

# Parametric Stereo for Multi-Pose Face Recognition and 3D-Face Modeling

Rik Fransens, Christoph Strecha, Luc Van Gool

PSI ESAT-KUL  
Leuven, Belgium

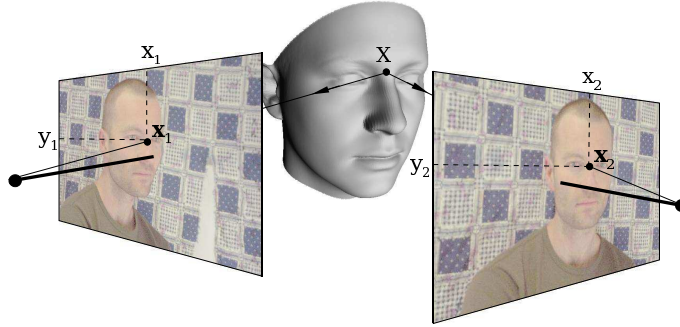
## Abstract

*This paper presents a new method for face modeling and face recognition from a pair of calibrated stereo cameras. In a first step, the algorithm builds a stereo reconstruction of the face by adjusting the global transformation parameters and the shape parameters of a 3D morphable face model. The adjustment of the parameters is such that stereo correspondence between both images is established, i.e. such that the 3D-vertices of the model project on similarly colored pixels in both images. In a second step, the texture information is extracted from the image pair and represented in the texture space of the morphable face model. The resulting shape and texture coefficients form a person specific feature vector and face recognition is performed by comparing query vectors with stored vectors. To validate our algorithm, an extensive image database was built. It consists of stereo-pairs of 70 subjects. For recognition testing, the subjects were recorded under 6 different head directions, ranging from a frontal to a profile view. The face recognition results are very good, with 100% recognition on frontal views and 97% recognition on half-profile views.*

## 1. Introduction

Over the past decades, the task of automatic face recognition has received considerable attention from the computer vision community. One of the driving forces behind this research is the wide range of commercial and law enforcement applications related to it [16]. Furthermore, the human capability of recognizing faces under variable viewing conditions which include light variations, differences in pose, and the presence or absence of facial features (glasses, beards,...) is remarkable, and keeps on attracting the attention of researchers from different fields.

Given the vast number of face recognition related publications, it is impossible to give a detailed account of past research. Here, we restrict ourselves to a short overview of some landmark papers, where we follow the taxonomy proposed by Zhao *et al.* [16]. For the particular task of face recognition from still images, Zhao *et al.* distinguish between three main categories, being (i) holistic matching methods, (ii) feature based or structural matching methods and (iii) hybrid methods which combine characteristics of both approaches. In the first category, the visual content of the complete face region is used as input for the classification system. The system then extracts a low-dimensional feature vector and compares it to stored examples. Typical examples are the PCA-based Eigenfaces technique [14,11], Fisherfaces [2] and ICA-based representations [1]. In the



**Fig. 1.** Geometry of the parametric stereo setting. The 3D-vertices of the face model are projected onto both images, and the model is manipulated to establish stereo correspondence between the image values at the locations of these projections.

second category, the position and appearance of local features like eyes, nose, etc. are determined and a feature vector is built from these descriptors. A typical example is the Elastic Bunch Graph Matching system [15], which uses 'wavelet jets' to encode local appearance. Many successful systems belong to the third category, and use both local and global descriptors. Notable contributions are the modular Eigenfaces approach [12] and the Flexible Appearance Model [10] which uses an ASM-model [8] to encode shape, and PCA to encode image intensities.

The major challenge in automatic face recognition is to develop a system that performs illumination and pose invariant recognition. An interesting approach to illumination invariant recognition is based on the so-called Illumination Cone [3]. One of the most early attempts to solve the multi-pose recognition problem is due to Beymer et al. [4,5]. The method uses a vectorized image representation at each pose, which allows to map the texture information onto a (frontal) reference shape. Arguably the most principled approach to pose invariant recognition makes use of 3D morphable face models. Blanz and Vetter [6] introduced a flexible 3D model learned from examples of individual 3D face data. In [7] a morphable component model is fitted against a multi-pose database of 68 subjects. The resulting shape and texture coefficients form a person specific feature vector, and face recognition is performed by comparing the computed feature vector with a set of stored vectors.

In this paper, we propose a multi-camera approach to face recognition, which addresses the problems of illumination and pose variation. In our setup, two calibrated cameras are used, and the algorithm computes a 3D-shape and texture representation of the face in front of the system. These representations are parametrized by the linear shape and texture coefficients of a 3D-morphable face model. In a first step, the 3D-shape of the face is determined. Rather than first computing a dense depth map of the scene, and then approximating the face related part of this map within the shape-space of the 3D-model, we *directly* fit the morphable 3D-model to the set of stereo-images, hence the name *parametric stereo*. This greatly reduces the degrees of freedom (DOFs) in the depth-from-stereo problem: from one DOF per pixel to the number of shape parameters of the 3D-model plus 6 (the DOFs related to rotational and translational com-

ponents of the global transformation). Next, the texture from both images are mapped onto the vertices of the 3D-model, and this shape and pose free texture is described in terms of the linear texture model of the 3D-morphable model. The geometry of parametric stereo is shown in Fig.(1).

Using a 3D face model to constraint 3D solutions to possible model realizations is not new. For example, in the context of structure-from-motion, such an approach was followed by Shan *et al.* [13] and Dimitrijevic *et al.* [9]. In structure-from-motion, an uncalibrated video stream is used as input, and the algorithm must simultaneously estimate the unknown camera parameters and the facial model parameters. In [9], for a given video frame, 2D point-correspondences are established in neighboring frames and the camera and model parameters are optimized by means of bundle-adjustment. The minimization criterion is the reprojection error of the 3D-points that are obtained by intersecting the current model hypothesis with the camera rays defined by the 2D-points in the central frame. This criterion is not symmetric w.r.t. the input images, however, the authors argue that the introduced biases cancel each other because many point correspondence pairs are used. In our approach, on the other hand, the cameras are already calibrated and the stereo images are captured simultaneously. This allows us to formulate of a symmetric criterion, which measures the quality of the model fit by color-differences, rather than reprojection distances, of corresponding points.

