Face Recognition Based on Fitting a 3D Morphable Model

Volker Blanz and Thomas Vetter, *Member*, *IEEE*

Abstract—This paper presents a method for face recognition across variations in pose, ranging from frontal to profile views, and across a wide range of illuminations, including cast shadows and specular reflections. To account for these variations, the algorithm simulates the process of image formation in 3D space, using computer graphics, and it estimates 3D shape and texture of faces from single images. The estimate is achieved by fitting a statistical, morphable model of 3D faces to images. The model is learned from a set of textured 3D scans of heads. We describe the construction of the morphable model, an algorithm to fit the model to images, and a framework for face identification. In this framework, faces are represented by model parameters for 3D shape and texture. We present results obtained with 4,488 images from the publicly available CMU-PIE database and 1,940 images from the FERET database.

Index Terms—Face recognition, shape estimation, deformable model, 3D faces, pose invariance, illumination invariance.

1 Introduction

In face recognition from images, the gray-level or color values provided to the recognition system depend not only on the identity of the person, but also on parameters such as head pose and illumination. Variations in pose and illumination, which may produce changes larger than the differences between different people's images, are the main challenge for face recognition [39]. The goal of recognition algorithms is to separate the characteristics of a face, which are determined by the intrinsic shape and color (texture) of the facial surface, from the random conditions of image generation. Unlike pixel noise, these conditions may be described consistently across the entire image by a relatively small set of extrinsic parameters, such as camera and scene geometry, illumination direction and intensity. Methods in face recognition range within two fundamental strategies: One approach is to treat these parameters as separate variables and model their functional role explicitly. The other approach does not formally distinguish between intrinsic and extrinsic parameters, and the fact that extrinsic parameters are not diagnostic for faces is only captured statistically.

The latter strategy is taken in algorithms that analyze intensity images directly using statistical methods or neural networks (for an overview, see Section 3.2 in [39]).

To obtain a separate parameter for orientation, some methods parameterize the manifold formed by different views of an individual within the eigenspace of images [16], or define separate view-based eigenspaces [28]. Another way of capturing the viewpoint dependency is to represent faces by eigen-lightfields [17].

- *V. Blanz is with the Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany. E-mail: blanz@mpi-sb.mpg.de.*
- *T. Vetter is with the University of Basel, Departement Informatik, Bernoullistrasse 16, 4057 Basel, Switzerland. E-mail: thomas.vetter@unibas.ch.*

Manuscript received 9 Aug. 2002; accepted 10 Mar. 2003. Recommended for acceptance by P. Belhumeur. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117108.

Two-dimensional face models represent gray values and their image locations independently [3], [4], [18], [23], [13], [22]. These models, however, do not distinguish between rotation angle and shape, and only some of them separate illumination from texture [18]. Since large rotations cannot be generated easily by the 2D warping used in these algorithms due to occlusions, multiple view-based 2D models have to be combined [36], [11]. Another approach that separates the image locations of facial features from their appearance uses an approximation of how features deform during rotations [26].

Complete separation of shape and orientation is achieved by fitting a deformable 3D model to images. Some algorithms match a small number of feature vertices to image positions, and interpolate deformations of the surface in between [21]. Others use restricted, but class-specific deformations, which can be defined manually [24], or learned from images [10], from nontextured [1] or textured 3D scans of heads [8].

In order to separate texture (albedo) from illumination conditions, some algorithms, which are derived from shape-from-shading, use models of illumination that explicitly consider illumination direction and intensity for Lambertian [15], [38] or non-Lambertian shading [34]. After analyzing images with shape-from-shading, some algorithms use a 3D head model to synthesize images at novel orientations [15], [38].

The face recognition system presented in this paper combines deformable 3D models with a computer graphics simulation of projection and illumination. This makes intrinsic shape and texture fully independent of extrinsic parameters [8], [7]. Given a single image of a person, the algorithm automatically estimates 3D shape, texture, and all relevant 3D scene parameters. In our framework, rotations in depth or changes of illumination are very simple operations, and all poses and illuminations are covered by a single model. Illumination is not restricted to Lambertian reflection, but takes into account specular reflections and

cast shadows, which have considerable influence on the appearance of human skin.

Our approach is based on a morphable model of 3D faces that captures the class-specific properties of faces. These properties are learned automatically from a data set of 3D scans. The morphable model represents shapes and textures of faces as vectors in a high-dimensional face space, and involves a probability density function of natural faces within face space.

Unlike previous systems [8], [7], the algorithm presented in this paper estimates all 3D scene parameters automatically, including head position and orientation, focal length of the camera, and illumination direction. This is achieved by a new initialization procedure that also increases robustness and reliability of the system considerably. The new initialization uses image coordinates of between six and eight feature points. Currently, most face recognition algorithms require either some initialization, or they are, unlike our system, restricted to front views or to faces that are cut out from images.

In this paper, we give a comprehensive description of the algorithms involved in 1) constructing the morphable model from 3D scans (Section 3), 2) fitting the model to images for 3D shape reconstruction (Section 4), which includes a novel algorithm for parameter optimization (Appendix B), and 3) measuring similarity of faces for recognition (Section 5). Recognition results for the image databases of CMU-PIE [33] and FERET [29] are presented in Section 5. We start in Section 2 by describing two general strategies for face recognition with 3D morphable models.

2 Paradigms for Model-Based Recognition

In face recognition, the set of images that shows all individuals who are known to the system is often referred to as *gallery* [39], [30]. In this paper, one gallery image per person is provided to the system. Recognition is then performed on novel *probe* images. We consider two particular recognition tasks: For *identification*, the system reports which person from the gallery is shown on the probe image. For *verification*, a person claims to be a particular member of the gallery. The system decides if the probe and the gallery image show the same person (cf. [30]).

Fitting the 3D morphable model to images can be used in two ways for recognition across different viewing conditions:

Paradigm 1. After fitting the model, recognition can be based on model coefficients, which represent intrinsic shape and texture of faces, and are independent of the imaging conditions. For identification, all gallery images are analyzed by the fitting algorithm, and the shape and texture coefficients are stored (Fig. 1). Given a probe image, the fitting algorithm computes coefficients which are then compared with all gallery data in order to find the nearest neighbor. Paradigm 1 is the approach taken in this paper (Section 5).

Paradigm 2. Three-dimension face reconstruction can also be employed to generate synthetic views from gallery or probe images [3], [35], [15], [38]. The synthetic views are then transferred to a second, viewpoint-dependent recognition system. This paradigm has been evaluated with 10 face recognition systems in the Face Recognition Vendor Test 2002 [30]: For 9 out of 10 systems, our morphable model and fitting procedure (Sections 3 and 4) improved performance on nonfrontal faces substantially.

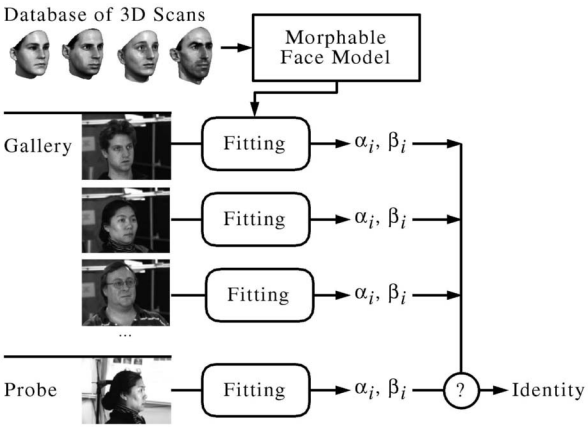


Fig. 1. Derived from a database of laser scans, the 3D morphable face model is used to encode gallery and probe images. For identification, the model coefficients α_i, β_i of the probe image are compared with the stored coefficients of all gallery images.

In many applications, synthetic views have to meet standard imaging conditions, which may be defined by the properties of the recognition algorithm, by the way the gallery images are taken (mug shots), or by a fixed camera setup for probe images. Standard conditions can be estimated from an example image by our system (Fig. 2). If more than one image is required for the second system or no standard conditions are defined, it may be useful to synthesize a set of different views of each person.

