire face model, including modeling mouth changes. The modular face model is composed of four regions (see Fig. 3). Some frames of the sequence (240×320 pixels and 320 frames) are shown in Fig. 5a. In this sequence, the person can suddenly move more than 20 pixels from frame to frame, along with large scale and rotation changes. In this case we make use of the stochastic initialization with the GA for an initial estimation of the parameters.

Fig. 5b shows the normalized face (w.r.t. the first frame) reconstructed with the learned bases after the convergence of the algorithm. Recall that we have just initialized the regions in the first image and no previous appearance model was given. Notice that the reconstructed images in the "b" rows are stabilized indicating that the affine transformation from the input images to the learned eigenspaces has been accurately recovered.

The faces in Fig. 5b display variations due simply to appearance (expression) and not to motion. In this case we preserve 85% of the energy in each modular eigenspace. At this point, it is interesting to observe that ME achieves better compression factors than the regular eigenspace for the same number of parameters. Each face image (Fig. 5b) can be reconstructed with 23 parameters and further work needs to be done to determine the viability of this model for applications such as videoconferencing. Note also that these figures show the results for automatic registration and learning with respect to the training data. For video conferencing (or similar) applications where one needs to track and reconstruct the appearance and motion of the face one needs to solve for the transformation between the model and the data to be reconstructed. This is the "eigentracking" problem addressed in [4].

5.3. Virtual avatars

In this experiment we animate one face given another using PSFAMs. In general it is hard to model and animate faces and often complex models encoding the underlying physical musculature of the face are used (e.g., Candide model [23]). Here we learn the PSFAM of two people with parameterized component analysis introduced in this paper. Then, we manually select all pairs of corresponding images which share a common emotional state, i.e., we associate the face regions with equal expression content, and collect two training sets $\mathbf{D}$ and $\hat{\mathbf{D}}$ (for more information see [14]), one training set for each person. Once we have $\mathbf{D}$ and $\hat{\mathbf{D}}$, we use the recently proposed asymmetric coupled component analysis (ACCA) [14] to learn the relationship between these two sets, and predict one from the other. Fig. 6 shows frames of a virtual female face animated by the appearance of the input male face. The first column shows the original input stream ($\hat{\mathbf{D}}$); the second one, ($\mathbf{D}$), is the result of animating

Fig. 5. (a) Original image sequence; (b) reconstructed normalized face.

the face with ACCA plus the affine motion of the head. As we can observe this approach allows us to model the rich texture present on the face providing fairly realistic animations.

Fig. 6. (a) Original face; (b) animated virtual face.

6. Discussion and future work

This paper has introduced robust parameterized component analysis to learn modular subspaces that are invariant to various geometric transformations. The robust formulation of the problem extends previous work and has proven effective for learning low dimensional models of human faces. In particular, we have shown how the method can simultaneously construct an eigenspace while aligning unregistered training images. The learned eigenspace captures the motion-invariant appearance variation in the training data and the method can be applied to arbitrary parameterized deformations.

Due to the complexity of the objective function, a stochastic initialization of the algorithm has proven to be essential for avoiding local minima. Since the final solution is sensitive to the initialization from the genetic algorithm, one extension of the work here would be to take multiple initial estimates from the stochastic initialization, solve for the bases and then perform model selection. We are exploring another extension to the optimization technique that incrementally aligns the training images with an increasing number of bases (e.g., beginning with the bases corresponding to 40% of the energy and successively increasing it until 85%). Intuitively, this would first align the data w.r.t. to the most common features and later w.r.t. the more detailed ones.

While our parameterized component analysis method is a general technique for learning linear subspaces, here we have illustrated it with examples from face modeling. In particular, we have illustrated the method in the context of 2D PSFAMs

and have presented several applications of these models. Observe that parameterized component analysis always improves the quality of the appearance basis if some misalignment exists in the training set (due to manual cropping, motion of the person, etc). Although we have presented a method for learning PSFAM, the method can be also useful when improving the basis of a training set containing faces from different people. As described here, the method is appropriate for learning appearance models in an off-line process. The method could be extended to be useful for on-line learning by simply replacing the closed form solution with a gradient descent algorithm or any adaptive method. Based on the recent extension of EigenTracking [4] to deal with Support Vector Machines [1], it would also be interesting and quite straightforward to consider extending our method to other statistical learning techniques like SVM, independent component analysis, etc.