The advantage of the proposed method, compared to the approach of Blanz and Vetter [7], is that the shape and texture computations are performed separately. Given predominant diffuse or Lambertian reflection, the perceived color of a particular point of the face is the same in all images. Therefore, shape optimization is possible without having to worry about the number of lights in the scene, their intensities and the shadows they cast on the face. Next, in a separate computation, and with knowledge of the facial shape (i.e. surface normal directions), the lighting effects can be compensated for while estimating the coefficients of the linear texture model. In the approach of Blanz and Vetter, on the other hand, all effects have to be accounted for simultaneously, resulting in a formidable optimization problem. Furthermore, the number of lights in the scene has to be specified beforehand. Note that the Lambertian assumption, which underlies the shape-from-stereo approach, is relatively mild, because the stereo solution is computed directly in the 3D model space. Because the modes of the morphable model are global (i.e. changing a parameter alters the global facial appearance), the method can deal with a fair amount of specular reflections which typically occur locally. The stereo-setup also puts strong constraints on the 3D-shape solution which, in principle, should allow for more accurate recognition performance than single image approaches. On the down-side, our approach requires a multi-camera setup. However, in many commercial and law enforcement applications like entrance control, PIN-code verification and surveillance, the employment of multiple cameras is no objection.

The remainder of this paper is organized as follows. In section 2 we briefly introduce the stereo setup and the morphable model, and explain the energy formulation underlying the shape and texture computations. In section 3 we discuss the model initialization and optimization related issues. Section 4 describes the experimental setup and discusses the multi-pose recognition results. We end the paper with some general conclusions and a description of future work.

## 2. Problem Setting

Suppose we have 2 images  $\mathcal{I}_i$ ,  $i \in \{1, 2\}$ , which associate a 2D-coordinate  $\mathbf{x}$  with an image value  $\mathcal{I}_i(\mathbf{x})$ . If we are dealing with color images, this value is a 3-vector and for intensity images it is a scalar. The images are taken with 2 cameras of which we know the internal and external calibrations. The cameras are represented by the  $3 \times 4$  projection matrices  $\mathbf{P}_i$ :

$$\mathbf{P}_i = \mathbf{K}_i[\mathbf{R}_i^T | -\mathbf{R}_i^T \mathbf{t}_i], \quad (1)$$

where  $\mathbf{K}_i$ ,  $\mathbf{R}_i$  and  $\mathbf{t}_i$  are the camera matrix, rotation and translation of the  $i^{\text{th}}$  camera, respectively. The projection matrices project homogeneous 3D points  $\mathbf{X}^h = [X \ Y \ Z \ 1]^T$  to homogeneous 2D points  $\mathbf{x}^h = \lambda[x \ y \ 1]^T$  linearly:  $\mathbf{x}^h = \mathbf{P}_i \mathbf{X}^h$ . The corresponding image coordinate  $\mathbf{x}$  is easily found by dividing out the homogeneous factor. We will denote the overall projection transformation as  $\mathbf{x} = \mathcal{P}_i(\mathbf{X})$ .

Furthermore, we have a morphable 3D-face model<sup>1</sup> which consists of an orthonormal shape and texture basis. This morphable model is the result of a PCA analysis of a set of 3D-laser scans of human faces. The scans have been brought into correspondence, such that the same vertex of each model corresponds to the same physical point on the face. Let  $\mathbf{S}$  be a  $3N$ -dimensional shape vector which is formed by the concatenation of the  $N$  3D-coordinates of the vertices of the facial model:

$$\mathbf{S} = [X_1 \ Y_1 \ Z_1 \ \dots \ X_N \ Y_N \ Z_N]^T.$$

Let  $\mathbf{T}$  be a  $3N$ -dimensional texture vector which is formed by the concatenation of the  $N$  RGB-color values associated with these vertices:

$$\mathbf{T} = [R_1 \ G_1 \ B_1 \ \dots \ R_N \ G_N \ B_N]^T.$$

The shape and texture vectors of a particular face can now be realized independently as linear combinations of the so-called *eigen-shapes*  $\mathbf{S}_j$  and *eigen-textures*  $\mathbf{T}_j$ :

$$\mathbf{S} = \bar{\mathbf{S}} + \sum_{j=1}^m \alpha_j \mathbf{S}_j, \quad \mathbf{T} = \bar{\mathbf{T}} + \sum_{j=1}^m \beta_j \mathbf{T}_j. \quad (2)$$

Here,  $\bar{\mathbf{S}}$  and  $\bar{\mathbf{T}}$  are the average shape and texture vector, and the linear coefficients  $\alpha_j$  and  $\beta_j$  constitute the shape and texture description vectors  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  which fully characterize the model realization. The effects of the first shape and texture eigenvectors on the average face are visualized in Fig.(2). In what follows, we will use the term *face model* to describe a particular shape and texture combination  $(\mathbf{S}, \mathbf{T})$ , and we will preserve the term *PCA model* for the generative statistical model (i.e. the collection of shape and texture averages and eigenvectors) itself. Let  $\mathbf{X}_k$ ,  $k \in \{1, \dots, N\}$ , be the  $k^{\text{th}}$  vertex of the face model, then the shape transformation of this vertex is denoted as  $\mathcal{S}(\mathbf{X}_k)$ .

The 3D-coordinates of the vertices of the face model are defined w.r.t. an object centered coordinate system. The model can be moved around by a rigid body transformation  $\mathcal{T}$  applied to each (shape-transformed) vertex of the model:

$$\mathcal{T} \circ \mathcal{S}(\mathbf{X}_k) = \mathbf{R}(\mathcal{S}(\mathbf{X}_k) - \mathbf{C}) + \mathbf{C} + \mathbf{t}, \quad (3)$$

<sup>1</sup> USF Human ID 3-D Database and Morphable Faces [6]



**Fig. 2.** Textured and untextured renderings of the face model. Left: the average model shape and the effect of the 1<sup>st</sup> eigen-shape ( $\pm 2\sigma$ ) on the average. Note the changes in scale, as well as the transition from female to male characteristics. Right: the average model texture and the effect of the 1<sup>st</sup> eigen-texture ( $\pm 2\sigma$ ) on the average.

where  $\mathbf{R}$  is a  $3 \times 3$  rotation matrix,  $\mathbf{t}$  is a translation vector, and  $\mathbf{C}$  is the geometrical mean of the average face shape. The transformation has 6 free parameters which are jointly denoted as  $\boldsymbol{\theta}$ . Note that we have not included a scale parameter because the scale variation of human faces is incorporated in the first eigen-shapes of the PCA-model.