3 A Morphable Model of 3D Faces

The morphable face model is based on a vector space representation of faces [36] that is constructed such that any convex combination[1] of shape and texture vectors $\mathbf{S}_i$ and $\mathbf{T}_i$ of a set of examples describes a realistic human face:

$$\mathbf{S} = \sum_{i=1}^{m} a_i \mathbf{S}_i, \quad \mathbf{T} = \sum_{i=1}^{m} b_i \mathbf{T}_i. \tag{1}$$

Continuous changes in the model parameters a_i generate a smooth transition such that each point of the initial surface moves toward a point on the final surface. Just as in morphing, artifacts in intermediate states of the morph are avoided only if the initial and final points are corresponding structures in the face, such as the tip of the nose. Therefore, dense point-to-point correspondence is crucial for defining shape and texture vectors. We describe an automated method to establish this correspondence in Section 3.2, and give a definition of $\mathbf{S}$ and $\mathbf{T}$ in Section 3.3.

3.1 Database of Three-Dimensional Laser Scans

The morphable model was derived from 3D scans of 100 males and 100 females, aged between 18 and 45 years. One person is Asian, all others are Caucasian. Applied to image databases that cover a much larger ethnic variety

1. To avoid changes in overall size and brightness, a_i and b_i should sum to 1. The additional constraints $a_i, b_i \in [0,1]$ imposed on convex combinations will be replaced by a probabilistic criterion in Section 3.4.

Fig. 2. In 3D model fitting, light direction and intensity are estimated automatically, and cast shadows are taken into account. The figure shows original PIE images (top), reconstructions rendered into the originals (second row), and the same reconstructions rendered with standard illumination (third row) taken from the top right image.

(Section 5), the model seemed to generalize well beyond ethnic boundaries. Still, a more diverse set of examples would certainly improve performance.

Recorded with a $Cyberware^{TM}$ 3030PS laser scanner, the scans represent face shape in cylindrical coordinates relative to a vertical axis centered with respect to the head. In 512 angular steps ϕ covering 360° and 512 vertical steps h at a spacing of 0.615mm, the device measures radius r, along with red, green, and blue components of surface texture R, G, B. We combine radius and texture data:

$$\begin{aligned}&\mathbf{I}(h,\phi) = (r(h,\phi), R(h,\phi), G(h,\phi), B(h,\phi))^T,\\ &h,\phi \in \{0,\ldots,511\}.\end{aligned} \tag{2}$$

Preprocessing of raw scans involves:

1. filling holes and removing spikes in the surface with an interactive tool,
2. automated 3D alignment of the faces with the method of 3D-3D Absolute Orientation [19],
3. semiautomatic trimming along the edge of a bathing cap, and
4. a vertical, planar cut behind the ears and a horizontal cut at the neck, to remove the back of the head, and the shoulders.

3.2 Correspondence Based on Optic Flow

The core step of building a morphable face model is to establish dense point-to-point correspondence between each face and a reference face. The representation in cylindrical coordinates provides a parameterization of the two-dimensional manifold of the facial surface by parameters h and ϕ. Correspondence is given by a dense vector field $\mathbf{v}(h,\phi) = (\Delta h(h,\phi), \Delta\phi(h,\phi))^T$ such that each point $\mathbf{I}_1(h,\phi)$ on the first scan corresponds to the point $\mathbf{I}_2(h + \Delta h, \phi + \Delta\phi)$ on the second scan. We employ a modified optic flow algorithm to determine this vector field. The following two sections describe the original algorithm and our modifications.

Optic Flow on Gray-Level Images. Many optic flow algorithms (e.g., [20], [25], [2]) are based on the assumption that objects in image sequences $I(x,y,t)$ retain their brightnesses as they move across the image at a velocity $(v_x, v_y)^T$. This implies

$$\frac{dI}{dt} = v_x\frac{\partial I}{\partial x} + v_y\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0. \tag{3}$$

For pairs of images I_1, I_2 taken at two discrete moments, temporal derivatives v_x, v_y, $\frac{\partial I}{\partial t}$ in (3) are approximated by finite differences Δx, Δy, and $\Delta I = I_2 - I_1$. If the images are not from a temporal sequence, but show two different objects, corresponding points can no longer be assumed to have equal brightnesses. Still, optic flow algorithms may be applied successfully.

A unique solution for both components of $\mathbf{v} = (v_x, v_y)^T$ from (3) can be obtained if $\mathbf{v}$ is assumed to be constant on each neighborhood $R(x_0, y_0)$, and the following expression [25], [2] is minimized in each point (x_0, y_0):

$$E(x_0,y_0) = \sum_{x,y\in R(x_0,y_0)} \left(v_x\frac{\partial I(x,y)}{\partial x} + v_y\frac{\partial I(x,y)}{\partial y} + \Delta I(x,y)\right)^2. \tag{4}$$

We use a 5×5 pixel neighborhood $R(x_0, y_0)$. In each point (x_0, y_0), $\mathbf{v}(x_0, y_0)$ can be found by solving a 2×2 linear system (Appendix A).

In order to deal with large displacements $\mathbf{v}$, the algorithm of Bergen and Hingorani [2] employs a coarse-to-fine strategy using Gaussian pyramids of downsampled images: With the gradient-based method described above, the algorithm computes the flow field on the lowest level of resolution and refines it on each subsequent level.

Generalization to three-dimensional surfaces. For processing 3D laser scans $\mathbf{I}(h,\phi)$, (4) is replaced by

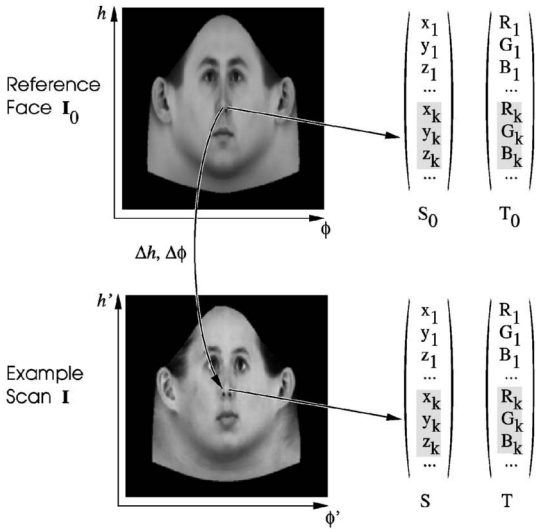


Fig. 3. For 3D laser scans parameterized by cylindrical coordinates (h, ϕ), the flow field that maps each point of the reference face (top) to the corresponding point of the example (bottom) is used to form shape and texture vectors $\mathbf{S}$ and $\mathbf{T}$.

$$E = \sum_{h,\phi \in R} \left\| v_h \frac{\partial \mathbf{I}(\mathbf{h},\phi)}{\partial h} + v_\phi \frac{\partial \mathbf{I}(\mathbf{h},\phi)}{\partial \phi} + \Delta \mathbf{I} \right\|^2, \quad (5)$$

$$\text{with a norm } \|\mathbf{I}\|^2 = w_r r^2 + w_R R^2 + w_G G^2 + w_B B^2. \quad (6)$$

Weights w_r, w_R, w_G, and w_B compensate for different variations within the radius data and the red, green, and blue texture components, and control the overall weighting of shape versus texture information. The weights are chosen heuristically. The minimum of (5) is again given by a 2×2 linear system (Appendix A).

Correspondence between scans of different individuals, who may differ in overall brightness and size, is improved by using Laplacian pyramids (band-pass filtering) rather than Gaussian pyramids (low-pass filtering). Additional quantities, such as Gaussian curvature, mean curvature, or surface normals, may be incorporated in $\mathbf{I}(h, \phi)$. To obtain reliable results even in regions of the face with no salient structures, a specifically designed smoothing and interpolation algorithm (Appendix A.1) is added to the matching procedure on each level of resolution.

