Face Recognition Based on 3D Shape Estimation from Single Images

Computer Graphics Technical Report No. 2

Volker Blanz, Thomas Vetter

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 4488 images from the publicly available CMU-PIE database, and 1940 images from the FERET database.

1 Introduction

Color values in an image of a face depend on head pose and illumination conditions even more than on the identity of the person who is depicted. Changes in pose and illumination are therefore the main challenges 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]).

A separate parameter for orientation is obtained by parameterizing for each individual the manifold formed by different views within the eigenspace of images [17], or by defining separate view-based eigenspaces [30]. Another way of capturing the viewpoint dependency is to represent faces by eigen-lightfields [18].

Two-dimensional face models represent grey values and their image locations independently [4, 5, 19, 24, 14, 23]. However, these models do not distinguish between rotation angle and shape, and only [19] separates illumination from texture. Since large rotations cannot be easily generated by the 2D warping used in these algorithms due to occlusions, multiple view-based 2D models have to be combined [36, 12]. Another approach that separates the image locations of facial features from their appearance uses an approximation of how each feature is deformed during rotations [27].

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 [22]. Others use restricted, but class-specific deformations, which can be defined manually [25], or learned from images [11], from non-textured [1] or textured 3D scans of heads [9].

In order to separate texture (albedo) from illumination conditions, some algorithms derived from shape-from-shading use models of illumination that explicitly consider illumination direction and intensity for Lambertian [16, 38] or non-Lambertian shading [35]. After analyzing images with shape-from-shading, some algorithms use a 3D head model to synthesize novel orientations [16, 38].

In this paper, we use a combination of deformable 3D models with a computer graphics simulation of illumination effects. This makes intrinsic shape and texture fully independent from extrinsic parameters [9, 8]. In our

framework, rotations in depth or changes of illumination are very simple operations, and specular reflections and cast shadows are easy to simulate.

Given a single image of a person, the algorithm automatically estimates 3D shape, texture and all relevant 3D scene parameters. The crucial element of our approach is a morphable model of 3D faces. The model represents shapes and textures of faces as vectors in a high dimensional face space, and estimates their probability density. This class-specific information is learned automatically from examples.

The automated parameter estimation includes focal length of the camera and illumination direction, which had to be chosen by the user in previous systems [9, 8]. A new optimization algorithm (Appendix B) and a novel initialization procedure based on image coordinates of between 6 and 8 feature points make the algorithm more robust and more reliable. 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 the images.

With a single model, we are able to compensate for variations both in pose and in illumination, using only a single image of a person for recognition. Our approach is not restricted to Lambertian reflection, but takes into account specular reflections, which have considerable influence on the appearance of human skin.

In the following section, we discuss different applications of 3D shape reconstruction in face recognition systems. Section 3 describes a method to derive a morphable face model from 3D scans. In Section 4, we present an algorithm for reconstructing 3D shape and recovering model parameters from images. Finally, we present results obtained with the image databases of CMU-PIE [34] and FERET [31].

2 3D Shape Reconstruction for Identification

The task of identification is to decide which individual from a *gallery* of given images is shown on a novel *probe* image (cf. [39]). In this paper, we consider galleries consisting of a single image for each individual. Fitting the 3D morphable model to images can be used in two ways for identification across different viewing conditions:

Paradigm 1: Identification can be based on the model coefficients, which represent intrinsic shape and texture of faces independent from the imaging conditions. Prior to identification, all gallery images are analyzed by the fitting algorithm, and the shape and texture coefficients are stored (Figure 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. We apply this paradigm in Section 5.

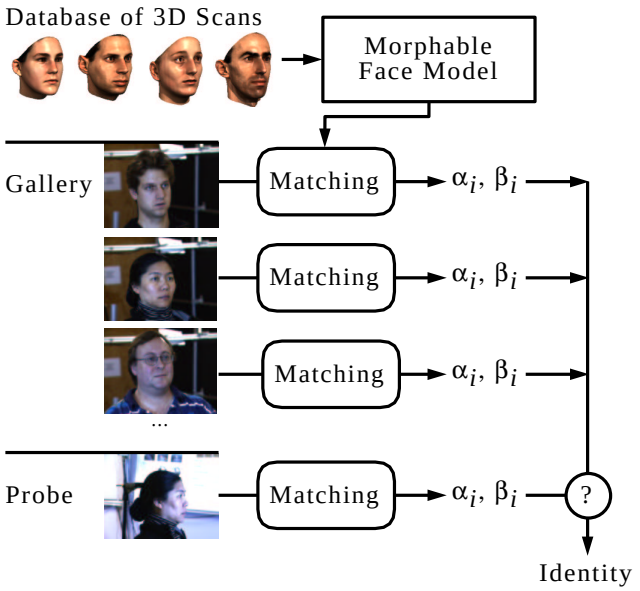

Figure 1: *Derived from a database of laser scans, the 3D morphable face model is used to encode gallery and probe images. For identification, the model coefficients α_i, β_i of the probe image are compared with the stored coefficients of all gallery images.*

Paradigm 2: 3D face reconstruction can also be employed to generate synthetic views from gallery or probe images for another recognition system. The second system may be view-dependent and rely on more than one image per person. Many applications involve standard imaging conditions defined by the view-dependent recognition algorithm, by the way the gallery images are taken (mug shots), or by a fixed camera setup for probe images. After estimating standard pose and illumination from an example image, we can generate standard views of each individual (Figure 9). Instead of one standard view, we can also synthesize a set of different views.