Our goal is to estimate a particular set of global transformation, shape and texture parameters ( $\boldsymbol{\theta}$ ,  $\boldsymbol{\alpha}$ ,  $\boldsymbol{\beta}$ ), which best explain the input images  $\mathcal{I}_1$  and  $\mathcal{I}_2$ . We proceed as follows. First, in the *shape recovery* step, we determine the values of  $\boldsymbol{\theta}$  and  $\boldsymbol{\alpha}$  which establish stereo-correspondence between both input images. Put differently, we wish to find those parameter values, such that for all model vertices  $\mathbf{X}$  which are visible in both images, the image color at their respective projections in  $\mathcal{I}_1$  and  $\mathcal{I}_2$  are as much alike as possible, i.e.  $\mathcal{I}_1(\mathcal{P}_1 \circ \mathcal{T} \circ \mathcal{S}(\mathbf{X})) \sim \mathcal{I}_2(\mathcal{P}_2 \circ \mathcal{T} \circ \mathcal{S}(\mathbf{X}))$ . To reach this objective, we only manipulate the parameter sets  $\boldsymbol{\theta}$  and  $\boldsymbol{\alpha}$ . Next, in the *texture recovery* step, the color information of both images is back-projected onto the face model, giving rise to a shape-free facial texture vector. This is then described as a linear combination of eigen-textures, while simultaneously the effects of ambient and directional lighting are accounted for.

## 2.1. Shape Computation

If we write  $\mathbf{x}_{ik}$  for the projection of the  $k^{\text{th}}$  vertex of the face model in the  $i^{\text{th}}$  image, i.e.  $\mathbf{x}_{ik} = \mathcal{P}_i \circ \mathcal{T} \circ \mathcal{S}(\mathbf{X}_k)$ , the objective function we minimize is the following:

$$E_S = \sum_{k \in \mathcal{V}} w_{S,k} \|\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k})\|^2 + \lambda_S \sum_{j=1}^m \frac{\alpha_j^2}{\sigma_{S,j}^2}, \quad (4)$$

where  $\mathcal{V} \subset \{1, \dots, N\}$  indexes the points which are visible from both images. This energy consists of a data-term, which measures the color difference between the images at corresponding projection positions, and a prior-term, which constraints the shape deformation to reasonable values.

In the data-term, the contribution of the  $k^{\text{th}}$  color difference is weighted with a vertex specific weight  $w_{S,k}$ . The purpose of this weight is two-fold. First, it allows us to account for foreshortening effects in the model projection, as a result of which the majority of vertex projections cumulate nearby the contours of the face in both  $\mathcal{I}_1$  and  $\mathcal{I}_2$ . Next, it allows us to assign more importance to the frontal part of the face, i.e. the

eyes, nose and mouth regions, which are more important for revealing identity than, say the cheek or forehead regions. We use the following weighting function:

$$w_{S,k} \propto d(\mathbf{X}_k) S_k \mathbf{n}_k \cdot \mathbf{v} . \quad (5)$$

The function  $d(\mathbf{X}_k)$  is an exponentially decaying function which depends on the distance (in cylindrical coordinates) from the  $k^{th}$  vertex to the center of the face,  $S_k$  is the area of the surface patch around the  $k^{th}$  vertex,  $\mathbf{n}_k$  is the surface normal vector at this vertex and  $\mathbf{v}$  is the average viewing direction of both cameras. We include the patch area  $S_k$  because the vertices are not evenly distributed over the surface of the model (the 3D-laser sensor samples the facial surface at cylindrical coordinates).

In the prior-term,  $\sigma_{S,j}^2$  is the variance (i.e. eigenvalue) associated with the  $j^{th}$  eigen-shape of the PCA-model. The parameter  $\lambda_S$ , which we take proportional to the sum of all weights in the data-term, allows us to balance the influence of the prior-term relative to the data-term.

## 2.2. Texture Computation

Let  $I_{amb}^R, I_{amb}^G, I_{amb}^B$  be the red, green and blue intensities of the ambient light. Furthermore, let  $I_{dir}^R, I_{dir}^G, I_{dir}^B$  be the red, green and blue intensities of the directional (parallel) light, which has direction  $\mathbf{l}$ . Then the observable color  $I_k = [R_k \ G_k \ B_k]^T$  of the  $k^{th}$  vertex of the face model can be modeled as follows:

$$R_k = R_{off} + (\bar{R}_k + \sum_{j=1}^m \beta_j R_{jk}) (I_{amb}^R + I_{dir}^R \mathbf{n}_k \cdot \mathbf{l}) , \quad (6)$$

where similar equations hold for  $G_k$  and  $B_k$ . In this equation,  $R_{off}$  is an offset,  $\bar{R}_k$  and  $R_{jk}$  are the red values of the  $k^{th}$  vertex of the average texture and  $j^{th}$  eigen-texture, and  $\mathbf{n}_k$  is the normal surface vector emanating from the  $k^{th}$  vertex. Note that the model texture is used as the reflectance coefficient of a diffuse lighting model. Unlike in [7], we do not add a specular component, because we experimentally observed that the diffuse lighting model is sufficient to account for the lighting effects in our images. Given this color model, the objective function we minimize is the following:

$$E_T = \sum_{k \in \mathcal{V}} \sum_{i=1}^2 w_{T,k} \|\mathcal{I}_i(\mathbf{x}_{ik}) - I_k\|^2 + \lambda_T \sum_{j=1}^m \frac{\beta_j^2}{\sigma_{T,j}^2} . \quad (7)$$