3.3 Definition of Face Vectors

The definition of shape and texture vectors is based on a reference face $\mathbf{I}_0$, which can be any three-dimensional face model. Our reference face is a triangular mesh with 75,972 vertices derived from a laser scan. Let the vertices $k \in \{1, \ldots, n\}$ of this mesh be located at $(h_k, \phi_k, r(h_k, \phi_k))$ in cylindrical and at (x_k, y_k, z_k) in Cartesian coordinates and have colors (R_k, G_k, B_k). Reference shape and texture vectors are then defined by

$$\mathbf{S}_0 = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T, \quad (7)$$

$$\mathbf{T}_0 = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \quad (8)$$

To encode a novel scan $\mathbf{I}$ (Fig. 3, bottom), we compute the flow field from $\mathbf{I}_0$ to $\mathbf{I}$, and convert $\mathbf{I}(h', \phi')$ to Cartesian coordinates $x(h', \phi')$, $y(h', \phi')$, $z(h', \phi')$. Coordinates (x_k, y_k, z_k) and color values (R_k, G_k, B_k) for the shape and texture vectors $\mathbf{S}$ and $\mathbf{T}$ are then sampled at $h'_k = h_k + \Delta h(h_k, \phi_k)$, $\phi'_k = \phi_k + v_\phi(h_k, \phi_k)$.

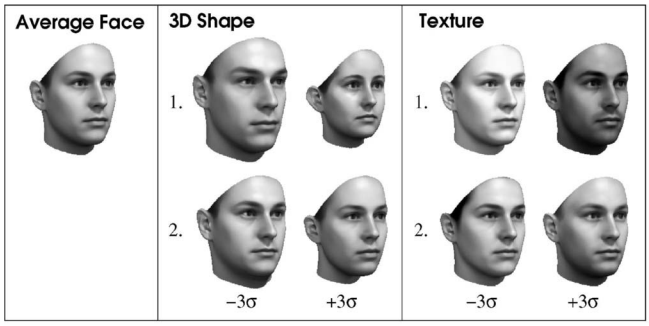


Fig. 4. The average and the first two principal components of a data set of 200 3D face scans, visualized by adding $\pm 3\sigma_{S,i}\mathbf{s}_i$ and $\pm 3\sigma_{T,i}\mathbf{t}_i$ to the average face.

3.4 Principal Component Analysis

We perform a Principal Component Analysis (PCA, see [12]) on the set of shape and texture vectors $\mathbf{S}_i$ and $\mathbf{T}_i$ of example faces $i = 1 \ldots m$. Ignoring the correlation between shape and texture data, we analyze shape and texture separately.

For shape, we subtract the average $\overline{\mathbf{s}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{S}_i$ from each shape vector, $\mathbf{a}_i = \mathbf{S}_i - \overline{\mathbf{s}}$, and define a data matrix $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m)$.

The essential step of PCA is to compute the eigenvectors $\mathbf{s}_1, \mathbf{s}_2, \ldots$ of the covariance matrix $\mathbf{C} = \frac{1}{m}\mathbf{A}\mathbf{A}^T = \frac{1}{m}\sum_{i=1}^{m} \mathbf{a}_i \mathbf{a}_i^T$, which can be achieved by a Singular Value Decomposition [31] of $\mathbf{A}$. The eigenvalues of $\mathbf{C}$, $\sigma_{S,1}^2 \geq \sigma_{S,2}^2 \geq \ldots$, are the variances of the data along each eigenvector. By the same procedure, we obtain texture eigenvectors t_i and variances $\sigma_{T,i}^2$. Results are visualized in Fig. 4. The eigenvectors form an orthogonal basis,

$$\mathbf{S} = \overline{\mathbf{s}} + \sum_{i=1}^{m-1} \alpha_i \cdot \mathbf{s}_i, \quad \mathbf{T} = \overline{\mathbf{t}} + \sum_{i=1}^{m-1} \beta_i \cdot \mathbf{t}_i \quad (9)$$

and PCA provides an estimate of the probability density within face space:

$$p_S(\mathbf{S}) \sim e^{-\frac{1}{2}\sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2}}, \quad p_T(\mathbf{T}) \sim e^{-\frac{1}{2}\sum_i \frac{\beta_i^2}{\sigma_{T,i}^2}}. \quad (10)$$

3.5 Segments

From a given set of examples, a larger variety of different faces can be generated if linear combinations of shape and texture are formed separately for different regions of the face. In our system, these regions are the eyes, nose, mouth, and the surrounding area [8]. Once manually defined on the reference face, the segmentation applies to the entire morphable model.

For continuous transitions between the segments, we apply a modification of the image blending technique of [9]: x, y, z coordinates and colors R, G, B are stored in arrays $x(h, \phi)$, ... based on the mapping $i \mapsto (h_i, \phi_i)$ of the reference face. The blending technique interpolates x, y, z and R, G, B across an overlap in the (h, ϕ)-domain, which is large for low spatial frequencies and small for high frequencies.

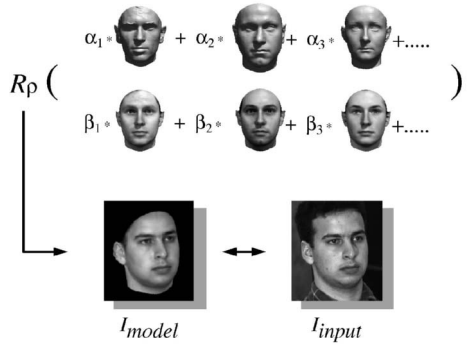


Fig. 5. The goal of the fitting process is to find shape and texture coefficients α_i and β_i describing a three-dimensional face model such that rendering R_ρ produces an image I_{model} that is as similar as possible to I_{input}.

4 Model-Based Image Analysis

The goal of model-based image analysis is to represent a novel face in an image by model coefficients α_i and β_i (9) and provide a reconstruction of 3D shape. Moreover, it automatically estimates all relevant parameters of the three-dimensional scene, such as pose, focal length of the camera, light intensity, color, and direction.

In an analysis-by-synthesis loop, the algorithm finds model parameters and scene parameters such that the model, rendered by computer graphics algorithms, produces an image as similar as possible to the input image $\mathbf{I}_{input}$ (Fig. 5).[2] The iterative optimization starts from the average face and standard rendering conditions (front view, frontal illumination, cf. Fig. 6).

For initialization, the system currently requires image coordinates of about seven facial feature points, such as the corners of the eyes or the tip of the nose (Fig. 6). With an interactive tool, the user defines these points $j = 1 \ldots 7$ by alternately clicking on a point of the reference head to select a vertex k_j of the morphable model and on the corresponding point $q_{x,j}, q_{y,j}$ in the image. Depending on what part of the face is visible in the image, different vertices k_j may be selected for each image. Some salient features in images, such as the contour line of the cheek, cannot be attributed to a single vertex of the model, but depend on the particular viewpoint and shape of the face. The user can define such points in the image and label them as contours. During the fitting procedure, the algorithm determines potential contour points of the 3D model based on the angle between surface normal and viewing direction and selects the closest contour point of the model as k_j in each iteration.

The following section summarizes the image synthesis from the model, and Section 4.2 describes the analysis-by-synthesis loop for parameter estimation.

4.1 Image Synthesis

The three-dimensional positions and the color values of the model's vertices are given by the coefficients α_i and β_i and (9). Rendering an image includes the following steps.

2. Fig. 5 is illustrated with linear combinations of example faces according to (1) rather than principal components (9) for visualization.