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 towards a point on the final surface. Just as in morphing [29], artefacts in intermediate states of the morph are avoided only if the initial and final points

[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.

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 (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 persons' heads. 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:

$$\mathbf{I}(h,\phi) = (r(h,\phi), R(h,\phi), G(h,\phi), B(h,\phi))^T,$$
$$h, \phi \in \{0, \ldots, 511\}.$$

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 [20], (3) semi-automatic trimming along the edge of a bathing cap, and (4) two planar cuts behind the ears and at the neck.

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 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 $\mathbf{I}_1(h,\phi)$ in the first scan corresponds to the point $\mathbf{I}_2(h+\Delta h,\ \phi+\Delta\phi)$ in 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 Grey Level Images: Many optic flow algorithms (e.g. [21, 26, 3]) are based on the assumption that objects in motion sequences $I(x,y,t)$ retain their brightnesses as they move across the image at a velocity $(v_x, v_y)^T$. This implies

$$\frac{dI}{dt} = v_x\frac{\partial I}{\partial x} + v_y\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \qquad (2)$$

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 Equation (2) 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 Equation (2) can be obtained if $\mathbf{v}$ is assumed to be constant on each neighborhood $R(x_0, y_0)$, and the following expression [26, 3] is minimized at each point (x_0, y_0):

$$E(x_0,y_0) = \qquad (3)$$
$$\sum_{x,y\in R(x_0,y_0)} (v_x\frac{\partial I(x,y)}{\partial x} + v_y\frac{\partial I(x,y)}{\partial y} + \Delta I(x,y))^2.$$

We used a 5x5 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 2x2 linear system (Appendix A).

In order to deal with large displacements $\mathbf{v}$, the algorithm of Bergen and Hingorani [3] employs a coarse-to-fine strategy using a Gaussian pyramid [2] 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 $\mathbf{I}(h,\phi)$, Equation (3) is replaced by

$$E = \sum_{h,\phi\in R} \left\| v_h\frac{\partial \mathbf{I}(h,\phi)}{\partial h} + v_\phi\frac{\partial \mathbf{I}(h,\phi)}{\partial \phi} + \Delta\mathbf{I} \right\|^2, \qquad (4)$$

with a norm

$$\|\mathbf{I}\|^2 = w_r r^2 + w_R R^2 + w_G G^2 + w_B B^2. \qquad (5)$$

Weights w_r, w_R, w_G, 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 Equation (4) is again given by a 2x2 linear system (Appendix A).

Additional quantities, such as Gaussian curvature, mean curvature, or the surface normal, may be incorporated in $\mathbf{I}(h,\phi)$ to improve results. 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.

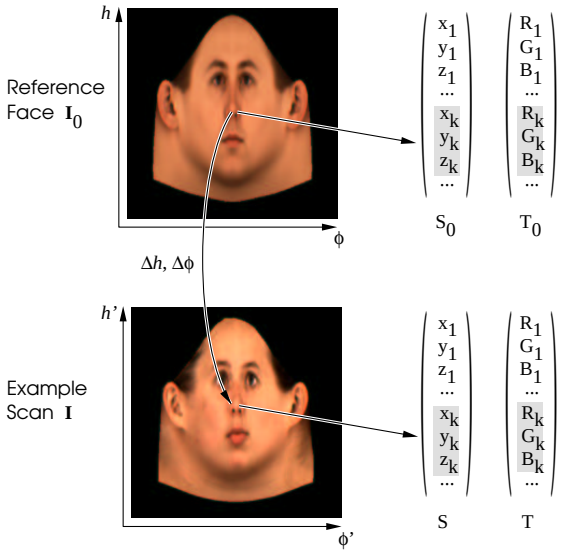

Figure 2: *For scans parameterized by cylindrical coordinates (h, ϕ), the flow field that maps each point of the reference face (top) to the corresponding point of the example (bottom) is used to form shape and texture vectors* $\mathbf{S}$ *and* $\mathbf{T}$.

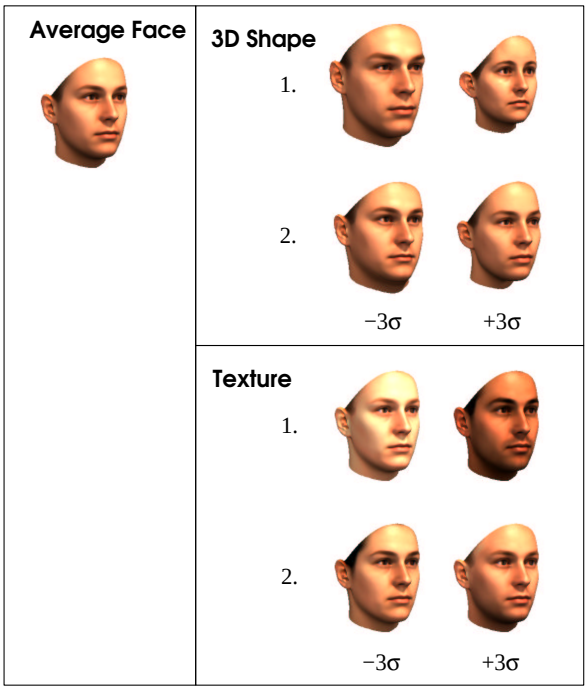

Figure 3: *The average and the first two principal components of a dataset of 200 3D face scans, visualized by adding* $\pm 3\sigma_{S,i}\mathbf{s}_i$ *and* $\pm 3\sigma_{T,i}\mathbf{t}_i$ *to the average face.*