Like in the shape computation, this energy consists of a data-term, which measures the color difference between the input images and the texture reconstruction, and a prior-term which constrains the texture deformation to reasonable values. The contribution of each vertex color is weighted by a vertex specific weight  $w_{T,k}$ , which accounts for the aforementioned foreshortening effects, and also allows us to diminish the influence of outliers in the texture reconstruction. These outliers are vertices, for whom the sampled image colors  $\mathcal{I}_i(\mathbf{x}_{ik})$  are significantly different. The differences might be caused by wrong shape reconstructions (i.e. image locations where stereo correspondence was

not established), but also by specular highlights in either of both images. We use the following weighting function:

$$w_{T,k} \propto w_{S,k} \exp\left(-\frac{1}{2} d_{\mathbf{S}}^2(\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k}))\right), \quad (8)$$

where  $d_{\mathbf{S}}^2(\mathbf{x})$  is a squared distance defined by  $\mathbf{x}^T \mathbf{S}^{-1} \mathbf{x}$ . For  $\mathbf{S}$  we take a robust estimate of the covariance matrix of the color differences  $\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k})$ .

### 3. Model Initialization and Optimization

#### 3.1. Model Initialization

Before the shape energy  $E_S$  defined in Eq.(4) is optimized w.r.t. the global transformation parameters  $\theta$  and shape parameters  $\alpha$ , the 3D-model needs to be at a reasonable start position. In this paper we assume that we have a set of feature detectors at our disposal, which are able to localize typical facial features (eyes, nose, corners of the mouth, etc.) if they are visible. Furthermore, these detectors provide us with some indication of the spatial uncertainty of the detection. Typically, feature detectors provide a detection value at each location in a certain region of interest, and report the position of maximal detection value. Let  $\hat{\mathbf{x}}_{ip}$  be the estimated position of the  $p^{\text{th}}$  feature in the  $i^{\text{th}}$  image, and let  $\mathbf{S}_{ip}$  be a  $2 \times 2$  scatter matrix which characterizes the spatial uncertainty of this estimate. For the feature points of interest, we also know the 3D-coordinates of the corresponding vertex on the morphable model. Let  $\mathbf{X}_p$  be the 3D-coordinates of the  $p^{\text{th}}$  feature, and  $\mathbf{x}_{ip} = \mathcal{P}_i \circ \mathcal{T} \circ \mathcal{S}(\mathbf{X}_p)$  be the projection of this point in the  $i^{\text{th}}$  image. The objective function we minimize is the following:

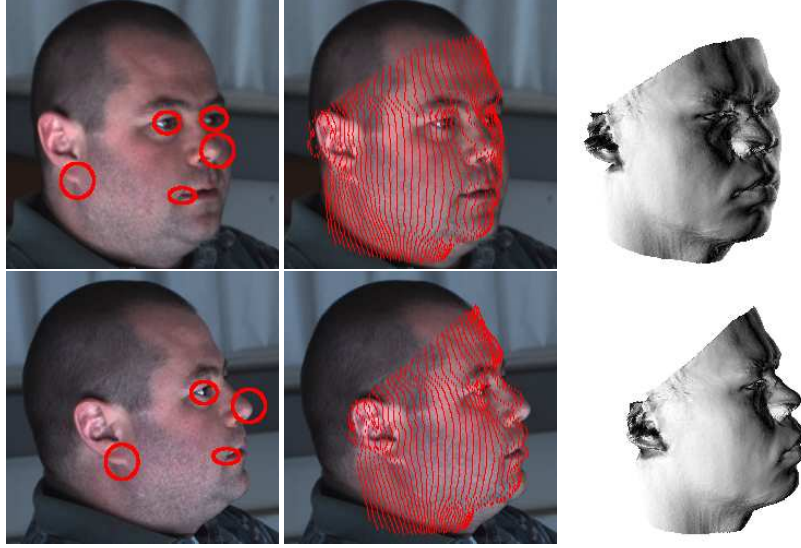
$$E_I = \sum_{i=1}^2 \sum_{p=1}^{N_p} \delta_{ip} (\hat{\mathbf{x}}_{ip} - \mathbf{x}_{ip})^T \mathbf{S}_{ip}^{-1} (\hat{\mathbf{x}}_{ip} - \mathbf{x}_{ip}), \quad (9)$$

where  $N_p$  is the total number of features we consider and  $\delta_{ip} \in \{0, 1\}$  is a binary variable which indicates whether or not the  $p^{\text{th}}$  feature was detected in the  $i^{\text{th}}$  image. The initial model position is found by minimizing  $E_I$  w.r.t. the 6 parameters of  $\theta$ . If the number of detections is large enough to render a unique solution (e.g.  $> 3$  non-colinear features are detected in both images), it is possible to further optimize  $E_I$  w.r.t. the model shape parameters  $\alpha$ . Using the same prior as in Eq.(4), the objective function becomes:

$$E_I = \sum_{i=1}^2 \sum_{p=1}^{N_p} \delta_{ip} (\hat{\mathbf{x}}_{ip} - \mathbf{x}_{ip})^T \mathbf{S}_{ip}^{-1} (\hat{\mathbf{x}}_{ip} - \mathbf{x}_{ip}) + \lambda_I \sum_{j=1}^m \frac{\alpha_j^2}{\sigma_{S,j}^2}. \quad (10)$$

We minimize this energy by Levenberg-Marquardt iterations. The gradient of  $E_I$  w.r.t. the  $j^{\text{th}}$  global transformation parameter  $\theta_j$  is given by:

$$\frac{\partial E_I}{\partial \theta_j} = -2 \sum_{i=1}^2 \sum_{p=1}^{N_p} \delta_{ip} (\hat{\mathbf{x}}_{ip} - \mathbf{x}_{ip})^T \mathbf{S}_{ip}^{-1} \mathbf{J}_{\mathcal{P}_i} \frac{\partial \mathcal{T}}{\partial \theta_j}. \quad (11)$$



**Fig. 3.** Model initialization. Left column: the input stereo pair with feature points and their spatial uncertainty. Middle column: the fit of the model guided by the feature points. The fit is relatively accurate, but alignment errors are still visible at the contour of the face. Right column: renderings of the initialized model. The reconstruction is relatively poor, but the main facial features are already visible.