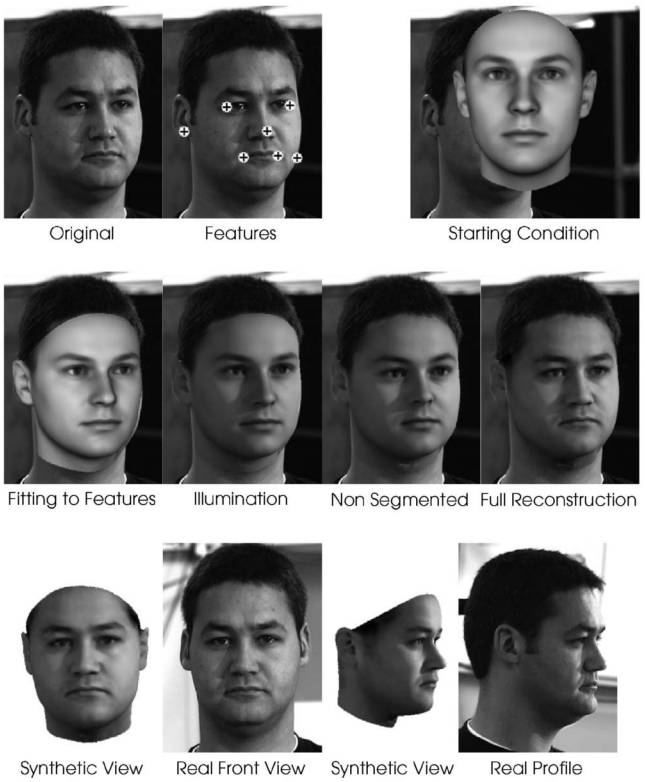


Fig. 6. Face reconstruction from a single image (top, left) and a set of feature points (top, center): Starting from standard pose and illumination (top, right), the algorithm computes a rigid transformation and a slight deformation to fit the features. Subsequently, illumination is estimated. Shape, texture, transformation, and illumination are then optimized for the entire face and refined for each segment (second row). From the reconstructed face, novel views can be generated (bottom row).

4.1.1 Image Positions of Vertices

A rigid transformation maps the object-centered coordinates $\mathbf{x}_k = (x_k, y_k, z_k)^T$ of each vertex k to a position relative to the camera:

$$(w_{x,k}, w_{y,k}, w_{z,k})^T = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \mathbf{x}_k + \mathbf{t}_w. \tag{11}$$

The angles ϕ and θ control in-depth rotations around the vertical and horizontal axis, and γ defines a rotation around the camera axis. $\mathbf{t}_w$ is a spatial shift.

A perspective projection then maps vertex k to image plane coordinates $p_{x,k}, p_{y,k}$:

$$p_{x,k} = P_x + f\frac{w_{x,k}}{w_{z,k}}, \qquad p_{y,k} = P_y - f\frac{w_{y,k}}{w_{z,k}}. \tag{12}$$

f is the focal length of the camera which is located in the origin, and (P_x, P_y) defines the image-plane position of the optical axis (principal point).

4.1.2 Illumination and Color

Shading of surfaces depends on the direction of the surface normals $\mathbf{n}$. The normal vector to a triangle $k_1k_2k_3$ of the face mesh is given by a vector product of the edges, $(\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) \times (\mathbf{x}_{k_1} - \mathbf{x}_{k_3})$, which is normalized to unit length and rotated along with the head (11). For fitting the model to an image, it is sufficient to consider the centers of triangles only, most of which are about 0.2mm^2 in size. The

three-dimensional coordinate and color of the center are the arithmetic means of the corners' values. In the following, we do not formally distinguish between triangle centers and vertices k.

The face is illuminated by ambient light with red, green, and blue intensities $L_{r,amb}$, $L_{g,amb}$, $L_{b,amb}$ and by directed, parallel light with intensities $L_{r,dir}$, $L_{g,dir}$, $L_{b,dir}$ from a direction $\mathbf{l}$ defined by two angles θ_l and ϕ_l:

$$\mathbf{l} = (\cos(\theta_l)\sin(\phi_l), \sin(\theta_l), \cos(\theta_l)\cos(\phi_l))^T. \quad (13)$$

The illumination model of Phong (see [14]) approximately describes the diffuse and specular reflection of a surface. In each vertex k, the red channel is

$$L_{r,k} = R_k \cdot L_{r,amb} + R_k \cdot L_{r,dir} \cdot \langle \mathbf{n}_k, \mathbf{l} \rangle + k_s \cdot L_{r,dir} \langle \mathbf{r}_k, \widehat{\mathbf{v}}_k \rangle^{\nu}, \quad (14)$$

where R_k is the red component of the diffuse reflection coefficient stored in the texture vector $\mathbf{T}$, k_s is the specular reflectance, ν defines the angular distribution of the specular reflections, $\widehat{\mathbf{v}}_k$ is the viewing direction, and $\mathbf{r}_k = 2 \cdot \langle \mathbf{n}_k, \mathbf{l} \rangle \mathbf{n}_k - \mathbf{l}$ is the direction of maximum specular reflection [14].

Input images may vary a lot with respect to the overall tone of color. In order to be able to handle a variety of color images as well as gray-level images and even paintings, we apply gains g_r, g_g, g_b, offsets o_r, o_g, o_b, and a color contrast c to each channel. The overall luminance L of a colored point is [14]

$$L = 0.3 \cdot L_r + 0.59 \cdot L_g + 0.11 \cdot L_b. \quad (15)$$

Color contrast interpolates between the original color value and this luminance, so, for the red channel, we set

$$I_r = g_r \cdot (cL_r + (1-c)L) + o_r. \quad (16)$$

Green and blue channels are computed in the same way. The colors I_r, I_g, and I_b are drawn at a position (p_x, p_y) in the final image $\mathbf{I}_{model}$.

Visibility of each point is tested with a z-buffer algorithm, and cast shadows are calculated with another z-buffer pass relative to the illumination direction (see, for example, [14].)

4.2 Fitting the Model to an Image

The fitting algorithm optimizes shape coefficients $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots)^T$ and texture coefficients $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots)^T$ along with 22 rendering parameters, concatenated into a vector $\boldsymbol{\rho}$: pose angles ϕ, θ, and γ, 3D translation $\mathbf{t}_w$, focal length f, ambient light intensities $L_{r,amb}, L_{g,amb}, L_{b,amb}$, directed light intensities $L_{r,dir}, L_{g,dir}, L_{b,dir}$, the angles θ_l and ϕ_l of the directed light, color contrast c, and gains and offsets of color channels g_r, g_g, g_b, o_r, o_g, o_b.

4.2.1 Cost Function

Given an input image

$$\mathbf{I}_{input}(x,y) = (I_r(x,y), I_g(x,y), I_b(x,y))^T,$$

the primary goal in analyzing a face is to minimize the sum of square differences over all color channels and all pixels between this image and the synthetic reconstruction,

$$E_I = \sum_{x,y} \left\| \mathbf{I}_{input}(x,y) - \mathbf{I}_{model}(x,y) \right\|^2. \quad (17)$$

The first iterations exploit the manually defined feature points $(q_{x,j}, q_{y,j})$ and the positions (p_{x,k_j}, p_{y,k_j}) of the corresponding vertices k_j in an additional function

$$E_F = \sum_j \left\| \begin{pmatrix} q_{x,j} \\ q_{y,i} \end{pmatrix} - \begin{pmatrix} p_{x,k_j} \\ p_{y,k_j} \end{pmatrix} \right\|^2. \quad (18)$$

Minimization of these functions with respect to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\rho}$ may cause overfitting effects similar to those observed in regression problems (see, for example, [12]). We therefore employ a maximum a posteriori estimator (MAP): Given the input image $\mathbf{I}_{input}$ and the feature points F, the task is to find model parameters with maximum posterior probability $p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} \mid \mathbf{I}_{input}, F)$. According to Bayes rule,

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} \mid \mathbf{I}_{input}, F) \sim p(\mathbf{I}_{input}, F \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \cdot P(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}). \quad (19)$$

If we neglect correlations between some of the variables, the right-hand side is

$$p(\mathbf{I}_{input} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \cdot p(F \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \cdot P(\boldsymbol{\alpha}) \cdot P(\boldsymbol{\beta}) \cdot P(\boldsymbol{\rho}). \quad (20)$$

The prior probabilities $P(\boldsymbol{\alpha})$ and $P(\boldsymbol{\beta})$ were estimated with PCA (10). We assume that $P(\boldsymbol{\rho})$ is a normal distribution and use the starting values for $\overline{\boldsymbol{\rho}}_i$ and ad hoc values for $\sigma_{R,i}$.

For Gaussian pixel noise with a standard deviation σ_I, the likelihood of observing $\mathbf{I}_{input}$, given $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}$, is a product of one-dimensional normal distributions, with one distribution for each pixel and each color channel. This can be rewritten as $p(\mathbf{I}_{input} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \sim exp(\frac{-1}{2\sigma_I^2} \cdot E_I)$. In the same way, feature point coordinates may be subject to noise, so $p(F \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \sim exp(\frac{-1}{2\sigma_F^2} \cdot E_F)$.

