A Morphable Model For The Synthesis Of 3D Faces

Volker Blanz   Thomas Vetter

Max-Planck-Institut für biologische Kybernetik,
Tübingen, Germany*

Abstract

In this paper, a new technique for modeling textured 3D faces is introduced. 3D faces can either be generated automatically from one or more photographs, or modeled directly through an intuitive user interface. Users are assisted in two key problems of computer aided face modeling. First, new face images or new 3D face models can be registered automatically by computing dense one-to-one correspondence to an internal face model. Second, the approach regulates the naturalness of modeled faces avoiding faces with an "unlikely" appearance.

Starting from an example set of 3D face models, we derive a morphable face model by transforming the shape and texture of the examples into a vector space representation. New faces and expressions can be modeled by forming linear combinations of the prototypes. Shape and texture constraints derived from the statistics of our example faces are used to guide manual modeling or automated matching algorithms.

We show 3D face reconstructions from single images and their applications for photo-realistic image manipulations. We also demonstrate face manipulations according to complex parameters such as gender, fullness of a face or its distinctiveness.

Keywords: facial modeling, registration, photogrammetry, morphing, facial animation, computer vision

1 Introduction

Computer aided modeling of human faces still requires a great deal of expertise and manual control to avoid unrealistic, non-face-like results. Most limitations of automated techniques for face synthesis, face animation or for general changes in the appearance of an individual face can be described either as the problem of finding corresponding feature locations in different faces or as the problem of separating realistic faces from faces that could never appear in the real world. The correspondence problem is crucial for all morphing techniques, both for the application of motion-capture data to pictures or 3D face models, and for most 3D face reconstruction techniques from images. A limited number of labeled feature points marked in one face, e.g., the tip of the nose, the eye corner and less prominent points on the cheek, must be located precisely in another face. The number of manually labeled feature points varies from application to application, but usually ranges from 50 to 300.

*MPI für biol. Kybernetik, Spemannstr. 38, 72076 Tübingen, Germany. E-mail: {volker.blanz, thomas.vetter}@tuebingen.mpg.de

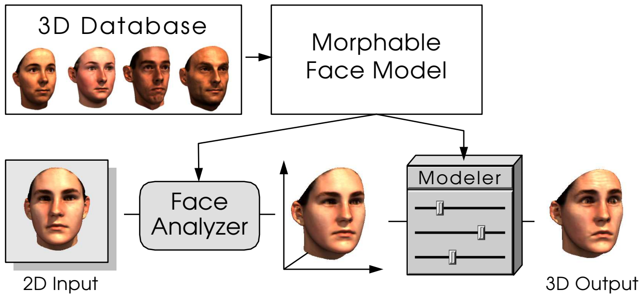

Figure 1: Derived from a dataset of prototypical 3D scans of faces, the morphable face model contributes to two main steps in face manipulation: (1) deriving a 3D face model from a novel image, and (2) modifying shape and texture in a natural way.

Only a correct alignment of all these points allows acceptable intermediate morphs, a convincing mapping of motion data from the reference to a new model, or the adaptation of a 3D face model to 2D images for 'video cloning'. Human knowledge and experience is necessary to compensate for the variations between individual faces and to guarantee a valid location assignment in the different faces. At present, automated matching techniques can be utilized only for very prominent feature points such as the corners of eyes and mouth.

A second type of problem in face modeling is the separation of natural faces from non faces. For this, human knowledge is even more critical. Many applications involve the design of completely new natural looking faces that can occur in the real world but which have no "real" counterpart. Others require the manipulation of an existing face according to changes in age, body weight or simply to emphasize the characteristics of the face. Such tasks usually require time-consuming manual work combined with the skills of an artist.