Here, the  $2 \times 3$ -matrix  $\mathbf{J}_{\mathcal{P}_i}$  is the Jacobian of the projection function  $\mathcal{P}_i$  evaluated at  $\mathcal{T} \circ \mathcal{S}(\mathbf{X}_p)$  and  $\partial \mathcal{T} / \partial \theta_j$  is a 3-derivative vector evaluated at  $\mathcal{S}(\mathbf{X}_p)$ . The gradient of  $E_I$  w.r.t. the  $j^{\text{th}}$  shape parameter  $\alpha_j$  is given by:

$$\frac{\partial E_I}{\partial \alpha_j} = -2 \sum_{i=1}^2 \sum_{p=1}^{N_p} \delta_{ip} (\hat{\mathbf{x}}_{ip} - \mathbf{x}_{ip})^T \mathbf{S}_{ip}^{-1} \mathbf{J}_{\mathcal{P}_i} \mathbf{J}_{\mathcal{T}} \frac{\partial \mathbf{X}_p}{\partial \alpha_j} + 2\lambda_I \frac{\alpha_j}{\sigma_{S,j}^2}, \quad (12)$$

where the  $3 \times 3$ -matrix  $\mathbf{J}_{\mathcal{T}}$  is the Jacobian of the rigid-body transformation evaluated at  $\mathcal{S}(\mathbf{X}_p)$ , and  $\partial \mathbf{X}_p / \partial \alpha_j$  is a 3-derivative vector, which contains the XYZ-values of the  $j^{\text{th}}$  eigen-shape at the position of  $\mathbf{X}_p$ . The initialization procedure is graphically illustrated in Fig.(3).

### 3.2. Shape Optimization

After the model initialization, the 3D face model is in approximate correspondence with both input images. We now proceed with the optimization of the shape energy  $E_S$  defined in Eq.(4) w.r.t. the global transformation parameters  $\theta$  and shape parameters  $\alpha$ . The purpose of this optimization is to establish stereo correspondence between both images. The gradient of  $E_S$  w.r.t. the  $j^{\text{th}}$  global transformation parameters  $\theta_j$  is given

by:

$$\begin{aligned} \frac{\partial E_S}{\partial \theta_j} &= 2 \sum_{k \in \mathcal{V}} w_{S,k} [\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k})]^T \nabla \mathcal{I}_1 \frac{\partial \mathbf{x}_{1k}}{\partial \theta_j} - \\ & 2 \sum_{k \in \mathcal{V}} w_{S,k} [\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k})]^T \nabla \mathcal{I}_2 \frac{\partial \mathbf{x}_{2k}}{\partial \theta_j} \end{aligned} \quad (13)$$

The image gradients  $\nabla \mathcal{I}_i$  are  $3 \times 2$ -matrices and contain the spatial derivatives of the R, G and B-component of  $\mathcal{I}_i$  evaluated at positions  $\mathbf{x}_{ik}$ . The differentials  $\partial \mathbf{x}_{ik} / \partial \theta_j$  are 2-vectors defined as follows:

$$\frac{\partial \mathbf{x}_{ik}}{\partial \theta_j} = \mathbf{J}_{\mathcal{P}_i} \frac{\partial \mathcal{T}(\mathcal{S}(\mathbf{X}_k))}{\partial \theta_j} . \quad (14)$$

The gradient of  $E_S$  w.r.t. the  $j^{\text{th}}$  shape transformation parameters  $\alpha_j$  can be derived in a similar fashion:

$$\begin{aligned} \frac{\partial E_S}{\partial \alpha_j} &= 2 \sum_{k \in \mathcal{V}} w_{S,k} [\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k})]^T \nabla \mathcal{I}_1 \frac{\partial \mathbf{x}_{1k}}{\partial \alpha_j} - \\ & 2 \sum_{k \in \mathcal{V}} w_{S,k} [\mathcal{I}_1(\mathbf{x}_{1k}) - \mathcal{I}_2(\mathbf{x}_{2k})]^T \nabla \mathcal{I}_2 \frac{\partial \mathbf{x}_{2k}}{\partial \alpha_j} \end{aligned} \quad (15)$$

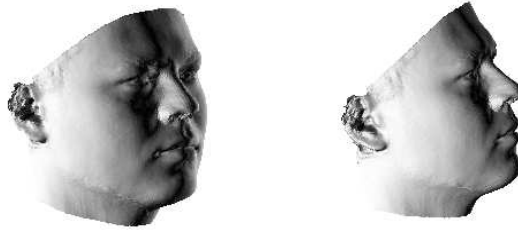
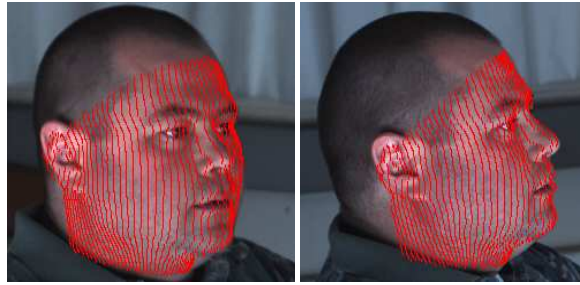
where the differentials  $\partial \mathbf{x}_{ik} / \partial \alpha_j$  are given by:

$$\frac{\partial \mathbf{x}_{ik}}{\partial \alpha_j} = \mathbf{J}_{\mathcal{P}_i} \mathbf{J}_T \mathbf{X}_{jk} . \quad (16)$$