Modeling the face with modular eigenspaces coupled by the motion can result in the loss of correlations between the parts (e.g., when smiling some wrinkles appear in the eye region). Now we are working on modeling the face with symmetric coupled component analysis [14] and are experimenting with hierarchical component analysis in which one set of coefficients model the coupling between regions while each individual region has its own local variation.

Finally, the work presented in this paper on automatic learning of 2D PSFAM has the limitation of being applicable to some particular view of the face, in this case the frontal view. However, it is likely that in many real applications the head will undergo 3D motions resulting in changes to the spatial domain of the facial eigenspaces. An extension to model 3D changes is needed. We are working on extending the PSFAM to model 3D changes by incorporating shape information. This can be done using the same continuous optimization techniques described here [11,18].

Videos with the results for all the experiments performed in this paper can be down-loaded from http://www.salleURL.edu/~ftorre/.

References

[1] S. Avidan, Support vector tracking, in: Conf. on Computer Vision and Pattern Recognition, 2001.

[2] J.R. Bergen, P. Anandan, K.J. Hanna, R. Hingorani, Hierarchical model- based motion estimation, in: European Conf. on Computer Vision, 1992, pp. 237–252.

[3] M. Betke, N. Makris, Fast object recognition in noisy images using simulated annealing, in: Internat. Conf. on Computer Vision, 1994, pp. 523–530.

[4] M.J. Black, A.D. Jepson, Eigentracking: Robust matching and tracking of objects using view-based representation, Internat. J. Comput. Vision 26 (1) (1998) 63–84.

[5] M.J. Black, Y. Yacoob, Recognizing facial expressions in image sequences using local parameterized models of image motion, Internat. J. Comput. Vision 25 (1) (1997) 23–48.

[6] A. Blake, M. Isard, Active Contours, Springer, Berlin, 1998.

[7] A. Blake, A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, MA, 1987.

[8] M. La Casia, S. Sclaroff, Fastj reliable tracking under varying illumination, in: Conf. on Computer Vision and Pattern Recognition, 1999, pp. 604–609.

[9] H. Chui, A. Rangarajan, A new algorithm for non-rigid point matching, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2000, pp. 44–51.

[10] R. Cipolla, A. Pentland, Computer Vision for Human–Machine Interaction, Cambridge University Press, Cambridge, 1998.

[11] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, in: European Conf. on Computer Vision, 1998, pp. 484–498.

[12] T.F. Cootes, C.J. Taylor, Statistical models of appearance for computer vision, in: World Wide Web Publication, February, 2001. Available from <http://www.isbe.man.ac.uk/bim/refs.html>.

[13] F. de la Torre, Automatic learning of appearance face models, in: Second Internat. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, 2001, pp. 32–39.

[14] F. de la Torre, M.J. Black, Dynamic coupled component analysis, Comput. Vision Pattern Recognition (2001) 643–650.

[15] F. de la Torre, M.J. Black, A framework for robust subspace learning, Int. J. Comput. Vision 54, (2003).

[16] F. de la Torre, M.J. Black, Robust principal component analysis for computer vision, in: Internat. Conf. on Computer Vision, 2001, pp. 362–369.

[17] F. de la Torre, S. Gong, S. McKenna, View alignment with dynamically updated affine tracking, in: Internat. Conf. on Automatic Face and Gesture Recognition, 1998, pp. 510–515.

[18] F. de la Torre, J. Vitrià, P. Radeva, J. Melenchón, Eigenfiltering for flexible eigentracking, in: Internat. Conf. on Pattern Recognition, Barcelona, 2000, pp. 1118–1121.

[19] F. de la Torre, Y. Yacoob, L. Davis, A probabilisitc framework for rigid and non-rigid appearance based tracking and recognition, in: Internat. Conf. on Automatic Face and Gesture Recognition, 2000, pp. 491–498.

[20] K.I. Diamantaras, Principal Component Neural Networks (Theory and Applications), Wiley, New York, 1996.

[21] C. Eckardt, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (1936) 211–218.

[22] G.J. Edwards, C.J. Taylor, T.F. Cootes, Improving identification performance by integrating evidence from sequences, Comput. Vision Pattern Recognition (1999) 486–491.

[23] P. Eisert, B. Girod. Model-based estimation of facial expression parameters from image sequences, in: Internat. Conf. on Image Processing, 1997, pp. 418–421.