In this paper, we present a parametric face modeling technique that assists in both problems. First, arbitrary human faces can be created simultaneously controlling the likelihood of the generated faces. Second, the system is able to compute correspondence between new faces. Exploiting the statistics of a large dataset of 3D face scans (geometric and textural data, $Cyberware^{TM}$) we built a morphable face model and recover domain knowledge about face variations by applying pattern classification methods. The morphable face model is a multidimensional 3D morphing function that is based on the linear combination of a large number of 3D face scans. Computing the average face and the main modes of variation in our dataset, a probability distribution is imposed on the morphing function to avoid unlikely faces. We also derive parametric descriptions of face attributes such as gender, distinctiveness, "hooked" noses or the weight of a person, by evaluating the distribution of exemplar faces for each attribute within our face space.

Having constructed a parametric face model that is able to generate almost any face, the correspondence problem turns into a mathematical optimization problem. New faces, images or 3D face scans, can be registered by minimizing the difference between the new face and its reconstruction by the face model function. We developed an algorithm that adjusts the model parameters automatically for an optimal reconstruction of the target, requiring only a minimum of manual initialization. The output of the matching procedure is a high quality 3D face model that is in full correspondence with our morphable face model. Consequently all face manipulations parameterized in our model function can be mapped to the target face. The prior knowledge about the shape and texture of faces in general that is captured in our model function is sufficient to make reasonable estimates of the full 3D shape and texture of a face even when only a single picture is available. When applying the method to several images of a person, the reconstructions reach almost the quality of laser scans.

1.1 Previous and related work

Modeling human faces has challenged researchers in computer graphics since its beginning. Since the pioneering work of Parke [25, 26], various techniques have been reported for modeling the geometry of faces [10, 11, 22, 34, 21] and for animating them [28, 14, 19, 32, 22, 38, 29]. A detailed overview can be found in the book of Parke and Waters [24].

The key part of our approach is a generalized model of human faces. Similar to the approach of DeCarlos et al. [10], we restrict the range of allowable faces according to constraints derived from prototypical human faces. However, instead of using a limited set of measurements and proportions between a set of facial landmarks, we directly use the densely sampled geometry of the exemplar faces obtained by laser scanning ($Cyberware^{TM}$). The dense modeling of facial geometry (several thousand vertices per face) leads directly to a triangulation of the surface. Consequently, there is no need for variational surface interpolation techniques [10, 23, 33]. We also added a model of texture variations between faces. The morphable 3D face model is a consequent extension of the interpolation technique between face geometries, as introduced by Parke [26]. Computing correspondence between individual 3D face data automatically, we are able to increase the number of vertices used in the face representation from a few hundreds to tens of thousands. Moreover, we are able to use a higher number of faces, and thus to interpolate between hundreds of 'basis' faces rather than just a few. The goal of such an extended morphable face model is to represent any face as a linear combination of a limited basis set of face prototypes. Representing the face of an arbitrary person as a linear combination (morph) of "prototype" faces was first formulated for image compression in telecommunications [8]. Image-based linear 2D face models that exploit large data sets of prototype faces were developed for face recognition and image coding [4, 18, 37].

Different approaches have been taken to automate the matching step necessary for building up morphable models. One class of techniques is based on optic flow algorithms [5, 4] and another on an active model matching strategy [12, 16]. Combinations of both techniques have been applied to the problem of image matching [36]. In this paper we extend this approach to the problem of matching 3D faces.

The correspondence problem between different three-dimensional face data has been addressed previously by Lee et al.[20]. Their shape-matching algorithm differs significantly from our approach in several respects. First, we compute the correspondence in high resolution, considering shape and texture data simultaneously. Second, instead of using a physical tissue model to constrain the range of allowed mesh deformations, we use the statistics of our example faces to keep deformations plausible. Third, we do not rely on routines that are specifically designed to detect the features exclusively found in faces, e.g., eyes, nose.

Our general matching strategy can be used not only to adapt the morphable model to a 3D face scan, but also to 2D images of faces. Unlike a previous approach [35], the morphable 3D face model is now directly matched to images, avoiding the detour of generating intermediate 2D morphable image models. As a consequence, head orientation, illumination conditions and other parameters can be free variables subject to optimization. It is sufficient to use rough estimates of their values as a starting point of the automated matching procedure.