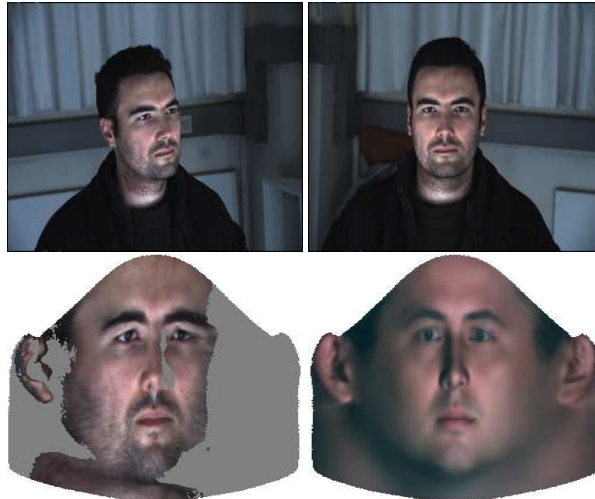
Here  $\mathbf{X}_{jk}$  is the  $k^{\text{th}}$  component of the  $j^{\text{th}}$  eigen-shape. We optimize  $E_S$  with conjugate gradient. During optimization, model vertices do not project onto integral positions in  $\mathcal{I}_i$ , and we use bilinear interpolation to sample pixel and gradient values from the images. To avoid local minima, a pyramidal coarse-to-fine strategy with 3 pyramidal levels is followed. At the most coarse image scale, the prior parameter  $\lambda_S$  is set to 20.0, whereas at the finest image scale this value is lowered to 5.0. To speed up convergence, we use a vertex sub-sampling approach, and the number of selected vertices is increased at every pyramidal level (1000, 2000 and 3000 at the respective pyramid levels). At regular intervals, visibility is recomputed. On a standard desktop (P4, 2.6GHz), it takes on average 35 seconds for the algorithm to converge. The effect of the optimization procedure on the model fit is graphically illustrated in Fig. (4). Different views of a subject, together with untextured renderings of the 3D model in the same pose, are shown in Fig. (7).

### 3.3. Texture Optimization

After the shape extraction step, the textures from both images are mapped onto the vertices of the 3D-model. The resulting shape and pose free texture is described in terms of the linear texture model of the 3D-morphable model. This is done by minimizing the energy  $E_T$  in Eq.(7) w.r.t. the light source variables and texture coefficients  $\beta$ , where we only take into account the texture of the points which are visible in both images. We optimize  $E_T$  with conjugate gradient, and set  $\lambda_T$  to 5.0. An example of a texture reconstruction is shown in Fig.(5).



**Fig. 4.** Shape optimization. Top row: the input stereo-pair with an overlay of the final model shape. Note that, compared to the initialization result in Fig.(3), the accuracy of the fit has improved. Particularly the alignment errors at the contour of the face have largely disappeared. Bottom row: renderings of the untextured model at its final position.



**Fig. 5.** Texture reconstruction. Top row: the stereo-pair of test view one. Bottom row, left: the average of the textures extracted from both images. The facial regions which are not visible from both images are displayed in gray. Note that the average has remained sharp, which is an indication of the quality of the shape reconstruction. Bottom row, right: the texture reconstruction by the texture model.

## 4. Experiments and Discussion

To validate our algorithm, an extensive image database was built. It consists of stereo-pairs of 70 subjects (35 males, 35 females), recorded from 6 different viewpoints. An example of the stereo-pairs of one subject is shown in Fig.(6). The first viewpoint, which is frontal w.r.t. the stereo-pair, is used as training or enrollment data. An example is shown in the left column of Fig.(6). The shape and texture vectors of these faces are stored in the memory of the recognition system, and all queries are compared to them. The next 5 viewpoints range from a frontal to a profile view w.r.t. the viewing direction of the first camera, in equal steps of  $\pi/8$  radians. These views will serve as test data from which query vectors are computed. Note that the first test view, which is frontal, was recorded separately from the training data. The lighting conditions remained constant over the course of the recordings. Lighting is complex with multiple light sources and reflectors in the neighborhood of the subject. From Fig.(6) it can be appreciated that the recorded intensities on the facial part of the image vary considerably over the different viewpoints.



**Fig. 6.** Stereo-pair database: one face from the stereo database. The first row shows the images from the left camera of the stereo-pair, the second row shows the images taken from the right camera. Left column: the training viewpoint, which shows the subjects in frontal pose w.r.t. the stereo cameras. Columns 2 to 6: the five test views with increasing angle w.r.t. the training view.

For a particular person and particular viewpoint, we then compute the face model parameters  $(\alpha, \beta)$ . These are used as a query vector, and all training vectors are sorted according to their distance from the query vector. The distance function we use is a weighted sum of Mahalanobis distances, defined as follows:

$$d(\alpha_1, \beta_1; \alpha_2, \beta_2) = \lambda_\alpha (\alpha_1 - \alpha_2)^T \mathbf{C}_\alpha^{-1} (\alpha_1 - \alpha_2) + \lambda_\beta (\beta_1 - \beta_2)^T \mathbf{C}_\beta^{-1} (\beta_1 - \beta_2). \quad (17)$$

Here,  $\lambda_\alpha$  and  $\lambda_\beta$  are weights which allow us to manipulate the importance of the shape coefficients w.r.t. the texture coefficients, and  $\mathbf{C}_\alpha$  and  $\mathbf{C}_\beta$  are the model covariance matrices of shape and geometry. If the correct person is at the first position of the sorted list of training vectors, we denote this as a correct identification or 'rank-1' match. In the results, we report the percentage of correct identifications for each test viewpoint. We also show the percentage of queries for which the correct person is amongst the first 3 and 5 positions ('rank-3' and 'rank-5' matches). To gain more insight in the roles of shape and

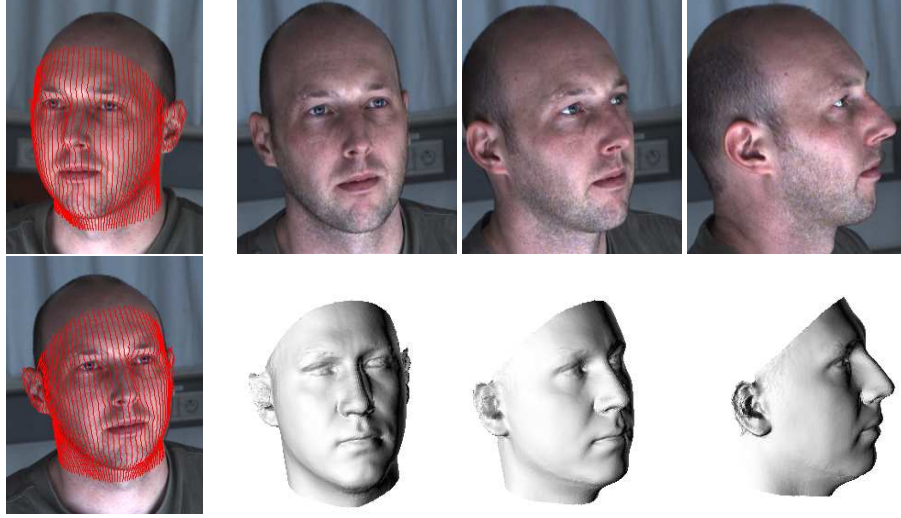
texture in the recognition performance, we also report recognition rates when we only use the shape or the texture vectors in the queries. In all experiments, 50 shape and 50 texture components were used. The results are shown in Table (1). From these figures,