3.3 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 75972 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

$$\mathbf{S}_0 = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T, \quad (6)$$

$$\mathbf{T}_0 = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \quad (7)$$

To encode a novel scan $\mathbf{I}$ (Figure 2, right), 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 $h'_k = h_k + \Delta h(h_k, \phi_k)$, $\quad \phi'_k = \phi_k + v_\phi(h_k, \phi_k)$.

3.4 Principal Component Analysis

We perform a Principal Component Analysis (PCA, see [13]) 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, shape and texture are analyzed separately.

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

The eigenvectors $\mathbf{s}_1, \mathbf{s}_2, \ldots$ of the covariance matrix $\mathbf{C} = \frac{1}{m}\mathbf{A}A^T = \frac{1}{m}\sum_{i=1}^{m} \mathbf{a}_i \mathbf{a}_i^T$ can be calculated by a Singular Value Decomposition [32] of $\mathbf{A}$. The eigenvalues of $\mathbf{C}$, $\sigma_{S,1}^2 \geq \sigma_{S,2}^2 \geq \ldots$, are the variances within the data set along each eigenvector.

By the same procedure, we obtain texture eigenvectors $\mathbf{t}_i$ and variances $\sigma_{T,i}^2$. Results are visualized in Figure 3. The eigenvectors form an orthogonal basis,

$$\mathbf{S} = \overline{\mathbf{s}} + \sum_{i=1}^{m-1} \alpha_i \cdot \mathbf{s}_i, \qquad \mathbf{T} = \overline{\mathbf{t}} + \sum_{i=1}^{m-1} \beta_i \cdot \mathbf{t}_i \quad (8)$$

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

$$p_S(\mathbf{S}) \sim e^{-\frac{1}{2}\sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2}}, \qquad p_T(\mathbf{T}) \sim e^{-\frac{1}{2}\sum_i \frac{\beta_i^2}{\sigma_{T,i}^2}}. \quad (9)$$

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 [9]. 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 [10]: 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.

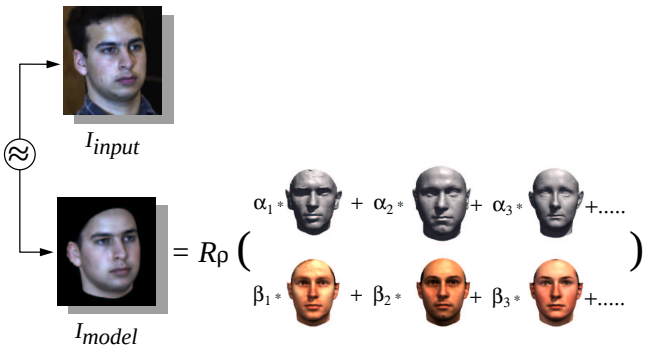

Figure 4: *The goal of the fitting process is to find shape and texture coefficients α_i and β_i describing a three-dimensional face model such that rendering R_ρ produces an image I_{model} that is as similar as possible to I_{input}.*

4 Model-Based Image Analysis

Model-based image analysis represents a novel face by model coefficients α_i and β_i (Equation 8), and provides 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 light 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 close as possible to the input image $\mathbf{I}_{input}$ (Figure 4).[2] The iterative optimization starts from the average face and standard rendering conditions (front view, frontal illumination, full color contrast, Figure 5).

For initialization, the system currently requires image coordinates of about 7 facial feature points, such as the corners of the eyes or the tip of the nose (Figure 5). 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 synthesis framework that creates an image from the model, and then discusses how the model parameters are estimated.

[2] Figure 4 is illustrated with linear combinations of example faces according to (1) rather than principal components (8) for visualization.