Posterior probability is then maximized by minimizing

$$E = -2 \cdot log\, p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} \mid \mathbf{I}_{input}, F)$$
$$E = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_F^2} E_F + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho}_i)^2}{\sigma_{R,i}^2}. \quad (21)$$

Ad hoc choices of σ_I and σ_F are used to control the relative weights of E_I, E_F, and the prior probability terms in (21). At the beginning, prior probability and E_F are weighted high. The final iterations put more weight on E_I and no longer rely on E_F.

4.2.2 Optimization Procedure

The core of the fitting procedure is a minimization of the cost function (21) with a stochastic version of Newton's method (Appendix B). The stochastic optimization avoids local minima by searching a larger portion of parameter space and reduces computation time: In E_I, contributions of the pixels of the entire image would be redundant. Therefore, the algorithm selects a set $\mathcal{K}$ of 40 random triangles in each iteration and evaluates E_I and its gradient only at their centers:

$$E_{I,approx.} = \sum_{k \in \mathcal{K}} \|\mathbf{I}_{input}(p_{x,k}, p_{y,k}) - \mathbf{I}_{model,k})\|^2. \quad (22)$$

To make the expectation value of $E_{I,approx.}$ equal to E_I, we set the probability of selecting a particular triangle proportional to its area in the image. Areas are calculated along with occlusions and cast shadows at the beginning of the process and once every 1,000 iterations by rendering the entire face model.

The fitting algorithm computes the gradient of the cost function (21), (22) analytically using chain rule. Texture coefficients β_i and illumination parameters only influence the color values $\mathbf{I}_{model,k}$ of a vertex. Shape coefficients α_i and rigid transformation, however, influence both the image coordinates $(p_{x,k}, p_{y,k})$ and color values $\mathbf{I}_{model,k}$ due to the effect of geometry on surface normals and shading (14).

The first iterations only optimize the first parameters $\alpha_i, \beta_i, i \in \{1, \ldots, 10\}$ and all parameters ρ_i. Subsequent iterations consider more and more coefficients. From the principal components of a database of 200 faces, we only use the most relevant 99 coefficients α_i, β_i. After fitting the entire face model to the image, the eyes, nose, mouth, and the surrounding region (Section 3.5) are optimized separately. The fitting process takes 4.5 minutes on a workstation with a 2GHz Pentium 4 processor.

5 Results

Model fitting and identification were tested on two publicly available databases of images. The individuals in these databases are not contained in the set of 3D scans that form the morphable face model (Section 3.1).

The colored images in the PIE database from CMU [33] vary in pose and illumination. We selected the portion of this database where each of 68 individuals is photographed from three viewpoints (front, side, and profile, labeled as camera 27, 05, 22) and at 22 different illuminations (66 images per individual). Illuminations include flashes from different directions and one condition with ambient light only.

From the gray-level images of the FERET database [29], we selected a portion that contains 11 poses (labeled *ba* – *bk*) per individual. We discarded pose *bj*, where participants have various facial expressions. The remaining 10 views, most of them at a neutral expression, are available for 194 individuals (labeled 01013 – 01206). While illumination in images *ba* – *bj* is fixed, *bk* is recorded at a different illumination.

Both databases cover a wide ethnic variety. Some of the faces are partially occluded by hair and some individuals wear glasses (28 in the CMU-PIE database, none in the FERET database.) We do not explicitly compensate for these effects. Optimizing the overall appearance, the algorithm tends to ignore image structures that are not represented by the morphable model.

5.1 Results of Model Fitting

The reconstruction algorithm was run on all 4,488 PIE and 1,940 FERET images. For all images, the starting condition was the average face at a front view, with frontal illumination, rendered in color from a viewing distance of two meters (Fig. 6).

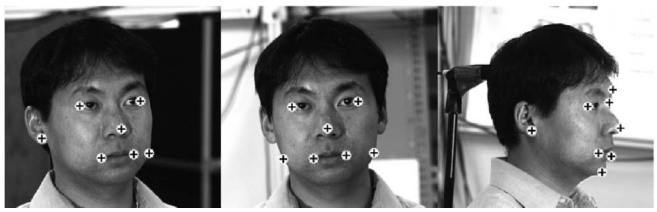


Fig. 7. Up to seven feature points were manually labeled in front and side views, up to eight were labeled in profile views.

On each image, we manually defined between six and eight feature points (Fig. 7). For each viewing direction, there was a standard set of feature points, such as the corners of the eyes, the tip of the nose, corners of the mouth, ears, and up to three points on the contour (cheeks, chin, and forehead). If any of these were not visible in an image, the fitting algorithm was provided with fewer point coordinates.

Results of 3D face reconstruction are shown in Figs. 8 and 9. The algorithm had to cope with a large variety of illuminations. In the third column of Fig. 9, part of the specular reflections were attributed to texture by the algorithm. This may be due to shortcomings of the Phong illumination model for reflection at grazing angles or to a prior probability that penalizes illumination from behind too much.

The influence of different illuminations is shown in a comparison in Fig. 2. The fitting algorithm adapts to different illuminations, and we can generate standard images with fixed illumination from the reconstructions. In Fig. 2, the standard illumination conditions are the estimates obtained from a photograph (top right).

For each image, the fitting algorithm provides an estimate of pose angle. Heads in the CMU-PIE database are not fully aligned in space, but, since front, side, and profile images are taken simultaneously, the relative angles between views should be constant. Table 1 shows that the error of pose estimates is within a few degrees.

5.2 Recognition From Model Coefficients

For face recognition according to Paradigm 1 described in Section 2, we represent shape and texture by a set of coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{99})^T$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{99})^T$ for the entire face and one set $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ for each of the four segments of the face (Section 3.5). Rescaled according to the standard deviations $\sigma_{S,i}$, $\sigma_{T,i}$ of the 3D examples (Section 3.4), we combine all of these $5 \cdot 2 \cdot 99 = 990$ coefficients $\frac{\alpha_i}{\sigma_{S,i}}$, $\frac{\beta_i}{\sigma_{T,i}}$ to a vector $\mathbf{c} \in \mathbb{R}^{990}$.

Comparing two faces $\mathbf{c}_1$ and $\mathbf{c}_2$, we can use the sum of the Mahalanobis distances [12] of the segments' shapes and textures, $d_M = \|\mathbf{c}_1 - \mathbf{c}_2\|^2$. An alternative measure for similarity is the cosine of the angle between two vectors [6], [27]: $d_A = \frac{\langle \mathbf{c}_1, \mathbf{c}_2 \rangle}{\|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\|}$.

Another similarity measure that is evaluated in the following section takes into account variations of model

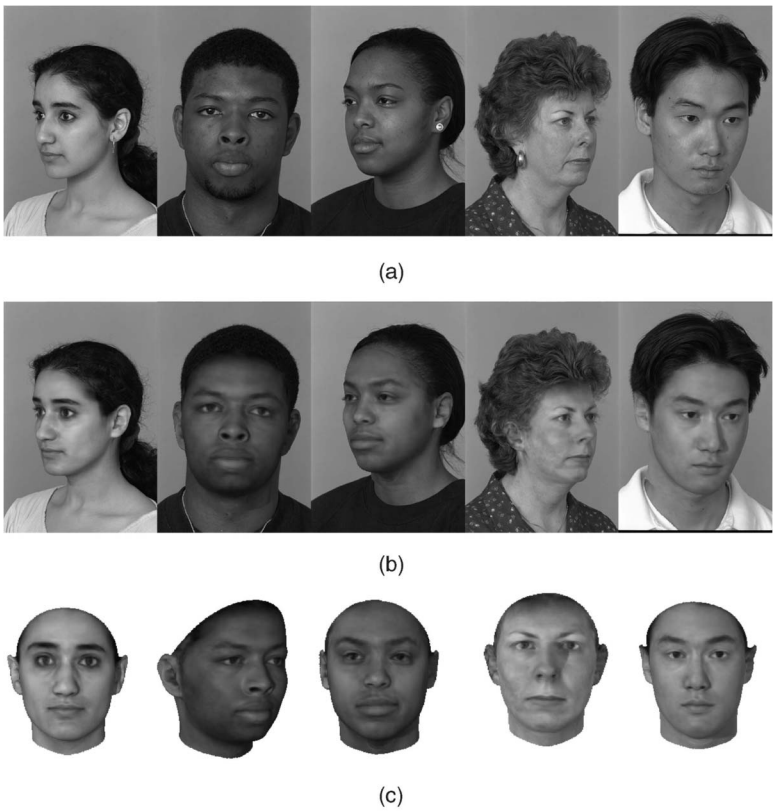


Fig. 8. Reconstructions of 3D shape and texture from FERET images (top row). In the second row, results are rendered into the original images with pose and illumination recovered by the algorithm. The third row shows novel views.