	test 1	test 2	test 3	test 4	test 5
rank 1	<b>90.0</b>	<b>87.1</b>	<b>68.6</b>	<b>52.9</b>	<b>41.4</b>
rank 3	92.9	98.6	84.3	71.4	60.0
rank 5	94.3	98.6	90.0	85.7	72.9
rank 1	<b>91.4</b>	<b>67.1</b>	<b>30.0</b>	<b>17.1</b>	<b>11.4</b>
rank 3	92.9	84.3	44.3	25.7	12.9
rank 5	92.9	90.0	52.9	38.6	17.1
rank 1	<b>94.3</b>	<b>94.3</b>	<b>77.1</b>	<b>58.6</b>	<b>45.7</b>
rank 3	97.1	95.7	87.1	80.0	62.9
rank 5	97.1	95.7	92.9	85.7	68.6

**Table 1.** Recognition rates without coefficient weighting. Top table: recognition rates based on geometry only. Middle table: recognition rates based on texture only. Bottom table: recognition rates based on combined geometry and texture features. Rank-1 matches are indicated in bold.

we immediately see that, except for the frontal test view ('test 1'), shape based recognition performs better than texture based recognition. Also, the texture based recognition rates drop sharply when the test views have increasing angle w.r.t. the training view ('test 2,3,...'). Both cues seem to be co-operative, i.e. the results based on both shape and geometry features are better than the results based on the separate features.

In Blanz *et al.* [7], a coefficient weighting method was introduced, which takes into account the variation of model coefficients obtained from different images of the same person. These variations may be due to several reasons. First of all, when the model is fitted against images of the same person but from a different viewpoint, different facial features are estimated with a different accuracy. For example, on the frontal views we can expect an accurate assessment of the width and height of the face. For the 'depth related features' like the profile of nose, the prominence of eyebrows etc..., we can expect a much poorer assessment. On the profile views, on the other hand, the assessment of the width of the face is much more difficult, whereas the profile of the nose can be estimated accurately. Secondly, different lighting conditions can introduce ambiguities in the texture reconstruction, such as skin complexion versus intensity of illumination [7]. We also noticed that there is light-source variation within the eigen-textures of the model. This causes instabilities in the computation of texture coefficients, because the model is able to explain lighting conditions both with its light source variables and its linear model. This probably explains the relatively poor texture based recognition results from Table (1). Finally, if the PCA model is not able to reproduce the faces in the input images, the algorithm will do as well as possible and will distribute the residual error over its coefficients. This distribution is likely to be different for different viewpoints.



**Fig. 7.** *Shape optimization. Left column: the stereo-pair from which the 3D reconstruction is computed with an overlay of the final model shape. Columns 2,3 and 4: new views of the subject and the untextured renderings of the 3D model at the corresponding positions and orientations.*

To account for these effects, the distance function in Eq.(17) is modified, to suppress directions with high within-person variation in the whitened coefficient spaces. The whitening transformation compensates for the relative magnitude of the coefficients and transforms  $\alpha$  and  $\beta$  to  $\alpha' = C_\alpha^{-1/2} \alpha$  and  $\beta' = C_\beta^{-1/2} \beta$ , respectively. To suppress directions with high within-person variation, the pooled within-person scatter matrices  $\mathbf{W}_\alpha$  and  $\mathbf{W}_\beta$  are introduced into the Mahalanobis distances. To estimate  $\mathbf{W}_\alpha$  and  $\mathbf{W}_\beta$  independently from our test-set, we recorded a training set consisting of stereo-pairs of 30 more subjects (15 males, 15 females). The viewing conditions of this second database are similar, but the lighting conditions are slightly different. Let  $N = 30$  and  $V = 5$  be the number of persons and number of viewpoints per person in this trainingset. Furthermore, let  $\alpha'_{ij}$  and  $\beta'_{ij}$  be the computed (whitened) shape and texture coefficients of the  $i^{th}$  person in the  $j^{th}$  view point, and let  $\langle \alpha'_i \rangle$  and  $\langle \beta'_i \rangle$  be the average shape and texture coefficients of the  $i^{th}$  person over all  $V$  viewpoints, respectively. The weighting matrices are defined as follows:

$$\begin{aligned} \mathbf{W}_\alpha &= \frac{1}{N} \sum_i \frac{1}{V} \sum_j (\alpha'_{ij} - \langle \alpha'_i \rangle) (\alpha'_{ij} - \langle \alpha'_i \rangle)^T \\ \mathbf{W}_\beta &= \frac{1}{N} \sum_i \frac{1}{V} \sum_j (\beta'_{ij} - \langle \beta'_i \rangle) (\beta'_{ij} - \langle \beta'_i \rangle)^T . \end{aligned} \quad (18)$$

These matrices estimate the spread of the model coefficients w.r.t. changes in viewpoint, and can be used to identify consistent and inconsistent directions in the shape and texture feature spaces. Taking the shape coefficients as an example, directions  $\alpha'$  charac-

terized by a high value of  $\alpha'^T \mathbf{W}_\alpha \alpha'$  are inconsistent w.r.t. the viewpoint from which these coefficients are computed, whereas directions  $\alpha'$  characterized by a low value of  $\alpha'^T \mathbf{W}_\alpha \alpha'$  are relatively stable w.r.t. viewpoint. By incorporating these weights in Eq.(17), the importance of inconsistent directions can be diminished. The new distance function is given by:

$$d(\alpha_1, \beta_1; \alpha_2, \beta_2) = \lambda_\alpha (\alpha_1 - \alpha_2)^T \mathbf{C}_\alpha^{-\frac{1}{2}} \mathbf{W}_\alpha^{-1} \mathbf{C}_\alpha^{-\frac{1}{2}} (\alpha_1 - \alpha_2) + \lambda_\beta (\beta_1 - \beta_2)^T \mathbf{C}_\beta^{-\frac{1}{2}} \mathbf{W}_\beta^{-1} \mathbf{C}_\beta^{-\frac{1}{2}} (\beta_1 - \beta_2). \quad (19)$$