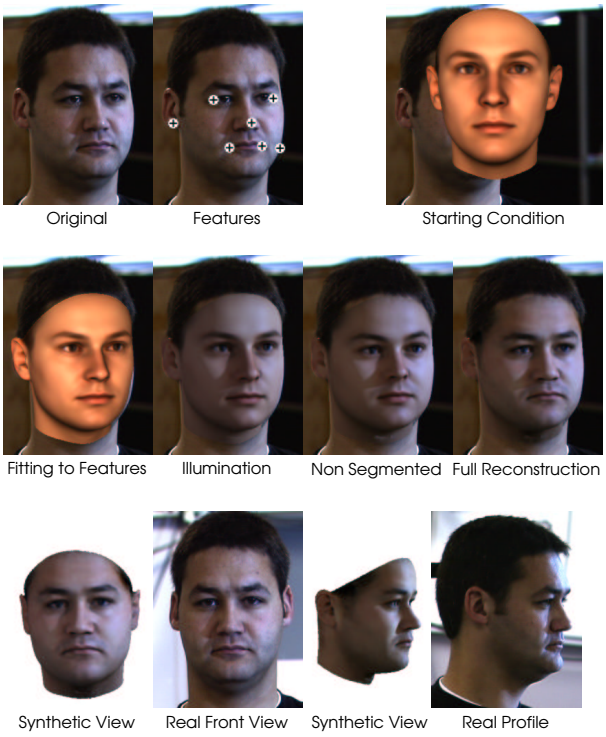

Figure 5: *The process of face reconstruction from a single image (top, left) and a set of feature point coordinates (top, center). Starting from standard pose and illumination (top, right), the algorithm computes a rigid transformation and a conservative deformation to fit the feature points. Then, illumination is estimated. In subsequent iterations, shape, texture, transformation and illumination are optimized for the entire model, and for each segment separately (second row). From the full reconstruction, novel views can be generated (bottom row).*

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 Equation (8). Rendering an image includes the following steps:

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:

$$(w_{x,k}\,,\, w_{y,k}\,,\, w_{z,k})^T = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \mathbf{x}_k + \mathbf{t}_w. \qquad (10)$$

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 $p_{x,k}$, $p_{y,k}$:

$$p_{x,k} = P_x + f\frac{w_{x,k}}{w_{z,k}}\,, \qquad p_{y,k} = P_y - f\frac{w_{y,k}}{w_{z,k}}\,. \qquad (11)$$

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 (Equation 10). For fitting the model to an image, it is sufficient to consider the centers of triangles only, most of which are about $0.2\text{m}m^2$ in size. 3D 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 defined by two angles θ_l and ϕ_l:

$$\mathbf{l} = (\cos(\theta_l)\sin(\phi_l),\ \sin(\theta_l),\ \cos(\theta_l)\cos(\phi_l))^T. \quad (12)$$

The illumination model of Phong (see [15]) approximately describes the diffuse and specular reflection on a surface. On each vertex k, the red channel is

$$L_{r,k} = R_k \cdot L_{r,amb} + R_k \cdot L_{r,dir} \cdot \langle \mathbf{n}_k, \mathbf{l} \rangle + k_s \cdot L_{r,dir} \langle \mathbf{r}_k, \widehat{\mathbf{v}}_k \rangle^\nu \quad (13)$$

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 [15].

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 grey 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 [15]

$$L = 0.3 \cdot L_r + 0.59 \cdot L_g + 0.11 \cdot L_b. \quad (14)$$

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

$$I_r = g_r(cL_r + (1-c)L) + o_r. \quad (15)$$

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 [15].)

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 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,

$$E_I = \sum_{x,y} \left\| \mathbf{I}_{input}(x,y) - \mathbf{I}_{model}(x,y) \right\|^2. \quad (16)$$

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

$$E_F = \sum_j \left\| \begin{pmatrix} q_{x,j} \\ q_{x,j} \end{pmatrix} - \begin{pmatrix} p_{x,k_j} \\ p_{y,k_j} \end{pmatrix} \right\|^2. \quad (17)$$

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 [13]). We therefore employ a maximum a posteriori estimator (MAP) derived from a Bayesian approach [9]. 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,

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} \mid \mathbf{I}_{input}, F) \sim p(\mathbf{I}_{input}, F \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \cdot P(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}). \quad (18)$$

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

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

The prior probabilities $P(\boldsymbol{\alpha})$ and $P(\boldsymbol{\beta})$ were estimated with PCA (Equation 9). For $P(\boldsymbol{\rho})$, we assume 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 $p(\mathbf{I}_{input} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \sim exp(\frac{-1}{2\sigma_I^2} \cdot E_I)$. In the same way, feature point coordinates may be subject to noise, and $p(F | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}) \sim exp(\frac{-1}{2\sigma_F^2} \cdot E_I)$.

Posterior probability is then maximized by minimizing

$$E = -2 \cdot log\, p\,(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho} \mid \mathbf{I}_{input}, F)$$
$$E = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_F^2} E_F + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho}_i)^2}{\sigma_{R,i}^2}. \quad (20)$$

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 (20). 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

For each iteration of the optimization process, the fitting algorithm analytically computes the gradient of the cost function (20) . The derivatives are summarized in Appendix C.

The optimization algorithm is a stochastic version of Newton's method (Appendix B, cf. [23]): Since contributions of the pixels of the entire image might be redundant, the algorithm selects 40 random triangles at each iteration, and evaluates E_I and its gradient only at their centers. This does not only speed up the optimization, but also avoids local minima by searching a larger portion of parameter space.

The random selection is implemented such that the probability of selecting a particular triangle is proportional to its area in the image. The expectation value of the approximate cost function is therefore equal to the full cost function (20). Areas of triangles are determined along with occlusions and cast shadows at the beginning of the process, and once every 1000 iterations, by rendering the entire face model.

