Fitting a Morphable Model to 3D Scans of Faces

Volker Blanz
Universität Siegen,
Siegen, Germany
blanz@informatik.uni-siegen.de

Kristina Scherbaum
MPI Informatik,
Saarbrücken, Germany
scherbaum@mpi-inf.mpg.de

Hans-Peter Seidel
MPI Informatik,
Saarbrücken, Germany
hpseidel@mpi-inf.mpg.de

Abstract

This paper presents a top-down approach to 3D data analysis by fitting a Morphable Model to scans of faces. In a unified framework, the algorithm optimizes shape, texture, pose and illumination simultaneously. The algorithm can be used as a core component in face recognition from scans. In an analysis-by-synthesis approach, raw scans are transformed into a PCA-based representation that is robust with respect to changes in pose and illumination. Illumination conditions are estimated in an explicit simulation that involves specular and diffuse components. The algorithm inverts the effect of shading in order to obtain the diffuse reflectance in each point of the facial surface. Our results include illumination correction, surface completion and face recognition on the FRGC database of scans.

1. Introduction

Face recognition from 3D scans has become a very active field of research due to the rapid progress in 3D scanning technology. In scans, changes in pose are easy to compensate by a rigid transformation. On the other hand, range data are often noisy and incomplete, and using shape only would ignore many person-specific features such as the colors of the eyes.

The main idea of our approach is to exploit both shape and texture information of the input scan in a simultaneous fitting procedure, and to use a 3D Morphable Model for a PCA-based representation of faces. Our method builds upon an algorithm for fitting a Morphable Model to photographs [6]. We generalize this algorithm by including range data in the cost function that is optimized during fitting. More specifically, the algorithm synthesizes a random subset of pixels from the scan in each iteration by simulating rigid transformation, perspective projection and illumination. In an iterative optimization, it makes these as similar as possible to the color and depth values found in the scan. Based on an analytical derivative of the cost function, the algorithm optimizes pose, shape, texture and lighting. For initialization, the algorithm uses a set of about 7 feature points that have to be defined manually, or may be identified automatically by feature detection algorithms.

One of the outputs of the system is a set of model coefficients that can be used for face recognition. Moreover, we obtain a textured 3D model from the linear span of example faces of the Morphable Model. The fitting procedure establishes point-to-point correspondence of the model to the scan, so we can sample the veridical cartesian coordinates and color values of the scan, and substitute them in the face model. The result is a resampled version of the original scan that can be morphed with other faces. We estimate and remove the effect of illumination, and thus obtain the approximate diffuse reflectance at each surface point. This is important for simulating new illuminations on the scan.

The contributions of this paper are:

- An algorithm for fitting a model to shape and texture *simultaneously*,
- The algorithm is specifically designed for perspective projection, which is found in most scanners, such as structured light scanners, time-of-flight scanners and those laser scanners that have a fixed center of projection,
- Compensation of lighting effects as an integrated part of the fitting procedure,
- Simulation of both specular and diffuse reflection,
- Model-based handling of saturated color values in the texture (which are common in scans and pose problems to non model-based approaches), and
- A comparison between recognition from photographs (textures) only, with recognition from scans in a sophisticated, model-based algorithm.

2. Related Work

Many of the early methods on face recognition from range data have relied on feature points, curvatures or curves [8, 16, 13, 4]. Other geometrical criteria include Hausdorff-distance [17], free-form surfaces and point signatures [25, 10], or bending-invariant canonical forms for surface representation [7].

Similar to the Eigenface approach in image data, several authors have applied Principal Component Analysis (PCA)