Fig. 9. Three-dimensional reconstructions from CMU-PIE images. Top: originals, middle: reconstructions rendered into originals, bottom: novel views. The pictures shown here are difficult due to harsh illumination, profile views, or eye glasses. Illumination in the third image is not fully recovered, so part of the reflections are attributed to texture.

coefficients obtained from different images of the same person. These variations may be due to ambiguities of the fitting problem, such as skin complexion versus intensity of illumination, and residual errors of optimization. Estimated from the CMU-PIE database, we apply these variations to the FERET images and vice versa, using a method motivated by Maximum-Likelihood Classifiers and Linear Discriminant Analysis (see [12]): Deviations of each persons' coefficients $\mathbf{c}$ from their individual average are pooled and analyzed by PCA. The covariance matrix $\mathbf{C}_W$ of this within-subject variation then defines

$$d_W = \frac{\langle \mathbf{c}_1, \mathbf{c}_2 \rangle_W}{\|\mathbf{c}_1\|_W \cdot \|\mathbf{c}_2\|_W}, \text{with } \langle \mathbf{c}_1, \mathbf{c}_2 \rangle_W = \langle \mathbf{c}_1, \mathbf{C}_W^{-1} \mathbf{c}_2 \rangle. \quad (23)$$

5.3 Recognition Performance

For evaluation on the CMU-PIE data set, we used a front, side, and profile gallery, respectively. Each gallery contained one view per person, at illumination number 13. The

TABLE 1
The Precision of Pose Estimates in Terms of the Rotation Angle between Two Views for Each Individual in the CMU-PIE Database

Angular distance	front-side	front-profile	side-profile
Average Estimate	18.1°	63.2°	45.2°
Standard Deviation	2.4°	4.6°	4.5°
True Angle	16.5°	62.1°	45.6°

Angles are a 3D combination of ϕ, θ, and γ. The table lists averages and standard deviations, based on 68 individuals, for illumination number 13. True angles are computed from the 3D coordinates provided with the database.

TABLE 2
Overall Percentage of Successful Identifications for Different Criteria of Comparing Faces

Database	d_M	d_A	d_W
CMU-PIE	87.2%	94.2%	**95.0%**
FERET	80.3%	92.2%	**95.9%**

For CMU-PIE images, data were computed for the side view gallery.

gallery for the FERET set was formed by one front view (pose *ba*) per person. The gallery and probe sets are always disjoint, but show the same individuals.

Table 2 provides a comparison of d_M, d_A, and d_W for identification (Section 2). d_W is clearly superior to d_M and d_A. All subsequent data are therefore based on d_W. The higher performance of angular measures (d_W and d_A) compared to d_M indicates that directions of coefficient vectors $\mathbf{c}$, relative to the average face $\mathbf{c} = 0$, are diagnostic for faces, while distances from the average may vary, causing variations in d_M. In our MAP approach, this may be due to the trade off between likelihood and prior probability ((19) and (21)): Depending on image quality, this may produce distinctive or conservative estimates.

A detailed comparison of different probe and gallery views for the PIE database is given in Table 3. In an identification task, performance is measured on probe sets of $68 \cdot 21$ images if probe and gallery viewpoint is equal (yet illumination differs; diagonal cells in the table) and $68 \cdot 22$ images otherwise (off-diagonal cells). Overall performance is best for the side-view gallery (95.0 percent correct). Table 4 lists the percentages of correct identifications on the FERET set, based on front view gallery images *ba*, along with the estimated head poses obtained from fitting. In total, identification was correct in 95.9 percent of the trials.

TABLE 3
Mean Percentages of Correct Identification on the CMU-PIE Data Set, Averaged over All Lighting Conditions for Front, Side, and Profile View Galleries

probe view	gallery view		
	front	side	profile
front	99.8% (97.1–100)	99.5% (94.1–100)	83.0% (72.1–94.1)
side	97.8% (82.4–100)	99.9% (98.5–100)	86.2% (61.8–95.6)
profile	79.5% (39.7–94.1)	85.7% (42.6–98.5)	98.3% (83.8–100)
total	92.3 %	95.0 %	89.0 %

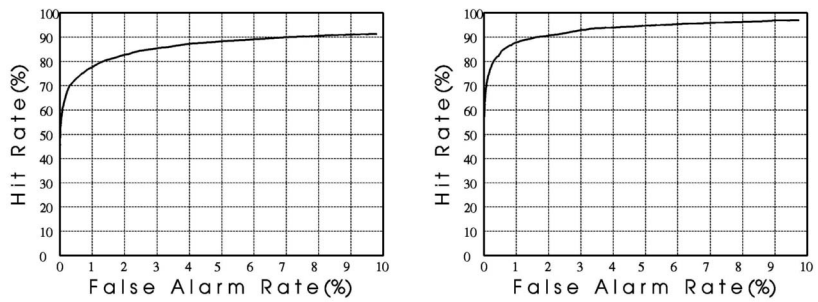
In brackets are percentages for the worst and best illumination within each probe set.

TABLE 4
Percentages of Correct Identification on the FERET Data Set

probe view	pose ϕ	correct identification
ba	1.1°	(gallery)
bb	38.9°	94.8%
bc	27.4°	95.4%
bd	18.9°	96.9%
be	11.2°	99.5%
bf	−7.1°	97.4%
bg	−16.3°	96.4%
bh	−26.5°	95.4%
bi	−37.9°	90.7%
bk	0.1°	96.9%
total		**95.9%**

The gallery images were front views ba. ϕ is the average estimated azimuth pose angle of the face. Ground truth for ϕ is not available. Condition bk has different illumination than the others.

Fig. 10 shows face recognition ROC curves [12] for a verification task (Section 2): Given pairs of images of the same person (one probe and one gallery image), *hit rate* is the percentage of correct verifications. Given pairs of images of different persons, *false alarm rate* is the percentage that is falsely accepted as the same person. For the CMU-PIE database, gallery images were side views (camera 05, light 13), the probe set was all 4,420 other images. For FERET, front views *ba* were gallery, and all other 1,746 images were probe images. At 1 percent false alarm rate, the hit rate is 77.5 percent for CMU-PIE and 87.9 percent for FERET.



(a) (b)

Fig. 10. ROC curves of verification across pose and illumination from a single side view for the CMU-PIE data set (a) and from a front view for FERET (b). At 1 percent false alarm rate, hit rate is 77.5 percent for CMU-PIE and 87.9 percent for FERET.

6 Conclusions

In this paper, we have addressed three issues: 1) learning class-specific information about human faces from a data set of examples, 2) estimating 3D shape and texture, along with all relevant 3D scene parameters, from a single image at any pose and illumination, and 3) representing and comparing faces for recognition tasks. Tested on two databases of images covering large variations in pose and illumination, our algorithm achieved promising results (95.0 and 95.9 percent correct identifications, respectively). This indicates that the 3D morphable model is a powerful and versatile representation for human faces. In image analysis, our explicit modeling of imaging parameters, such as head orientation and illumination, may help to achieve an invariant description of the identity of faces.

It is straightforward to extend our morphable model to different ages, ethnic groups, and facial expressions by including face vectors from more 3D scans. Our system currently ignores glasses, beards, or strands of hair covering part of the face, which are found in many images of the CMU-PIE and FERET sets. Considering these effects in the algorithm may improve 3D reconstructions and identification.