Most techniques for 'face cloning', the reconstruction of a 3D face model from one or more images, still rely on manual assistance for matching a deformable 3D face model to the images [26, 1, 30]. The approach of Pighin et al. [28] demonstrates the high realism that can be achieved for the synthesis of faces and facial expressions from photographs where several images of a face are matched to a single 3D face model. Our automated matching procedure could be used to replace the manual initialization step, where several corresponding features have to be labeled in the presented images.

For the animation of faces, a variety of methods have been proposed. For a complete overview we again refer to the book of Parke and Waters [24]. The techniques can be roughly separated in those that rely on physical modeling of facial muscles [38, 17], and in those applying previously captured facial expressions to a face [25, 3]. These performance based animation techniques compute the correspondence between the different facial expressions of a person by tracking markers glued to the face from image to image. To obtain photo-realistic face animations, up to 182 markers are used [14]. Working directly on faces without markers, our automated approach extends this number to its limit. It matches the full number of vertices available in the face model to images. The resulting dense correspondence fields can even capture changes in wrinkles and map these from one face to another.

1.2 Organization of the paper

We start with a description of the database of 3D face scans from which our morphable model is built.

In Section 3, we introduce the concept of the morphable face model, assuming a set of 3D face scans that are in full correspondence. Exploiting the statistics of a dataset, we derive a parametric description of faces, as well as the range of plausible faces. Additionally, we define facial attributes, such as gender or fullness of faces, in the parameter space of the model.

In Section 4, we describe an algorithm for matching our flexible model to novel images or 3D scans of faces. Along with a 3D reconstruction, the algorithm can compute correspondence, based on the morphable model.

In Section 5, we introduce an iterative method for building a morphable model automatically from a raw data set of 3D face scans when no correspondences between the exemplar faces are available.

2 Database

Laser scans ($Cyberware^{TM}$) of 200 heads of young adults (100 male and 100 female) were used. The laser scans provide head structure data in a cylindrical representation, with radii $r(h, \phi)$ of surface points sampled at 512 equally-spaced angles ϕ, and at 512 equally spaced vertical steps h. Additionally, the RGB-color values $R(h, \phi)$, $G(h, \phi)$,and $B(h, \phi)$, were recorded in the same spatial resolution and were stored in a texture map with 8 bit per channel.

All faces were without makeup, accessories, and facial hair. The subjects were scanned wearing bathing caps, that were removed digitally. Additional automatic pre-processing of the scans, which for most heads required no human interaction, consisted of a vertical cut behind the ears, a horizontal cut to remove the shoulders, and a normalization routine that brought each face to a standard orientation and position in space. The resultant faces were represented by approximately 70,000 vertices and the same number of color values.

3 Morphable 3D Face Model

The morphable model is based on a data set of 3D faces. Morphing between faces requires full correspondence between all of the faces. In this section, we will assume that all exemplar faces are in full correspondence. The algorithm for computing correspondence will be described in Section 5.

We represent the geometry of a face with a shape-vector $S = (X_1, Y_1, Z_1, X_2, ..., Y_n, Z_n)^T \in \Re^{3n}$, that contains the X, Y, Z-coordinates of its n vertices. For simplicity, we assume that the number of valid texture values in the texture map is equal to the number of vertices. We therefore represent the texture of a face by a texture-vector $T = (R_1, G_1, B_1, R_2, ..., G_n, B_n)^T \in \Re^{3n}$, that contains the R, G, B color values of the n corresponding vertices. A morphable face model was then constructed using a data set of m exemplar faces, each represented by its shape-vector S_i and texture-vector T_i. Since we assume all faces in full correspondence (see Section 5), new shapes S_{model} and new textures T_{model} can be expressed in barycentric coordinates as a linear combination of the shapes and textures of the m exemplar faces:

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

We define the morphable model as the set of faces $(S_{mod}(\vec{a}), T_{mod}(\vec{b}))$, parameterized by the coefficients $\vec{a} = (a_1, a_2...a_m)^T$ and $\vec{b} = (b_1, b_2...b_m)^T$. [1] Arbitrary new faces can be generated by varying the parameters $\vec{a}$ and $\vec{b}$ that control shape and texture.