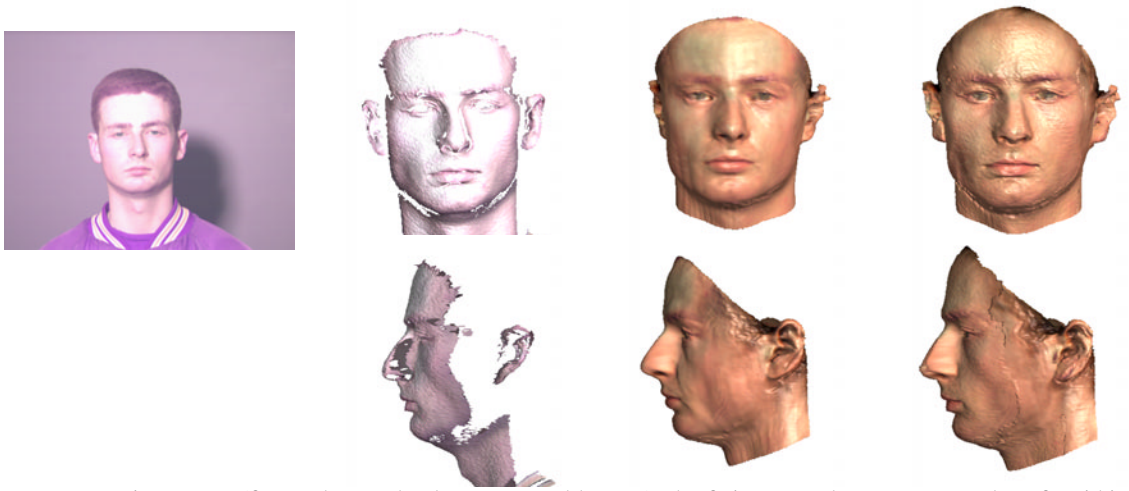

Figure 1. From a raw input scan (first and second column, top and bottom), the fitting procedure generates a best fit within the vector space of examples (third column). Shape and texture sampling (Section 5) includes the original data into the reconstructed model wherever possible (fourth column). Note that the algorithm has automatically removed lighting effects, including the saturated color values.

to range data after rigid alignment of the data [2, 1, 14, 9, 28], or after registration in the image plane using salient features [15].

In 3D Morphable Models, PCA is not applied to the depth or radius values, but to the cartesian coordinates of surface points [5]. It is important that prior to PCA, the scans are registered in a dense point-by-point correspondence using a modified optical flow algorithm that identifies corresponding features in the range data [5]. This algorithm has only been used for *building* the Morphable Model, but not for recognizing faces in new scans, which would probably be possible. The Morphable Model has been studied in the field of image analysis and shape reconstruction [5, 6].

On a very general level, it is a powerful approach to identify corresponding features in order to solve recognition problems. In face recognition from images, this is reflected in the paradigm shift from Eigenfaces [26] to Morphable Models [27, 5] and Active Appearance Models [11]. Corresponding features may either be found by dedicated feature detectors, or by an iterative fitting procedure that minimizes a distance function.

A number of algorithms have been presented for the alignment of face scans. Unlike Iterative Closest Point (ICP) algorithms [3], which is designed for registering parts of the same surface by a rigid transformation, these can be used to register scans of different individuals.

Blanz and Vetter briefly describe an algorithm for fitting the Morphable Model to new 3D scans [5]: In an analysis-by-synthesis loop that is similar to their image analysis algorithm [5, 6], the algorithm strives to reproduce the radius values of cylindrical $Cyberware^{TM}$ scans by a linear combination of example faces and a rigid transformation. The algorithm minimizes the sum of square distance of the radius and texture values of the model to those of the input scan. Our algorithm takes this approach several steps further by posing the problem of 3D shape fitting as a generalized problem of fitting to images, by dealing with the natural representation of most 3D scanners, which is based on perspective projection, and by taking illumination into account.

Zhang et al. [29] presented a method that tracks a face model during a 3D motion sequence, and that fits a template mesh to the initial frame. Similar to our approach, they project the model to the image plane of the depth map, and update the shape such that it follows the shape with minimal changes. They use texture to track the motion of points in the image plane during the sequence by an optical flow algorithm.

Mao et al [21] use elastic deformation of a generic model, starting from manually placed landmarks. For a comparison of untextured scans, Russ et al. [24] detect five facial features, perform ICP and a subsequent normal search to establish correspondence between an input scan and a reference face. Face recognition is then based on PCA coefficients of shape vectors. Unlike this algorithm, our approach deforms the PCA-based model to fit the input scan, solving the problem of correspondence and PCA decomposition simultaneously. Mian et al. [22] compare pairs of scans using a hybrid system that uses feature points, ICP of the nose and forehead areas, and PCA.

While the previous algorithms did not account for illumination effects in the scans, Malassiotis and Strintzis [20] use the depth information of scans for face detection, pose estimation and a warp-based rectification of the input image (texture of the scan). To make the system robust with respect to illumination, the faces in the database are rendered with different lightings, and a Support Vector Machine is trained on these data.

Lu et al. [19] construct a 3D model of a face by combining several 2.5D scans, and then match this to a new probe

scan by coarse alignment based on feature points, and fine alignment based on ICP. Root mean square distance is used as a measure for shape similarity. On a set of candidates, they synthesize different shadings of their textures, and use LDA for a comparison with the probe texture. Unlike this work, we fit a deformable model to the scan, and integrate this with a closed analysis-by-synthesis loop for simulating the effect of lighting on texture. Lu and Jain [18] consider non-rigid deformations due to facial expressions, and iteratively optimize in alternating order the rigid transformation by ICP, and the expression by minimizing the sum of squared distances. Texture is not estimated in this work. In contrast, we optimize rigid transformation, non-rigid deformation, texture and lighting in a unified framework.