The first iterations only optimize the first parameters $\alpha_i, \beta_i, i \in \{1, ..., 10\}$ 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 [34] vary in pose and illumination. We selected the portion of this database where each of 68 individuals is photographed from 3 viewpoints (front, side, and profile,

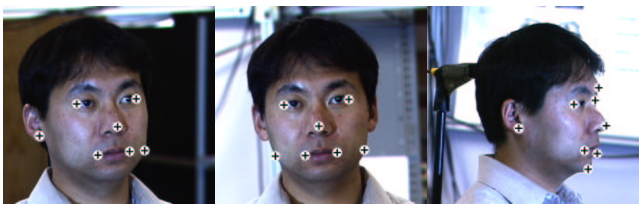

Figure 6: *Up to 7 features points were manually labeled in front and side views, up to 8 in profile views.*

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 grey-level images of the FERET database [31], 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 4488 PIE and 1940 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 2 meters (Figure 6).

On each image, we manually defined between 6 and 8 feature points (Figure 6). 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 3 points on the contour (cheeks, chin, and forehead). If any of these were not visible in an image, the fitting algorithm was provided with less point coordinates.

Results of 3D face reconstruction are shown in Figures 7 and 8. The algorithm had to cope with a large variety of illuminations. In the third column of Figure 8, 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 chosen such that illumination from behind is penalized.

The influence of different illuminations is shown in a comparison in Figure 9. The fitting algorithm adapts to different illuminations, and from the reconstructions,

Figure 7: *Reconstructions of 3D shape and texture from FERET images (top row). In the second row, results are rendered into the original image with pose and illumination recovered by the algorithm. The third row shows novel views.*

Angular distance	front to side	front to profile	side to 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°

Table 1: The precision of pose estimates in terms of the rotation angle between two views for each individual in the CMU-PIE database. 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.

standard images with fixed illumination can be generated. In Figure 9, 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 Identification From Model Coefficients

For identification 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 $\mathbf{c} \in I\!R^{990}$.

Comparing two faces $\mathbf{c}_1$ and $\mathbf{c}_2$, we might use the sum of Mahalanobis distances [13] of the segments' shapes and textures, $d_M = \|\mathbf{c}_1 - \mathbf{c}_1\|^2$. However, recognition performance is higher if we use the cosine of the angle between two vectors [7, 28]: $d_A = \dfrac{\langle \mathbf{c}_1, \mathbf{c}_2 \rangle}{\|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\|}$.

Model coefficients recovered from different images of the same person are affected by a number of sources of variation: Parameters of the fitting problem may be ambiguous, such as skin complexion versus intensity of illumination, illumination effects are not fully captured by our lighting model, and optimization may have residual errors. 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 [13]): 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

$$d_W = \frac{\langle \mathbf{c}_1, \mathbf{c}_2 \rangle_W}{\|\mathbf{c}_1\|_W \cdot \|\mathbf{c}_2\|_W}, \tag{21}$$

with $\langle \mathbf{c}_1, \mathbf{c}_2 \rangle_W = \left\langle \mathbf{c}_1, \mathbf{C}_W^{-1} \mathbf{c}_2 \right\rangle$.

Figure 8: *3D reconstructions from CMU-PIE images. Top: originals, middle: reconstructions rendered into original, 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.*

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

Table 2: Overall percentage of successful identifications for different criteria of comparing faces. For CMU-PIE images, data were computed for the side view gallery.

5.3 Recognition Performance

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

A detailed comparison of different probe and gallery views for the PIE database is given in Table 4. 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.

Table 3 lists the percentages of correct identifications on the FERET set, based on front view gallery images *ba*, along with the estimates of head pose obtained from fitting. Figure 10 shows face recognition ROC curves [13]. For the CMU-PIE database, gallery images were side views (camera 05, light 13), the probe set were all 4420 other images. For FERET, front views *ba* were gallery, and all other 1746 images were probe images.

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%**

Table 3: Mean identification percentages on the FERET dataset. 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.

6 Conclusions

In this paper, we have addressed three issues: (1) Learning class-specific information about human faces from a dataset 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.

Figure 9: *In 3D model fitting, light direction and intensity is estimated automatically, and cast shadows are taken into account. The figure shows original PIE images (top), reconstructions rendered into the image (second row), and the same reconstructions rendered with standard illumination (third row) taken from the top right image.*

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 %

Table 4: Mean identification percentages on the CMU–PIE dataset, averaged over all lighting conditions for front, side and profile view galleries. In brackets are percentages for the worst and best illumination within each probe set.

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.

Acknowledgements The database of laser scans was recorded by N. Troje and T. Philipps in the group of H. H. Bülthoff at Max-Planck-Institute for Biol. Cybernetics, Tübingen. Portions of the research in this paper use the FERET database of facial images collected under the FERET program. The authors wish to thank T. Poggio and S. Romdhani for many comments and discussions. This study 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 Neur. Syst.*, 7:1, 1996.

[2] K. D. Baker and G. D. Sullivan. Multiple bandpass filters in image processing. *Proc. IEEE*, 127:173–184, 1980.

[3] J.R. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center Princeton NJ 08540, 1990.

[4] D. Beymer and T. Poggio. Face recognition from one model view. In *Proceedings of the 5th International Conference on Computer Vision*, 1995.

[5] D. Beymer and T. Poggio. Image representation for visual learning. *Science*, 272:1905–1909, 1996.

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

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

[8] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Proc. of the 5th Int. Conf. on Automatic Face and Gesture Recognition*, pages 202–207, 2002.