For a useful face synthesis system, it is important to be able to quantify the results in terms of their plausibility of being faces. We therefore estimated the probability distribution for the coefficients a_i and b_i from our example set of faces. This distribution enables us to control the likelihood of the coefficients a_i and b_i and consequently regulates the likelihood of the appearance of the generated faces.

We fit a multivariate normal distribution to our data set of 200 faces, based on the averages of shape $\overline{S}$ and texture $\overline{T}$ and the covariance matrices C_S and C_T computed over the shape and texture differences $\Delta S_i = S_i - \overline{S}$ and $\Delta T_i = T_i - \overline{T}$.

A common technique for data compression known as Principal Component Analysis (PCA) [15, 31] performs a basis transformation to an orthogonal coordinate system formed by the eigenvectors s_i and t_i of the covariance matrices (in descending order according to their eigenvalues)[2]:

$$S_{model} = \overline{S} + \sum_{i=1}^{m-1} \alpha_i s_i\,, \quad T_{model} = \overline{T} + \sum_{i=1}^{m-1} \beta_i t_i\,, \qquad (1)$$

$\vec{\alpha}, \vec{\beta} \in \Re^{m-1}$. The probability for coefficients $\vec{\alpha}$ is given by

$$p(\vec{\alpha}) \sim exp[-\frac{1}{2} \sum_{i=1}^{m-1} (\alpha_i / \sigma_i)^2], \qquad (2)$$

with σ_i^2 being the eigenvalues of the shape covariance matrix C_S. The probability $p(\vec{\beta})$ is computed similarly.

Segmented morphable model: The morphable model described in equation (1), has $m-1$ degrees of freedom for texture and $m-1$ for shape. The expressiveness of the model can be increased by dividing faces into independent subregions that are morphed independently, for example into eyes, nose, mouth and a surrounding region (see Figure 2). Since all faces are assumed to be in correspondence, it is sufficient to define these regions on a reference face. This segmentation is equivalent to subdividing the vector space of faces into independent subspaces. A complete 3D face is generated by computing linear combinations for each segment separately and blending them at the borders according to an algorithm proposed for images by [7] .

[1] Standard morphing between two faces ($m = 2$) is obtained if the parameters a_1, b_1 are varied between 0 and 1, setting $a_2 = 1 - a_1$ and $b_2 = 1 - b_1$.

[2] Due to the subtracted average vectors $\overline{S}$ and $\overline{T}$, the dimensions of $Span\{\Delta S_i\}$ and $Span\{\Delta T_i\}$ are at most $m - 1$.

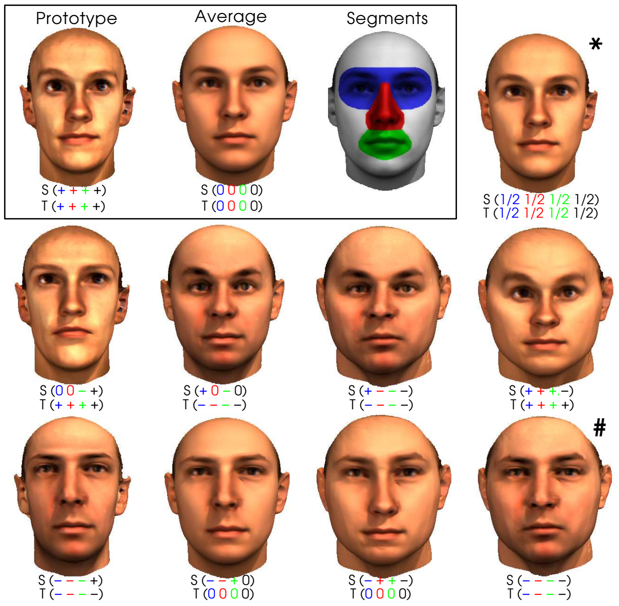

Figure 2: A single prototype adds a large variety of new faces to the morphable model. The deviation of a prototype from the average is added (+) or subtracted (-) from the average. A standard morph (*) is located halfway between average and the prototype. Subtracting the differences from the average yields an 'anti'-face (#). Adding and subtracting deviations independently for shape (S) and texture (T) on each of four segments produces a number of distinct faces.