3. A Morphable Model of 3D Faces

This section summarizes how a Morphable Model of 3D faces[27, 5] is built from a training set of 200 textured $Cyberware^{TM}$ laser scans that are stored in cylindrical coordinates. These scans cover most of the facial surface from ear to ear, and are relatively high quality, but it takes about 20 seconds to record a full scan because the sensor of the scanner moves around the persons' heads. In Section 4, this general Morphable Model will be applied to input scans of new individuals recorded with a scanner that uses a perspective projection.

In the Morphable Model, shape and texture vectors are defined such that any linear combination of examples

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

is a realistic face if $\mathbf{S}$, $\mathbf{T}$ are within a few standard deviations from their averages. In the conversion of the laser scans of the training set into shape and texture vectors $\mathbf{S}_i$, $\mathbf{T}_i$, it is essential to establish dense point-to-point correspondence of all scans with a reference face to make sure that vector dimensions in $\mathbf{S}$, $\mathbf{T}$ describe the same point, such as the tip of the nose, in all faces. Correspondence is computed automatically using optical flow [5].

Each vector $\mathbf{S}_i$ is the 3D shape, stored in terms of x, y, z-coordinates of all vertices $k \in \{1, \ldots, n\}$, $n = 75972$ of a 3D mesh:

$$\mathbf{S}_i = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T. \tag{2}$$

In the same way, we form texture vectors from the red, green, and blue values of all vertices' surface colors:

$$\mathbf{T}_i = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \tag{3}$$

Finally, we perform a Principal Component Analysis (PCA) to estimate the principal axes $\mathbf{s}_i$, $\mathbf{t}_i$ of variation around the averages $\overline{\mathbf{s}}$ and $\overline{\mathbf{t}}$, and the standard deviations $\sigma_{S,i}$ and $\sigma_{T,i}$. The principal axes form an orthogonal basis, so

$$\mathbf{S} = \overline{\mathbf{s}} + \sum_{i=1}^{m} \alpha_i \cdot \mathbf{s}_i, \qquad \mathbf{T} = \overline{\mathbf{t}} + \sum_{i=1}^{m} \beta_i \cdot \mathbf{t}_i. \tag{4}$$

4. Model-Based Shape Analysis

The fitting algorithm is a generalization of a model-based algorithm for image analysis [6]. As we have pointed out above, most 3D scans are parameterized and sampled in terms of image coordinates u, v in a perspective projection. In each sample point, the scan stores the r, g, b component of the texture, and the cartesian coordinates of x, y, z of the point, so we can write the scan as

$$\begin{aligned} \mathbf{I}_{input}(u,v) \quad = (\quad & r(u,v), g(u,v), b(u,v), \\ & x(u,v), y(u,v), z(u,v))^T. \end{aligned} \tag{5}$$

The algorithm solves the following optimization problem: Given $\mathbf{I}_{input}(u, v)$, find the shape and texture vectors $\mathbf{S}$, $\mathbf{T}$, the rigid pose transformation, camera parameters and lighting such that

1. the camera produces a color image that is as similar as possible to the texture $r(u, v), g(u, v), b(u, v)$, and
2. the cartesian coordinates of the surface points fit the shape of $\mathbf{x}(u, v) = (x(u, v), y(u, v), z(u, v))^T$.

Solving the first problem, which is equivalent to the 3D shape reconstruction from images [6], uniquely defines the rigid transformation and all the other parameters: Points such as the tip of the nose, which have coordinates $\mathbf{x}_k = (x_k, y_k, z_k)^T$ within the shape vector $\mathbf{S}$, are mapped by the rigid transformation and the perspective projection to a pixel u_k, v_k in the image, and the color values in this pixel should be reproduced by the estimated texture and lighting.

The same perspective projection solves the second problem, because the pixel u_k, v_k also stores the 3D coordinates of the same point, $\mathbf{x}(u_k, v_k)$. However, the 3D coordinates will, in general, differ by a rigid transformation that depends on the definition of coordinates by the manufacturer of the scanner.

The algorithm, therefore, has to find two rigid transformations, one that maps the Morphable Model to camera coordinates such that the perspective projection fits with the coordinates u_k, v_k, and one that aligns the coordinate system of the model (in our case the camera coordinates) with the coordinate system of the scanner. We separate the two problems by pre-aligning the scans with our camera coordinate system in a first step. Before we describe this alignment, let us introduce some notation.