[9] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Computer Graphics Proc. SIGGRAPH'99*, pages 187–194, Los Angeles, 1999.

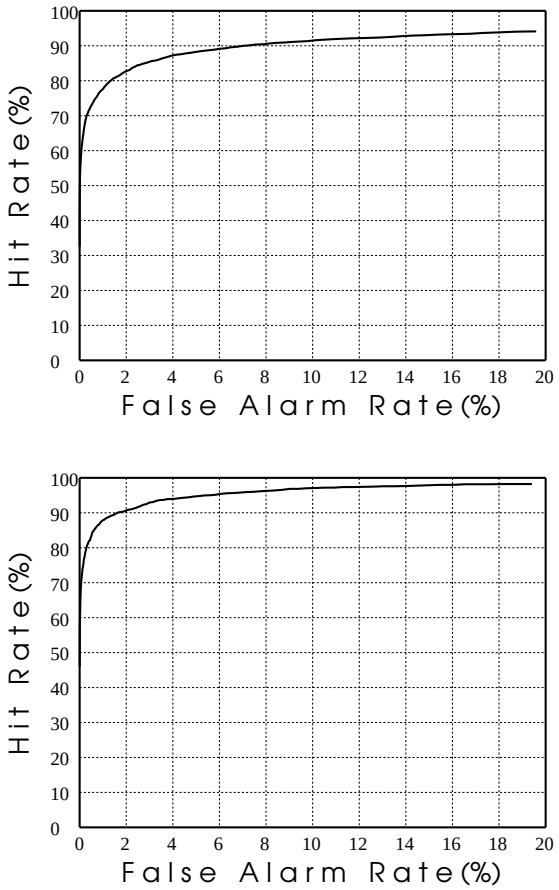

Figure 10: *ROC curves of recognition across pose and illumination from a single side view for the CMU-PIE dataset (top), and from a front view for FERET (bottom).*

[10] P.J. Burt and E.H. Adelson. Merging images through pattern decomposition. In *Applications of Digital Image Processing VIII*, number 575, pages 173–181. SPIE The International Society for Optical Engeneering, 1985.

[11] C.S. Choi, T. Okazaki, H. Harashima, and T. Takebe. A system of analyzing and synthesizing facial images. In *Proc. IEEE Int. Symposium of Circuit and Systems (ISCAS91)*, pages 2665–2668, 1991.

[12] T. F. Cootes, K. Walker, and C. J. Taylor. View-based active appearance models. In *Int. Conf. on Autom. Face and Gesture Recognition*, pages 227–232, 2000.

[13] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* John Wiley & Sons, New York, 2nd edition, 2001.

[14] G.J. Edwards, T.F. Cootes, and C.J. Taylor. Face recogition using active appearance models. In Burkhardt and Neumann, editors, *Computer Vision – ECCV'98*, Freiburg, 1998. Springer LNCS 1407.

[15] J.D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, Ma, 2. edition, 1996.

[16] 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. on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[17] D. B. Graham and N. M. Allison. Face recognition from unfamiliar views: Subspace methods and pose dependency. In *Int. Conf. on Autom. Face and Gesture Recognition*, pages 348–353, 1998.

[18] R. Gross, I. Matthews, and S. Baker. Eigen light-fields and face recognition across pose. In *Int. Conf. on Autom. Face and Gesture Recognition*, pages 3–9, 2002.

[19] P.W. Hallinan. *A deformable model for the recognition of human faces under arbitrary illumination.* PhD thesis, Harvard University, Cambridge, Mass., 1995.

[20] R.M. Haralick and L.G. Shapiro. *Computer and robot vision*, volume 2. Addison-Wesley, Reading, Ma, 1992.

[21] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[22] T. S. Huang and L. A. Tang. 3d face modeling and its applications. *Int. J. Pattern Recog. Artif. Intell.*, 10(5):491–519, 1996.

[23] M. Jones and T. Poggio. Multidimensional morphable models: A framework for representing and matching object classes. *Int. Journal of Comp. Vision*, 29(2):107–131, 1998.

[24] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401, 1995.

[25] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[26] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. IJCAI*, pages 674–679, 1981.

[27] T. Maurer and C. von der Malsburg. Single-view based recognition of faces rotated in depth. In *Int. Conf. on Autom. Face and Gesture Recognition*, pages 248–253, 1995.

[28] H. Moon and P. J. Phillips. Computational and performance aspects of pca-based face-recognition algorithms. *Perception*, 30:303–321, 2001.

[29] F.I. Parke. *A Parametric Model of Human Faces.* PhD thesis, University of Utah, Salt Lake City, 1974.

[30] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 84–91, 1994.

[31] 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*, 16(5):295–306, 1998.

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

[33] H. Robbins and S. Munroe. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[34] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *Int. Conf. on Autom. Face and Gesture Recognition*, pages 53–58, 2002.

[35] T. Sim and T. Kanade. Illuminating the face. Technical Report CMU-RI-TR-01-31, The Robotics Institute, Carnegie Mellon University, Sept. 2001.

[36] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 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. In *Int. Conf. on Autom. Face and Gesture Recognition*, pages 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.