Future work will also concentrate on automated initialization and a faster fitting procedure. In applications that require a fully automated system, our algorithm may be combined with an additional feature detector. For applications where manual interaction is permissible, we have presented a complete image analysis system.

Appendix A
Optic Flow Calculation

Optic flow $\mathbf{v}$ between gray-level images at a given point (x_0, y_0) can be defined as the minimum $\mathbf{v}$ of a quadratic function (4). This minimum is given by [25], [2]

$$\mathbf{W}\mathbf{v} = -\mathbf{b} \tag{24}$$

$$\mathbf{W} = \begin{pmatrix} \sum \partial_x I^2 & \sum \partial_x I \cdot \partial_y I \\ \sum \partial_x I \cdot \partial_y I & \sum \partial_y I^2 \end{pmatrix},$$
$$\mathbf{b} = \begin{pmatrix} \sum \partial_x I \cdot \Delta I \\ \sum \partial_y I \cdot \Delta I \end{pmatrix}.$$

$\mathbf{v}$ is easy to find by means of a diagonalization of the 2×2 symmetrical matrix $\mathbf{W}$.

For 3D laser scans, the minimum of (5) is again given by (24), but now

$$\mathbf{W} = \begin{pmatrix} \sum \|\partial_h \mathbf{I}\|^2 & \sum \langle \partial_h \mathbf{I}, \partial_\phi \mathbf{I} \rangle \\ \sum \langle \partial_h \mathbf{I}, \partial_\phi \mathbf{I} \rangle & \sum \|\partial_\phi \mathbf{I}\|^2 \end{pmatrix},$$
$$\mathbf{b} = \begin{pmatrix} \sum \langle \partial_h \mathbf{I}, \Delta \mathbf{I} \rangle \\ \sum \langle \partial_\phi \mathbf{I}, \Delta \mathbf{I} \rangle \end{pmatrix}, \tag{25}$$

using the scalar product related to (6). $\mathbf{v}$ is found by diagonalizing $\mathbf{W}$.

A.1 Smoothing and Interpolation of Flow Fields

On regions of the face where both shape and texture are almost uniform, optic flow produces noisy and unreliable results. The desired flow field would be a smooth interpolation between the flow vectors of more reliable regions, such as the eyes and the mouth. We therefore apply a method that is motivated by a set of connected springs or a continuous membrane, that is fixed to reliable landmark points, sliding along reliably matched edges, and free to assume a minimum energy state everywhere else. Adjacent flow vectors of the smooth flow field $\mathbf{v}_s(h, \phi)$, are connected by a potential

$$E_c = \sum_h \sum_\phi \|\mathbf{v}_s(h+1, \phi) - \mathbf{v}_s(h, \phi)\|^2 + \sum_h \sum_\phi \|\mathbf{v}_s(h, \phi+1) - \mathbf{v}_s(h, \phi)\|^2. \tag{26}$$

The coupling of $\mathbf{v}_s(h, \phi)$ to the original flow field $\mathbf{v}_0(h, \phi)$ depends on the rank of the 2×2 matrix $\mathbf{W}$ in (25), which determines if (24) has a unique solution or not: Let $\lambda_1 \geq \lambda_2$ be the two eigenvalues of $\mathbf{W}$ and $\mathbf{a}_1$, $\mathbf{a}_2$ be the eigenvectors. Choosing a threshold $s \geq 0$, we set

$$E_0(h, \phi) = \begin{cases} 0 & \text{if} \quad \lambda_1, \lambda_2 \leq s \\ \langle \mathbf{a}_1, \mathbf{v}_s(\mathbf{h}, \phi) - \mathbf{v}_0(\mathbf{h}, \phi) \rangle^2 & \text{if} \quad \lambda_1 \geq s \geq \lambda_2 \\ \|\mathbf{v}_s(\mathbf{h}, \phi) - \mathbf{v}_0(\mathbf{h}, \phi)\|^2 & \text{if} \quad \lambda_1, \lambda_2 \geq s. \end{cases}$$

In the first case, which occurs if $\mathbf{W} \approx 0$ and $\partial_h \mathbf{I}, \partial_\phi \mathbf{I} \approx 0$ in R, the output $\mathbf{v}_s$ will only be controlled by its neighbors. The second case occurs if (24) restricts $\mathbf{v}_0$ only in one direction $\mathbf{a}_1$. This happens if there is a consistent edge structure within R, and the derivatives of $\mathbf{I}$ are linearly dependent in R. $\mathbf{v}_s$ is then free to slide along the edge. In the third case, $\mathbf{v}_0$ is uniquely defined by (24) and, therefore, $\mathbf{v}_s$ is restricted in all directions. To compute $\mathbf{v}_s$, we apply Conjugate Gradient Descent [31] to minimize the energy

$$E = \eta E_c + \sum_{h,\phi} E_0(h, \phi).$$

Both the weight factor η and the threshold s are chosen heuristically. During optimization, flow vectors from reliable, high-contrast regions propagate to low-contrast regions, producing a smooth interpolation. Smoothing is performed at each level of resolution after the gradient-based estimation of correspondence.

Appendix B
Stochastic Newton Algorithm

For the optimization of the cost function (21), we developed a stochastic version of Newton's algorithm [5] similar to stochastic gradient descent [32], [37], [22]. In each iteration, the algorithm computes E_I only at 40 random surface points (Section 4.2). The first derivatives of E_I are computed analytically on these random points.

Newton's method optimizes a cost function E with respect to parameters α_j based on the gradient ∇E and the Hessian $\mathbf{H}$, $H_{i,j} = \frac{\partial^2 E}{\partial \alpha_i \partial \alpha_j}$. The optimum is

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha} - \mathbf{H}^{-1} \nabla E. \tag{27}$$

For simplification, we consider α_i as a general set of model parameters here and suppress β, ρ. Equation (21) is then

$$E(\boldsymbol{\alpha}) = \frac{1}{\sigma_I^2} E_I(\boldsymbol{\alpha}) + \frac{1}{\sigma_F^2} E_F(\boldsymbol{\alpha}) + \sum_i \frac{(\alpha_i - \overline{\alpha}_i)^2}{\sigma_{S,i}^2} \tag{28}$$

and

$$\nabla E = \frac{1}{\sigma_I^2} \frac{\partial E_I}{\partial \alpha_i} + \frac{1}{\sigma_F^2} \frac{\partial E_F}{\partial \alpha_i} + diag\left(\frac{2}{\sigma_{S,i}^2}\right)(\boldsymbol{\alpha} - \overline{\boldsymbol{\alpha}}). \tag{29}$$

The diagonal elements of $\mathbf{H}$ are

$$H_{i,i} = \frac{1}{\sigma_I^2} \frac{\partial^2 E_I}{\partial \alpha_i^2} + \frac{1}{\sigma_F^2} \frac{\partial^2 E_F}{\partial \alpha_i^2} + \frac{2}{\sigma_{S,i}^2}. \tag{30}$$

These second derivatives are computed by numerical differentiation from the analytically calculated first derivatives, based on 300 random vertices, at the beginning of the optimization and once every 1,000 iterations. The Hessian captures information about an appropriate order of magnitude of updates in each coefficient. In the stochastic Newton algorithm, gradients are estimated from 40 points and the updates in each iteration do not need to be precise. We therefore ignore off-diagonal elements (see [5]) of $\mathbf{H}$ and set $\mathbf{H}^{-1} \approx diag(1/H_{i,i})$. With (27), the estimated optimum is

$$\alpha_i^* = \frac{\frac{1}{\sigma_I^2}\frac{\partial^2 E_I}{\partial \alpha_i^2}\alpha_i + \frac{1}{\sigma_F^2}\frac{\partial^2 E_F}{\partial \alpha_i^2}\alpha_i - \frac{1}{\sigma_I^2}\frac{\partial E_I}{\partial \alpha_i}\Big|_{\boldsymbol{\alpha}} - \frac{1}{\sigma_F^2}\frac{\partial E_F}{\partial \alpha_i}\Big|_{\boldsymbol{\alpha}} + \frac{2}{\sigma_{S,i}^2}\overline{\alpha}_i}{\frac{1}{\sigma_I^2}\frac{\partial^2 E_I}{\partial \alpha_i^2} + \frac{1}{\sigma_F^2}\frac{\partial^2 E_F}{\partial \alpha_i^2} + \frac{2}{\sigma_{S,i}^2}}. \tag{31}$$

In each iteration, we perform small steps $\boldsymbol{\alpha} \mapsto \boldsymbol{\alpha} + \lambda(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})$ with a factor $\lambda \ll 1$.