4.1. Rigid Transformation and Perspective Projection

In our analysis-by-synthesis approach, each vertex k is mapped from the model-based coordinates $\mathbf{x}_k = (x_k, y_k, z_k)^T$ in $\mathbf{S}$ (Equation 2) to the screen coordinates u_k, v_k in the following way:

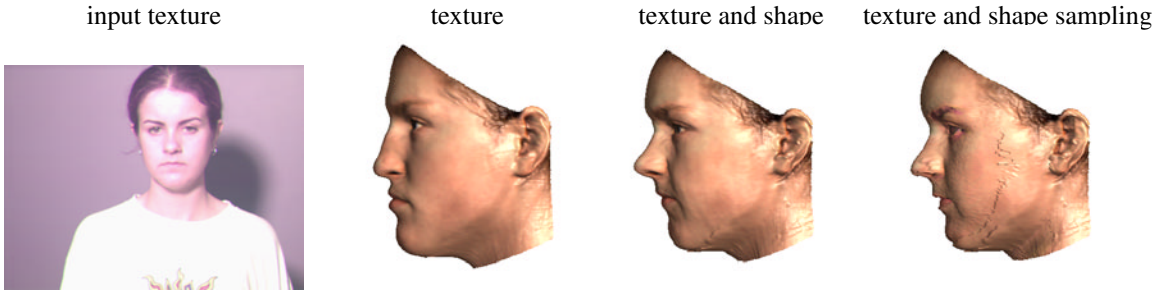

Figure 2. If the reconstruction is computed only from the input texture (first image), the algorithm estimates the most plausible 3D shape (second image), given the shading and shape of the front view. Fitting the model to both texture and shape (third image) captures more characteristics of the face, which are close to the ground truth that we obtain when sampling the texture and shape values (right image).

A rigid transformation maps $\mathbf{x}_k$ to a position relative to the camera:

$$\mathbf{w}_k = (w_{x,k}\ ,\ w_{y,k}\ ,\ w_{z,k})^T = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \mathbf{x}_k + \mathbf{t}_w. \quad (6)$$

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

A perspective projection then maps vertex k to image plane coordinates u_k, v_k:

$$u_k = u_0 + f\frac{w_{x,k}}{w_{z,k}}\ , \qquad v_k = v_0 - f\frac{w_{y,k}}{w_{z,k}}\ . \quad (7)$$

f is the focal length of the camera which is located in the origin, and u_0, v_0 defines the image-plane position of the optical axis (principal point).

4.2. Prealignment of Scans

In the scan, the 3D coordinates found in u_k, v_k are $\mathbf{x}(u_k, v_k)$. The camera of the scanner mapped these to u_k, v_k, so we can infer the camera calibration of the scanner and thus

1. transform the scan coordinates $\mathbf{x}(u, v)$ to camera coordinates $\mathbf{w}(u, v)$, i.e. estimate the extrinsic camera parameters.

2. estimate the focal length (and potentially more intrinsic camera parameters), and use these as fixed, known values in the subsequent model fitting.

In fact, the prealignment reverse-engineers the camera parameters that have been used in the software of the scanner from redundancies in the data. In the well-known literature on camera calibration, there are a number of algorithms that could be used for this task. For simplicity, we modified the model fitting algorithm [6] that will be used in the next processing step anyway. This makes sure that the definition of all camera parameters is consistent in both steps. The modified algorithm solves the non-linear problem of camera calibration iteratively.

First, we select $n = 10$ random non-void points u_i, v_i from the scan, and store their coordinates $\mathbf{x}(u_i, v_i)$. Equations (6) and (7) map these to image plane coordinates u'_i, v'_i. This defines a cost function

$$E_{cal}(\phi, \theta, \gamma, \mathbf{t}_w, f) = \sum_{i=1}^{n} (u'_i - u_i)^2 + (v'_i - v_i)^2.$$

We find the minimum of E_{cal} by Newton's algorithm, using analytic derivatives of the rigid transformation and the perspective projection (6), (7). Using the rigid transformation (6), we map all scan coordinates $\mathbf{x}(u, v)$ to our estimated camera coordinates $\mathbf{w}(u, v)$.

4.3. Fitting the Model to Scans

The fitting algorithm finds the model coefficients, rigid transformation and lighting such that each vertex k of the Morphable Model is mapped to image plane coordinates u_k, v_k such that the color values are matched and that the camera coordinates $w_{z,k}$ of the model are as close as possible to the camera coordinates $w_z(u_k, v_k)$ of the scan. Note that we only fit the depth w_z of the vertices: The frontoparallel coordinates w_x and w_y are fixed already by the fact that the model point and the scan point are in the same image position u_k, v_k, and an additional restriction in w_x and w_y would prevent the model from sliding along the surface to find the best match in terms of feature correspondence.

In order to fit the model to the texture of the scan, the algorithm has to compensate effects of illumination and of the overall color distribution.