[24] B.J. Frey, N. Jojic, Transformation-invariant clustering using the EM algorithm, IEEE Trans. Pattern Anal. Machine Intell. 25 (1) (2003) 1–17.

[25] S. Geman, D. McClure, Statistical methods for tomographic image reconstruction, Bull. Int. Statist. Inst. LII (1987) 4–5.

[26] S. Gong, S. Mckenna, A. Psarrou, Dynamic Vision: From Images to Face Recognition, Imperial College Press, 2000.

[27] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Trans. Pattern Anal. Machine Intell. 20 (10) (1998) 1025–1039.

[28] F. Hampel, E. Ronchetti, P. Rousseeuw, W. Stahel, Robust Statistics: The Approach Based on Influence Functions, Wiley, New York, 1986.

[29] A. Hill, C.J. Taylor, A.D. Brett, A framework for automatic landmark identification using a new method of nonrigid correspondence, Pattern Anal. Machine Intell. 3 (22) (2000) 241–251.

[30] P.W. Holland, R.E. Welsch, Robust regression using iteratively reweighted least-squares, Commun. Statist. A (6) (1977) 813–827.

[31] T. Jebara, K. Russell, A. Pentland, Mixtures of eigenfeatures for real-time structure from texture, in: Internat. Conf. on Computer Vision, 1998.

[32] T.F. Cootes K.N. Walker, C.J. Taylor, Determining correspondences for statistical models of appearance, in: European Conf. on Computer Vision, 2000, pp. 829–843.

[33] A. Lanitis, A. Hill, T.F. Cootes, C.J. Taylor, Locating facial feature using genetic algorithms, in: Internat. Conf. on Digital Signal Processing, 1995, pp. 520–525.

[34] A. Lanitis, C. Taylor, T. Cootes, Automatic interpretation and coding of face images using flexible models, IEEE Trans. Pattern Anal. Machine Intell. 19 (7) (1997) 743–756.

[35] G. Li, Robust regression, in: D.C. Hoaglin, F. Mosteller, J.W. Tukey (Eds.), Exploring Data, Tables, Trends and Shapes, Wiley, New York, 1985.

[36] E.D. Mandel, P.S. Penev, Facial feature tracking and pose estimation in video sequences by factorial coding of the low-dimensional entropy manifolds due to the partial symmetries of faces, in: IEEE ICASSP, vol. IV, 2000, pp. 2345–2348.

[37] A.M. Martínez, Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class, IEEE Trans. Pattern Anal. Machine Intell. 24 (6) (2002) 748–763.

[38] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.

[39] B. Moghaddam, Principal manifolds and bayesian subspaces for visual recognition, in: Seventh Internat. Conf. on Computer Vision, 1999, pp. 1131–1136.

[40] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, Pattern Anal. Machine Intell. 19 (7) (1997) 137–143.

[41] S.K. Nayar, T. Poggio, Early Visual Learning, Oxford University Press, Oxford, 1996.

[42] P.S. Penev, Local feature analysis: a statistical theory for information representation and transmission, Ph.D. Thesis, The Rockefeller University, 1998.

[43] P.S. Penev, J.J. Atick, Local feature analysis: a general statistical theory for object representation, Network: Comput. Neural Syst. 7 (3) (1996) 477–500.

[44] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: IEEE Conf. on Computer Vision and Pattern Recognition, 1994, pp. 84–91.

[45] R.P.N. Rao, Development of localized oriented receptive fields by learning a translation-invariant code for natural images, Network: Comput. Neural Syst. 9 (1998) 219–234.

[46] S, Romdhani, S. Gong, A. Psarrou, Multi-view nonlinear active shape model using kernel pca, in: British Machine Vision Conf., 1999, pp. 483–492.

[47] H. Schewitzer, Optimal eigenfeature selection by optimal image registration, in: Conf. on Computer Vision and Pattern Recognition, 1999, pp. 219–224.

[48] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, J. Opt. Soc. Am. A 4 (3) (1987) 519–524.

[49] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognitive Neurosci. 3 (1) (1991) 71–86.

[50] T. Vetter, N.F. Troje, Separation of texture and shape in images of faces for image coding and synthesis, J. Opt. Soc. Am. A (14) (1997) 2152–2161.

[51] L. Wiskott, J.M. Fellous, N. Krüger, C. von der Malsburg, Face recognition using by elastic bunch graph matching, PAMI 19 (7) (1997) 775–779.

[52] M. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Trans. Pattern Anal. Machine Intell. (2001).