3.1 Facial attributes

Shape and texture coefficients α_i and β_i in our morphable face model do not correspond to the facial attributes used in human language. While some facial attributes can easily be related to biophysical measurements [13, 10], such as the width of the mouth, others such as facial femininity or being more or less bony can hardly be described by numbers. In this section, we describe a method for mapping facial attributes, defined by a hand-labeled set of example faces, to the parameter space of our morphable model. At each position in face space (that is for any possible face), we define shape and texture vectors that, when added to or subtracted from a face, will manipulate a specific attribute while keeping all other attributes as constant as possible.

In a performance based technique [25], facial expressions can be transferred by recording two scans of the same individual with different expressions, and adding the differences $\Delta S = S_{expression} - S_{neutral}$, $\Delta T = T_{expression} - T_{neutral}$, to a different individual in a neutral expression.

Unlike facial expressions, attributes that are invariant for each individual are more difficult to isolate. The following method allows us to model facial attributes such as gender, fullness of faces, darkness of eyebrows, double chins, and hooked versus concave noses (Figure 3). Based on a set of faces (S_i, T_i) with manually assigned labels μ_i describing the markedness of the attribute, we compute weighted sums

$$\Delta S = \sum_{i=1}^{m} \mu_i (S_i - \overline{S}), \quad \Delta T = \sum_{i=1}^{m} \mu_i (T_i - \overline{T}). \qquad (3)$$

Multiples of $(\Delta S, \Delta T)$ can now be added to or subtracted from any individual face. For binary attributes, such as gender, we assign constant values μ_A for all m_A faces in class A, and $\mu_B \neq \mu_A$ for all m_B faces in B. Affecting only the scaling of ΔS and ΔT, the choice of μ_A, μ_B is arbitrary.

To justify this method, let $\mu(S, T)$ be the overall function describing the markedness of the attribute in a face (S, T). Since $\mu(S, T)$ is not available per se for all (S, T), the regression problem of estimating $\mu(S, T)$ from a sample set of labeled faces has to be solved. Our technique assumes that $\mu(S, T)$ is a linear function. Consequently, in order to achieve a change $\Delta\mu$ of the attribute, there is only a single optimal direction $(\Delta S, \Delta T)$ for the whole space of faces. It can be shown that Equation (3) defines the direction with minimal variance-normalized length $\|\Delta S\|_M^2 = \langle \Delta S, C_S^{-1} \Delta S \rangle$, $\|\Delta T\|_M^2 = \langle \Delta T, C_T^{-1} \Delta T \rangle$.

A different kind of facial attribute is its "distinctiveness", which is commonly manipulated in caricatures. The automated production of caricatures has been possible for many years [6]. This technique can easily be extended from 2D images to our morphable face model. Individual faces are caricatured by increasing their distance from the average face. In our representation, shape and texture coefficients α_i, β_i are simply multiplied by a constant factor.

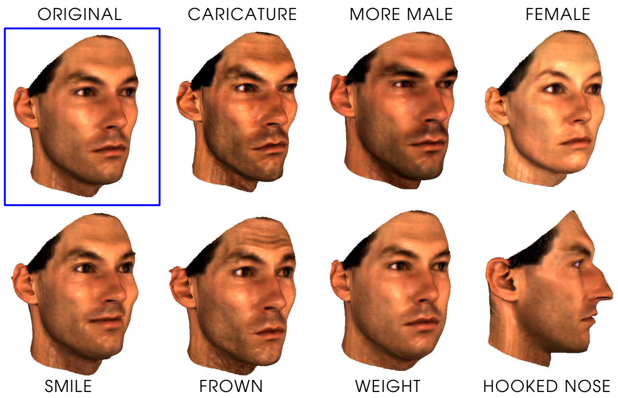

Figure 3: Variation of facial attributes of a single face. The appearance of an original face can be changed by adding or subtracting shape and texture vectors specific to the attribute.