4.3.1 Illumination and Color

We assume that the scanning setup involves similar lighting effects as a standard photograph. We propose to simulate this explicitly, in the same way as it has been done for fitting a model to images [6]. This paragraph summarizes the steps involved in image synthesis, which will be part of the analysis algorithm.

The normal vector to a triangle $k_1k_2k_3$ of the Morphable Model is given by a vector product of the edges, $\mathbf{n} = (\mathbf{x}_{k_1} - \mathbf{x}_{k_2}) \times (\mathbf{x}_{k_1} - \mathbf{x}_{k_3})$, which is normalized to unit length, and rotated along with the head (Equation 6). For fitting the model to an image, it is sufficient to consider the centers of triangles only, most of which are about $0.2\mathrm{mm}^2$ 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 algorithm simulates ambient light with red, green,

Figure 3. The textures on the right have been sampled from the scans in the left column. The inversion of illumination effects has removed most of the harsh lighting from the original textures. The method compensates both the results of overexposure and inhomogeneous shading of the face.

and blue intensities $L_{r,amb}$, $L_{g,amb}$, $L_{b,amb}$, and directed 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:

$$\mathbf{l} = (\cos(\theta_l)\sin(\phi_l),\ \sin(\theta_l),\ \cos(\theta_l)\cos(\phi_l))^T. \quad (8)$$

The illumination model of Phong (see [12]) approximately describes the diffuse and specular reflection of a surface. In each vertex k, the red channel is

$$L_{r,k} = R_k \cdot L_{r,amb} + R_k \cdot L_{r,dir} \cdot \langle \mathbf{n}_k, \mathbf{l} \rangle + k_s \cdot L_{r,dir} \langle \mathbf{r}_k, \widehat{\mathbf{v}}_k \rangle^\nu \quad (9)$$

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 [12].

Depending on the camera of the scanner, the textures may be color or gray level, and they may differ in overall tone. 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 [12]

$$L = 0.3 \cdot L_r + 0.59 \cdot L_g + 0.11 \cdot L_b. \quad (10)$$

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

$$r = g_r \cdot (cL_r + (1-c)L) + o_r. \quad (11)$$

Green and blue channels are computed in the same way. The colors r, g and b are drawn at a position (u, v) in the final image $\mathbf{I}_{model}$.

4.3.2 Optimization

Just as in image analysis [6], 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 21 rendering parameters, concatenated into a vector $\boldsymbol{\rho}$, that contains pose angles ϕ, θ and γ, 3D translation $\mathbf{t}_w$, 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. Unlike [6], we keep the focal length f fixed now.

The main part of the cost function is a least-squares difference between the transformed input scan

$$\mathbf{I}_{input}(u,v) = (r(u,v), g(u,v), b(u,v), w_z(u,v))^T \quad (12)$$

and the values $\mathbf{I}_{model}$ synthesized by the model

$$E_I = \sum_{u,v} (\mathbf{I}_{input} - \mathbf{I}_{model})^T \Lambda (\mathbf{I}_{input} - \mathbf{I}_{model}) \quad (13)$$

with a diagonal weight matrix Λ that contains an empirical scaling value between shape and texture, which is 128 in our system (depth is in mm, texture is in $\{0, ..., 255\}$.)

For initialization, another cost function is added to E_I that measures the distances between manually defined feature points j in the image plane, $u_{init,j}, v_{init,j}$, and the image coordinates of the projection $u_{model,k_j}, v_{model,k_j}$ of the corresponding, manually defined model vertices k_j:

$$E_F = \sum_j \| \begin{pmatrix} u_{init,j} \\ v_{init,j} \end{pmatrix} - \begin{pmatrix} u_{model,k_j} \\ v_{model,k_j} \end{pmatrix} \|^2. \quad (14)$$

This additional term pulls the face model to the approximate position in the image plane in the first iterations. Its weight is reduced to 0 during the process of optimization.

To avoid overfitting, we apply a regularization by adding penalty terms that measure the PCA-based Mahalanobis distance from the average face and the initial parameters [6, 5]:

$$E = \eta_I E_I + \eta_F E_F + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho}_i)^2}{\sigma_{R,i}^2}. \quad (15)$$

Ad-hoc choices of η_I and η_F are used to control the relative weights of E_I, E_F, and the prior probability terms in (15). 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.

Triangles that are invisible due to self-occlusion of the face are discarded in the cost function. This is tested by a z-buffer criterion. Also, we discard shape data that are void and colors that are saturated. The algorithm takes cast shadows into account in the Phong model, based on a shadow-buffer criterion.