The final results are shown in Table (2). The performance boost is quite significant. Especially the recognition rate of the texture-component seems to benefit from the coefficient weighting.

	test 1	test 2	test 3	test 4	test 5
rank 1	<b>94.3</b>	<b>84.3</b>	<b>80.0</b>	<b>74.3</b>	<b>60.0</b>
rank 3	98.6	95.7	94.3	88.6	75.7
rank 5	100.0	95.7	94.3	91.4	87.1
rank 1	<b>94.3</b>	<b>97.1</b>	<b>80.0</b>	<b>68.6</b>	<b>42.9</b>
rank 3	95.7	98.6	91.4	82.9	67.1
rank 5	95.7	98.6	97.1	85.7	81.4
rank 1	<b>100.0</b>	<b>98.6</b>	<b>97.1</b>	<b>91.4</b>	<b>82.9</b>
rank 3	100.0	98.6	98.6	92.9	90.0
rank 5	100.0	100.0	100.0	97.1	92.9

**Table 2.** Recognition results with coefficient weighting. Top table: recognition rates based on geometry only. Middle table: recognition rates based on texture only. Bottom table: recognition rates based on combined geometry and texture features.  $\lambda_\alpha$  and  $\lambda_\beta$  were set to 0.7 and 0.3.

## 5. Conclusions

We presented a new method for face modeling and face recognition from a pair of calibrated stereo cameras. In the *shape extraction* step, the algorithm builds a stereo reconstruction of the face by adjusting the global transformation and shape parameters of a 3D-morphable face model. Next, in the *texture extraction* step, texture is sampled from the image pair and represented in the texture space of the morphable face model. The resulting shape and texture parameters are characteristic for the analyzed face, and can subsequently be used for face recognition.

In a face recognition experiment on a stereo database of 70 subjects, we reported recognition rates for 5 different viewpoints. The initial recognition results are reasonable but a decrease in performance is noted for profile views. Particularly the texture feature vector has relatively low discriminative power. However, after weighting the coefficients with the pooled within-person scatter matrices – estimated independently

from the test set – detection rates increase significantly. The resulting face recognition system has state-of-the-art performance.

We believe that, with a refinement of the morphable face model, the level of performance can still increase. An obvious improvement is the usage of a component based model with enhanced representative power. Furthermore, we noticed that there is evidence of light-source variation within the eigen-textures of the model, which causes instabilities in the computation of texture coefficients. These variations should be accounted for, prior to PCA-analysis.

**Acknowledgments** The authors acknowledge support from EU project Reveal-This and IWT project 020195.

## References

1. Bartlett, M. S., Lades, H. M., Sejnowski, T. J., “Independent component representations for face recognition,” *Proc. of the SPIE Symposium on Electronic Imaging: Science and Technology*, pp. 528-539, 1998.
2. Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J., “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” *IEEE Trans. PAMI*, Vol. 19, No. 7, pp. 711-720, 1997.
3. Belhumeur, P., Kriegman, D., “What is the Set of Images of an Object Under All Possible Lighting Conditions?,” *IJCV*, Vol. 28, No. 3, pp. 245-260, 1998.
4. Beymer, D., “Face recognition under varying pose,” Tech. Rep. 1461. MIT AI Lab, Massachusetts Institute of Technology, Cambridge, MA
5. Beymer, D., “Vectorizing face images by interleaving shape and texture computations,” Tech. Rep. 1537, MIT AI Lab, Massachusetts Institute of Technology, Cambridge, MA
6. Blanz, V., Vetter, T., “A morphable model for the synthesis of 3D faces,” *SIGGRAPH '99: Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 187-194, 1999.
7. Blanz, V., Vetter, T., “Face Recognition Based on Fitting a 3D Morphable Model,” *IEEE Trans. PAMI*, Vol. 25, No. 9, pp. 1063-1074, 2003.
8. Cootes, T. F., Taylor, C. J., Cooper, D. H., Graham, J., “Active Shape Models - Their Training and Application,” *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp 38-59, 1995.
9. Dimitrijevic, M., Ilic, S., Fua, P., “Accurate Face Models from Uncalibrated and Ill-Lit Video Sequences,” *IEEE Proc. Int. Conf. CVPR*, Vol. 2, pp. 1034-1041, 2004.
10. Lanitis, A., Taylor, C. J., Cootes, T. F., “Automatic Face Identification System Using Flexible Appearance Models,” *Image Vis. Comput.*, Vol. 13, pp. 393-401, 1995.
11. Moghaddam, B., Pentland, A., “Probabilistic Visual Learning for Object Representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 19, pp. 696-710, 1997.
12. Pentland, A., Moghaddam, B., Starner, T., “View-Based and Modular Eigenspaces for Face Recognition,” *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pp. 84-91, 1994.
13. Shan, Y., Liu, Z., Zhang, Z., “Model-Based Bundle Adjustment with Application to Face Modeling,” *International Conference on Computer Vision*, Vol. 2, p. 644, 2001.
14. Turk, M., Pentland, A., “Eigenfaces for Recognition,” *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991.
15. Wiskott, L., Fellous, J.-M., Kruger, N., von der Malsburg, C., “Face Recognition by Elastic Bunch Graph Matching,” *IEEE Trans. PAMI*, Vol. 19, No. 7, pp. 775-779, 1997.
16. Zhao, W., Chellappa, R., Phillips, P. J., Rosenfeld, A., “Face recognition: A literature survey,” *ACM Comput. Surv.*, Vol. 35, No. 4, pp. 399-458, 2003.