A Optic Flow Calculation

Optic flow $\mathbf{v}$ between grey-level images at a given point (x_0, y_0) can be defined as the minimum $\mathbf{v}$ of a quadratic function (Equation 3). This minimum is given by [26, 3]

$$\mathbf{W}\mathbf{v} = -\mathbf{b} \tag{22}$$

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

$$\mathbf{b} = \begin{pmatrix} \sum \partial_x I \cdot \Delta I \\ \sum \partial_y I \cdot \Delta I \end{pmatrix}.$$

$\mathbf{v}$ is easy to find by means of a diagonalization of the 2x2 symmetrical matrix $\mathbf{W}$.

For 3D laser scans the minimum of Equation (4) is again given by (22), but now

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

$$\mathbf{b} = \begin{pmatrix} \sum \langle \partial_h \mathbf{I},\, \Delta \mathbf{I} \rangle \\ \sum \langle \partial_\phi \mathbf{I},\, \Delta \mathbf{I} \rangle \end{pmatrix}, \tag{23}$$

using the scalar product related to (5). $\mathbf{v}$ 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

$$E_c = \sum_h \sum_\phi \|\mathbf{v}_s(h+1, \phi) - \mathbf{v}_s(h, \phi)\|^2 + \sum_h \sum_\phi \|\mathbf{v}_s(h, \phi+1) - \mathbf{v}_s(h, \phi)\|^2 . \tag{24}$$

The coupling of $\mathbf{v}_s(h, \phi)$ to the original flow field $\mathbf{v}_0(h, \phi)$ depends on the rank of the 2x2 matrix $\mathbf{W}$ in (23), which determines if Equation (22) 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

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

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 (22) 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 (22), and therefore $\mathbf{v}_s$ is restricted in all directions. To compute $\mathbf{v}_s$, we apply Conjugate Gradient Descent [32] 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.

B Stochastic Newton Algorithm

For the optimization of the cost function (20), we developed a stochastic version of Newton's algorithm [6] similar to stochastic gradient descent [33, 37, 23]. 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. The derivatives are given in Appendix C.

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

$$\boldsymbol{\alpha}^* = \boldsymbol{\alpha} - \mathbf{H}^{-1} \nabla E. \tag{25}$$

For simplification, we consider α_i as a general set of model parameters here, and suppress $\boldsymbol{\beta}$, $\boldsymbol{\rho}$. Equation (20) is then

$$E(\boldsymbol{\alpha}) = \frac{1}{\sigma_I^2} E_I(\boldsymbol{\alpha}) + \frac{1}{\sigma_F^2} E_F(\boldsymbol{\alpha}) + \sum_i \frac{(\alpha_i - \overline{\alpha}_i)^2}{\sigma_{S,i}^2}, \tag{26}$$

and

$$\nabla E = \frac{1}{\sigma_I^2} \frac{\partial E_I}{\partial \alpha_i} + \frac{1}{\sigma_F^2} \frac{\partial E_F}{\partial \alpha_i} + diag\left(\frac{2}{\sigma_{S,i}^2}\right)(\boldsymbol{\alpha} - \overline{\boldsymbol{\alpha}}). \tag{27}$$

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

$$H_{i,i} = \frac{1}{\sigma_I^2} \frac{\partial^2 E_I}{\partial \alpha_i^2} + \frac{1}{\sigma_F^2} \frac{\partial^2 E_F}{\partial \alpha_i^2} + \frac{2}{\sigma_{S,i}^2}. \tag{28}$$

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 1000 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 [6]) of $\mathbf{H}$, and set $\mathbf{H}^{-1} \approx diag(1/H_{i,i})$. With Equation (25), the estimated optimum is

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

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

C Derivatives of E_I

In this section, we give the derivatives of E_I (Equation 16) by the model coefficients α_i. Derivatives of the other contributions to the cost function (20) and for the coefficients β_i, ρ_i are calculated in a similar way.

Rendering is a composition of many operations. Our implementation introduces variables for intermediate derivatives that are substituted on the next level, using chain rule. Summarizing over all randomly selected vertices k and color channels $f = r, g, b$,

$$\frac{\partial E_I}{\partial \alpha_i} = 2 \sum_k \sum_{f=r,g,b} (I_{f,input}(p_{x,k}, p_{y,k}) - I_{f,model,k}) \cdot \left(\frac{\partial}{\partial \alpha_i} I_{f,input}(p_{x,k}, p_{y,k}) - \frac{\partial}{\alpha_i} I_{f,model,k} \right),$$

$$\frac{\partial}{\partial \alpha_i} I_{f,input}(p_{x,k}, p_{y,k}) = \frac{\partial I_{f,input}}{\partial x}\frac{\partial p_{x,k}}{\partial \alpha_i} + \frac{\partial I_{f,input}}{\partial y}\frac{\partial p_{y,k}}{\partial \alpha_i}.$$

Derivatives $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ at image positions $(p_{x,k}, p_{y,k})$ are computed by a sobel operator.