Figure 4. Morph between the 3D face on the left and the face on the right. They both are in correspondence to the reference face due to the reconstruction using the Morphable Model and the sampling procedure. In the morph (middle), facial features are preserved while regions without remarkable characteristics are averaged smoothly.

The cost function is optimized with a stochastic version of Newton's method [6]: The algorithm selects 40 random triangles in each iteration, with a probability proportional to their area in the u, v domain, and evaluates E_I and its gradient only at their centers. The gradient is computed analytically using chain rule and the equations of the synthesis that were given in this section. After fitting the entire face model to the image, the eyes, nose, mouth, and the surrounding region are optimized separately. The fitting procedure takes about 4 minutes on a 3.4 GHz Xeon processor.

5. Shape and Texture Sampling

After the fitting procedure, the optimized face is represented as a shape vector $\mathbf{S}$ and a texture vector $\mathbf{T}$. Note that the texture values describe the diffuse reflectance of the face in each point, so the effect of illumination, which was part of the optimization problem, has already been compensated by the algorithm. However, both $\mathbf{S}$ and $\mathbf{T}$ are linear combinations of examples (Section 3), and they do not capture all details in shape or texture found in the original scan. For computer graphics applications, we can therefore sample the shape and texture of the original surface using the following algorithm, which is an extension of [5]:

For each vertex k, the optimized model and camera parameters predict an image-plane position u_k, v_k. There, we find the coordinates of the scanned point in camera coordinates, $\mathbf{w}(u_k, v_k)$ (Section 4.2). By inverting the rigid transformation (6) with the model parameters of the model fitting procedure (Section 4.3), we obtain coordinates that are consistent with the vertex coordinates $\mathbf{x}_k$ in $\mathbf{S}$ and can replace them. If the point u_k, v_k in the scan is void or the distance to the estimated position exceeds a threshold, the estimated value is retained.

In the color values in u_k, v_k, the effects of shading and color transformation have to be compensated. With the optimized parameters, we invert the color transformation (11), subtract the specular highlight (9) which we can estimate from the estimated light direction and surface normal, and divide by the sum of ambient and diffuse lighting to obtain the diffuse reflectances R_k, G_k, B_k. Note that the sampled scan is now a new shape and texture vector that is in full correspondence with the Morphable Model.

5.1. Saturated Color Values

In many scans and images, color values are saturated due to overexposure. On those pixels in the raw scan texture, the red, green or blue values are close to 255. We do not perform texture sampling in these pixels, because the assumptions of our illumination model are violated, so the inversion would not give correct results. Instead, the algorithm retains the estimated color values from the previous section. The model-based approach and the explicit simulation of lighting proves to be very convenient in this context.

For a smooth transition between sampled and estimated color values, the algorithm creates a lookup-mask in the u, v domain of the original scan, blurs this binary mask and uses the continuous values from the blurred mask as relative weights of sampled versus estimated texture. As a result, we obtain a texture vector that captures details of the eyes and other structures, but does not contain the specular highlights of the original data.

6. Results

We have tested the algorithm on a portion of the Face Recognition Grand Challenge (FRGC, [23].) We selected pairs of scans of 150 individuals, taken under uncontrolled conditions. The scans of each person were recorded on two different days. For fitting, we used the 100 most relevant principal components. We manually clicked 7 points in each face, such as the corners of the eyes and the tip of the nose.

Figure 1 shows a typical result of fitting the model to one of the scans. Given the shape and texture (top left image), which we rendered from novel viewpoints in the second column of the Figure, we obtained a best fit shown in the third column. The profile view of the reconstructed face shows many characteristic features of the face, such as the curved nose and the dent under the lower lip. To go beyond the linear span of examples, we sampled the true shape and texture of the face (right column). The images show that the reconstructed surface (at the ears) and the sampled surface are closely aligned. The mean depth error $|w_{z,k} - w_z(u_k, v_k)|$ between the vertices of the reconstruction in Figure 1 and the ground truth scan was $1.88\ mm$. The mean error on