ACKNOWLEDGMENTS

The database of laser scans was recorded by N. Troje in the group of H. H. Bülthoff at MPI for Biological Cybernetics, Tübingen. Portions of the research in this paper use the FERET database of facial images collected under the FERET program, and the CMU-PIE database. The authors wish to thank everyone involved in collecting these data. The authors thank T. Poggio and S. Romdhani for many discussions and the reviewers for useful suggestions, including the title of the paper. This work was partially funded by the DARPA HumanID project.

REFERENCES

[1] J.J. Atick, P.A. Griffin, and A.N. Redlich, "Statistical Approach to Shape from Shading: Reconstruction of 3D Face Surfaces from Single 2D Images," *Computation in Neurological Systems,* vol. 7, no. 1, 1996.

[2] J.R. Bergen and R. Hingorani, "Hierarchical Motion-Based Frame Rate Conversion," technical report, David Sarnoff Research Center, Princeton N.J., 1990.

[3] D. Beymer and T. Poggio, "Face Recognition from One Model View," *Proc. Fifth Int'l Conf. Computer Vision,* 1995.

[4] D. Beymer and T. Poggio, "Image Representations for Visual Learning," *Science,* vol. 272, pp. 1905-1909, 1996.

[5] C.M. Bishop, *Neural Networks for Pattern Recognition.* Oxford Univ. Press, 1995.

[6] V. Blanz, "Automatische Rekonstruktion der dreidimensionalen Form von Gesichtern aus einem Einzelbild," PhD thesis, Tübingen, Germany, 2000.

[7] V. Blanz, S. Romdhani, and T. Vetter, "Face Identification across Different Poses and Illuminations with a 3D Morphable Model," *Proc. Fifth Int'l Conf. Automatic Face and Gesture Recognition,* pp. 202-207, 2002.

[8] V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," *Computer Graphics Proc. SIGGRAPH '99,* pp. 187-194, 1999.

[9] P.J. Burt and E.H. Adelson, "Merging Images through Pattern Decomposition," *Proc. Applications of Digital Image Processing VIII,* no. 575, pp. 173-181, 1985.

[10] C.S. Choi, T. Okazaki, H. Harashima, and T. Takebe, "A System of Analyzing and Synthesizing Facial Images," *Proc. IEEE Int'l Symp. Circuit and Systems (ISCAS '91),* pp. 2665-2668, 1991.

[11] T.F. Cootes, K. Walker, and C.J. Taylor, "View-Based Active Appearance Models," *Proc. Int'l Conf. Automatic Face and Gesture Recognition,* pp. 227-232, 2000.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. John Wiley & Sons, 2001.

[13] G.J. Edwards, T.F. Cootes, and C.J. Taylor, "Face Recogition Using Active Appearance Models," *Proc. Conf. Computer Vision (ECCV '98),* 1998.

[14] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice,* second ed. Addison-Wesley, 1996.

[15] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, 2001.

[16] D.B. Graham and N.M. Allison, "Face Recognition from Unfamiliar Views: Subspace Methods and Pose Dependency," *Proc. Int'l Conf. Automatic Face and Gesture Recognition,* pp. 348-353, 1998.

[17] R. Gross, I. Matthews, and S. Baker, "Eigen Light-Fields and Face Recognition Across Pose," *Proc. Int'l Conf. Automatic Face and Gesture Recognition,* pp. 3-9, 2002.

[18] P.W. Hallinan, "A Deformable Model for the Recognition of Human Faces under Arbitrary Illumination," PhD thesis, Harvard Univ., Cambridge, Mass., 1995.

[19] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision,* vol. 2. Addison-Wesley, 1992.

[20] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, pp. 185-203, 1981.

[21] T.S. Huang and L.A. Tang, "3D Face Modeling and Its Applications," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 10, no. 5, pp. 491-519, 1996.

[22] M. Jones and T. Poggio, "Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes," *Int'l J. Computer Vision,* vol. 29, no. 2, pp. 107-131, 1998.

[23] A. Lanitis, C.J. Taylor, and T.F. Cootes, "Automatic Face Identification System Using Flexible Appearance Models," *Image and Vision Computing,* vol. 13, no. 5, pp. 393-401, 1995.

[24] D.G. Lowe, "Fitting Parameterized Three-Dimensional Models to Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 5, pp. 441-450, May 1991.

[25] B.D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 674-679, 1981.

[26] T. Maurer and C. von der Malsburg, "Single-View Based Recognition of Faces Rotated in Depth," *Proc. Int'l Conf. Automatic Face and Gesture Recognition,* pp. 248-253, 1995.

[27] H. Moon and P.J. Phillips, "Computational and Performance Aspects of PCA-Based Face-Recognition Algorithms," *Perception,* vol. 30, pp. 303-321, 2001.

[28] A. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 84-91, 1994.

[29] P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The Feret Database and Evaluation Procedure for Face Recognition Algorithms," *Image and Vision Computing J.,* vol. 16, no. 5, pp. 295-306, 1998.

[30] P.J. Phillips, P. Grother, R.J. Michaels, D.M. Blackburn, E. Tabassi, and M. Bone, "Face Recognition Vendor Test 2002: Evaluation Report," NISTIR 6965, Nat'l Inst. of Standards and Technology, 2003.

[31] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes in C.* Cambridge Univ. Press, 1992.

[32] H. Robbins and S. Munroe, "A Stochastic Approximation Method," *Annals of Math. Statistics,* vol. 22, pp. 400-407, 1951.

[33] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) Database," *Proc. Int'l Conf. Automatic Face and Gesture Recognition,* pp. 53-58, 2002.
[34] T. Sim and T. Kanade, "Illuminating the Face," Technical Report CMU-RI-TR-01-31, The Robotics Inst., Carnegie Mellon Univ., Sept. 2001.
[35] T. Vetter and V. Blanz, "Estimating Coloured 3D Face Models from Fingle Images: An Example Based Approach," *Proc. Conf. Computer Vision (ECCV '98),* vol. II, 1998.
[36] T. Vetter and T. Poggio, "Linear Object Classes and Image Synthesis from a Single Example Image," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 733-742, July 1997.
[37] P. Viola, "Alignment by Maximization of Mutual Information," A.I. Memo No. 1548, MIT Artificial Intelligence Laboratory, 1995.
[38] W. Zhao and R. Chellappa, "SFS Based View Synthesis for Robust Face Recognition," *Proc. Int'l Conf. Automatic Face and Gesture Recognition,* pp. 285-292, 2000.
[39] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face Recognition: A Literature Survey," UMD CfAR Technical Report CAR-TR-948, 2000.

Volker Blanz received the diploma-degree from University of Tübingen, Germany, in 1995. He then worked on a project on multiclass support vector machines at AT&T Bell Labs in Holmdel, New Jersey. He received the PhD degree in physics from University of Tübingen in 2000 for his thesis on reconstructing 3D shape from images, written at Max-Planck-Institute for Biological Cybernetics, Tübingen. He was a visiting researcher at the Center for Biological and Computational Learning at MIT and a research assistant at the University of Freiburg. In 2003, he joined the Max-Planck-Institute for Computer Science, Saarbrücken, Germany. His research interests are in the fields of face recognition, machine learning, facial modeling, and animation.

Thomas Vetter studied mathematics and physics and received the PhD degree in biophysics from the University of Ulm, Germany. As a postdoctoral researcher at the Center for Biological and Computational Learning at MIT, he started his research on computer vision. In 1993, he moved to the Max-Planck-Institut in Tübingen and, in 1999, he became a professor of computer graphics at the University of Freiburg. Since 2002, he has been a professor of applied computer science at the University of Basel in Switzerland. His current research is on image understanding, graphics, and automated model building. He is a member of the IEEE and the IEEE Computer Society.