Starting from the first steps of the rendering process, the derivative of the linear combination of basis shapes (8) is $\frac{\partial \mathbf{S}}{\partial \alpha_i} = \mathbf{s}_i$, so for the corners $\mathbf{x}_{k_1}$, $\mathbf{x}_{k_2}$, $\mathbf{x}_{k_3}$ of a triangle, we can simply look up $\frac{\partial \mathbf{x}_{k_1}}{\partial \alpha_i}, \ldots$ in $\mathbf{s}_i$. For the center $\mathbf{x}_k$, $\frac{\partial \mathbf{x}_k}{\partial \alpha_i} = \frac{1}{3}(\frac{\partial \mathbf{x}_{k_1}}{\partial \alpha_i} + \frac{\partial \mathbf{x}_{k_2}}{\partial \alpha_i} + \frac{\partial \mathbf{x}_{k_3}}{\partial \alpha_i})$. Derivatives of rigid transformation (10) and perspective projection (11) are

$$\frac{\partial \mathbf{w}_k}{\partial \alpha_j} = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \frac{\partial \mathbf{x}_k}{\partial \alpha_j} \tag{29}$$

$$\frac{\partial p_{x,k}}{\partial \alpha_j} = f \frac{1}{w_{z,k}} \left(\frac{\partial w_{x,k}}{\partial \alpha_j} - (p_{x,k} - P_x) \frac{\partial w_{z,k}}{\partial \alpha_j} \right) \tag{30}$$

$\frac{\partial p_{y,k}}{\partial \alpha_j}$ is calculated in the same way. Shape coefficients not only control the image positions $(p_{x,k}, p_{y,k})$ of vertices k, but due to the changes in surface normals, they also affect the vertex colors $I_{f,model,k}$. Calculating the normal vector of the triangle involves three steps $\underline{\mathbf{n}} = (\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) \times (\mathbf{x}_{k_1} - \mathbf{x}_{k_3})$, $\hat{\mathbf{n}} = \frac{\underline{\mathbf{n}}}{\|\underline{\mathbf{n}}\|}$, $\mathbf{n} = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \hat{\mathbf{n}}$ with derivatives

$$\begin{aligned}
\frac{\partial \underline{\mathbf{n}}}{\partial \alpha_i} &= (\frac{\partial \mathbf{x}_{k_1}}{\partial \alpha_i} - \frac{\partial \mathbf{x}_{k_2}}{\partial \alpha_i}) \times (\mathbf{x}_{k_1} - \mathbf{x}_{k_3}) \\
&\quad + (\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) \times (\frac{\partial \mathbf{x}_{k_1}}{\partial \alpha_i} - \frac{\partial \mathbf{x}_{k_3}}{\partial \alpha_i}) \\
\frac{\partial \hat{\mathbf{n}}}{\partial \alpha_i} &= \|\underline{\mathbf{n}}\|^{-1} \frac{\partial \underline{\mathbf{n}}}{\partial \alpha_i} - \|\underline{\mathbf{n}}\|^{-3} \left\langle \frac{\partial \underline{\mathbf{n}}}{\partial \alpha_i}, \underline{\mathbf{n}} \right\rangle \cdot \underline{\mathbf{n}} \\
&= \|\underline{\mathbf{n}}\|^{-1} \cdot (\frac{\partial \underline{\mathbf{n}}}{\partial \alpha_i} - \left\langle \frac{\partial \underline{\mathbf{n}}}{\partial \alpha_i}, \hat{\mathbf{n}} \right\rangle \hat{\mathbf{n}}) \\
\frac{\partial \mathbf{n}}{\partial \alpha_i} &= \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \frac{\partial \hat{\mathbf{n}}}{\partial \alpha_i}.
\end{aligned} \tag{31}$$

The direction of reflection changes according to

$$\frac{\partial \mathbf{r}_k}{\partial \alpha_j} = 2 \left\langle \frac{\partial \mathbf{n}_k}{\partial \alpha_j}, \mathbf{l} \right\rangle \mathbf{n}_k + 2 \langle \mathbf{n}_k, \mathbf{l} \rangle \frac{\partial \mathbf{n}_k}{\partial \alpha_j}. \tag{32}$$

We ignore the influence of $\frac{\partial \mathbf{x}_k}{\partial \alpha_j}$ on the viewing direction $\hat{\mathbf{v}}_k$. Then, Phong illumination (13) yields for the red channel

$$\begin{aligned}
\frac{\partial L_r}{\partial \alpha_j} &= R_k \cdot L_{r,dir} \cdot \left\langle \frac{\partial \mathbf{n}_k}{\partial \alpha_j}, \mathbf{l} \right\rangle \\
&\quad + k_s \cdot L_{r,dir} \cdot \nu \cdot \langle \mathbf{r}_k, \hat{\mathbf{v}}_k \rangle^{\nu-1} \cdot \left\langle \frac{\partial \mathbf{r}_k}{\partial \alpha_j}, \hat{\mathbf{v}}_k \right\rangle
\end{aligned} \tag{33}$$

Finally, the derivative of the color transformation (14),(15) for $f = r, g, b$ is

$$\begin{aligned}
\frac{\partial I_{f,model,k}}{\alpha_i} &= g_f (c \frac{\partial L_f}{\alpha_i} + (1 - c) \\
&\quad \cdot (0.3 \cdot \frac{\partial L_r}{\alpha_i} + 0.59 \cdot \frac{\partial L_g}{\alpha_i} + 0.11 \cdot \frac{\partial L_b}{\alpha_i})).
\end{aligned}$$