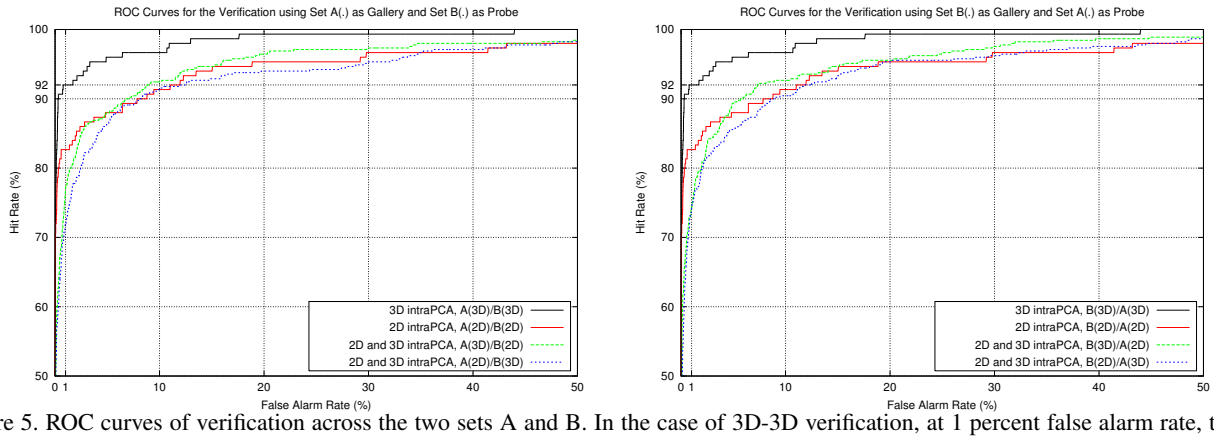

Figure 5. ROC curves of verification across the two sets A and B. In the case of 3D-3D verification, at 1 percent false alarm rate, the hit rate is 92 % for both types of comparison (A(3D)/B(3D) in the left image and B(3D)/A(3D) in the right image).

all $2 \cdot 150$ scans was $1.02\ mm$ when neglecting outliers above an euclidean distance of $10\ mm$ and vertex viewing angles above $80°$. The average percentage of outliers per face is 24%. When including also the outliers into the average mean error for all $2 \cdot 150$ scans, the result is $2.74\ mm$. Figure 2 shows an additional set of results and demonstrates how the 3D information improves the reconstruction of the profile from the front view.

As shown in Figure 1, the texture of the reconstructed and of the sampled face are normalized in terms of lighting and overall hue, so they can be used for simulating new illumination in computer graphics. To show how the algorithm removes lighting effects, Figure 3 gives a side-by-side comparison of two textures that were reconstructed and sampled from two different harsh illuminations. The result shows that the saturation of color values and the shading are removed successfully, and only a relatively small difference between the textures remains, so the textures are ready for simulating new illuminations.

In Figure 4, we show how two of the scans (the one from Figure 1 and the right face in Figure 4), can be morphed within the framework or the Morphable Model. This is due to the fact that both the reconstructed and the sampled faces are in correspondence with the reference face.

Finally, we investigated a face recognition scenario with our algorithm, and evaluated how the additional shape information improves the performance compared to the image-only condition. After model fitting, we rescaled the model coefficients to $\frac{\alpha_i}{\sigma_{S,i}}$ and $\frac{\beta_i}{\sigma_{T,i}}$ and concatenated them to a coefficient vector. By using 100 coefficients for shape and texture for the entire face and the segments eyes, nose, mouth and the surrounding region each, this adds up to 1000 dimensions. As a criterion for similarity, we used the scalar product. This is the same method as in [6]. We also performed a PCA of intra-object variation, and compensated for these variations [6]. Intra-object PCA was done with reconstructions from other faces that are not in the test set, and on image- or scan-based reconstructions for the image- or scan-based recognition. In the cross-modal condition, we used the intra-object PCA pooled from reconstructions from scans and from images.

Table 1 gives the percentage of correct identification for a comparison of scans versus scans, images versus images, and cross-modal recognition. The results in Table 1 indicate that the use of range data improves the performance, compared to the image-only condition. This is also shown in the ROC curve (Figure 5). The cross-modal condition is competitive to 2D-2D in verification, but not yet in identification. We plan a more sophisticated treatment of the intra-person variation between reconstructions from scans and those from images, but the results show already that the joint representation in the Morphable Model is a viable way for cross-modal recognition.

Gallery	Probe	Correct Ident.	intraPCA
A(3D)	B(3D)	96.0	3D
B(3D)	A(3D)	92.0	3D
A(2D)	B(2D)	84.7	2D
B(2D)	A(2D)	79.3	2D
A(3D)	B(2D)	71.3	2D and 3D
B(3D)	A(2D)	66.0	2D and 3D
A(2D)	B(3D)	66.7	2D and 3D
B(2D)	A(3D)	70.0	2D and 3D

Table 1. Percentages of correct identification of n=150 individuals in two sets of scans (A and B), comparing scans (3D shape and texture) or texture only (2D). The last four rows show cross-modal recognition.

7. Conclusion

Our results demonstrate that analysis-by-synthesis is not only a promising strategy in image analysis, but can also be applied to range data. The main idea is to simulate explicitly the projection of surface data into pixels of a scan, and the effects of illumination that are found in the texture. The technique has a number of applications in biometric identification, but also in Computer Graphics, for example as a robust and reliable way to transform scans into shape and texture vectors in a Morphable Model for animation and

other high-level manipulations. It can be used for bootstrapping the Morphable Model [5] by including more and more scans in the vector space of faces. The algorithm may also be a tool for preprocessing raw scans, filling in missing regions automatically, and registering multiple scans.

References

[1] B. Achermann, X. Jiang, and H. Bunke. Face recognition using range images. In *VSMM '97: Proc. of the 1997 Int. Conf. on Virtual Systems and MultiMedia*, page 129, Washington, DC, USA, 1997. IEEE Computer Society.

[2] J. J. Atick, P. A. Griffin, and A. N. Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8:1321–1340, 1996.

[3] P. J. Besl and N. D. McKay. A method for registration of 3–D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[4] C. Beumier and M. Acheroy. Face verification from 3D and grey level clues. *Pattern Recognition Letters*, 22:1321–1329, 2001.

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

[6] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(9):1063– 1074, 2003.

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Expression-invariant 3d face recognition. In *Proc. Audio- and Video-based Biometric Person Authentication (AVBPA)*, Lecture Notes in Comp. Science No. 2688, pages 62–69. Springer, 2003.

[8] J. Y. Cartoux, J. T. Lapreste, and M. Richetin. Face authentification or recognition by profile extraction from range images. In *Workshop on Interpretation of 3D Scenes*, pages 194– 199, 1989.

[9] K. I. Chang, K. W. Bowyer, and P. J. Flynn. An evaluation of multimodal 2d+3d face biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):619–624, 2005.

[10] C.-S. Chua, F. Han, and Y. K. Ho. 3D human face recognition using point signature. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 233– 238, 2000.

[11] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In Burkhardt and Neumann, editors, *Computer Vision – ECCV'98 Vol. II*, Freiburg, Germany, 1998. Springer, Lecture Notes in Computer Science 1407.

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

[13] G. G. Gordon. Face recognition based on depth and curvature features. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 808– 810, 1992.

[14] T. Heseltine, N. Pears, and J. Austin. Three-dimensional face recognition: An eigensurface approach. In *Proc. IEEE International Conference on Image Processing*, pages 1421–1424, Singapore, 2004. poster.

[15] C. Hesher, A. Srivastava, and G. Erlebacher. A novel technique for face recognition using range imaging. In *Proc. Seventh International Symposium on Signal Processing and Its Applications*, volume 2, pages 201–204, 2003.

[16] J. C. Lee and E. Milios. Matching range images of human faces. In *Proc. IEEE International Conference on Computer Vision*, pages 722– 726, 1990.

[17] Y. H. Lee and J. C. Shim. Curvature based human face recognition using depth weighted hausdorff distance. In *Proc. IEEE International Conference on Image Processing*, pages 1429–1432, Singapore, 2004.

[18] X. Lu and A. K. Jain. Deformation modeling for robust 3d face matching. In *CVPR '06*, pages 1377–1383, Washington, DC, USA, 2006. IEEE Computer Society.

[19] X. Lu, A. K. Jain, and D. Colbry. Matching 2.5d face scans to 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(1):31–43, 2006.

[20] S. Malassiotis and M. G. Strintzis. Pose and illumination compensation for 3d face recognition. In *Proc. International Conference on Image Processing*, Singapore, 2004.

[21] Z. Mao, J. P. Siebert, W. P. Cockshott, and A. F. Ayoub. Constructing dense correspondences to analyze 3d facial change. In *ICPR '04, Volume 3*, pages 144–148, Washington, DC, USA, 2004. IEEE Computer Society.

[22] A. S. Mian, M. Bennamoun, and R. Owens. 2d and 3d multimodal hybrid face recognition. In *ECCV 06*, pages 344–355, 2006.

[23] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *CVPR '05 - Volume 1*, pages 947–954, Washington, DC, USA, 2005. IEEE Computer Society.

[24] T. Russ, C. Boehnen, and T. Peters. 3d face recognition using 3d alignment for pca. In *CVPR '06*, pages 1391–1398, Washington, DC, USA, 2006. IEEE Computer Society.

[25] H. T. Tanaka, M. Ikeda, and H. Chiaki. Curvature-based face surface recognition using spherical correlation - principal direcions for curved object recognition. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 372– 377, 1998.

[26] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[27] 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.

[28] X. Yuan, J. Lu, and T. Yahagi. A method of 3d face recognition based on principal component analysis algorithm. In *IEEE International Symposium on Circuits and Systems*, volume 4, pages 3211–3214, 2005.

[29] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 548–558, New York, NY, USA, 2004. ACM Press.