4 Matching a morphable model to images

A crucial element of our framework is an algorithm for automatically matching the morphable face model to one or more images. Providing an estimate of the face's 3D structure (Figure 4), it closes the gap between the specific manipulations described in Section 3.1, and the type of data available in typical applications.

Coefficients of the 3D model are optimized along with a set of rendering parameters such that they produce an image as close as possible to the input image. In an analysis-by-synthesis loop, the algorithm creates a texture mapped 3D face from the current model parameters, renders an image, and updates the parameters according to the residual difference. It starts with the average head and with rendering parameters roughly estimated by the user.

Model Parameters: Facial shape and texture are defined by coefficients α_j and β_j, $j = 1, ..., m - 1$ (Equation 1). Rendering parameters $\vec{\rho}$ contain camera position (azimuth and elevation), object scale, image plane rotation and translation, intensity $i_{r,amb}, i_{g,amb}, i_{b,amb}$ of ambient light, and intensity

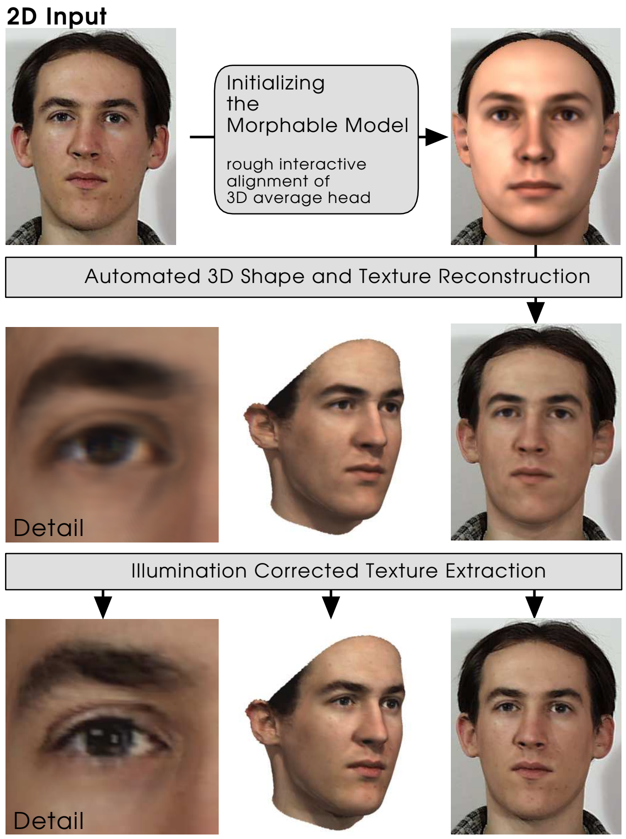

Figure 4: Processing steps for reconstructing 3D shape and texture of a new face from a single image. After a rough manual alignment of the average 3D head (top row), the automated matching procedure fits the 3D morphable model to the image (center row). In the right column, the model is rendered on top of the input image. Details in texture can be improved by illumination-corrected texture extraction from the input (bottom row).

$i_{r,dir}, i_{g,dir}, i_{b,dir}$ of directed light. In order to handle photographs taken under a wide variety of conditions, $\vec{\rho}$ also includes color contrast as well as offset and gain in the red, green, and blue channel. Other parameters, such as camera distance, light direction, and surface shininess, remain fixed to the values estimated by the user.

From parameters $(\vec{\alpha}, \vec{\beta}, \vec{\rho})$, colored images

$$\mathbf{I}_{model}(x, y) = (I_{r,mod}(x, y), I_{g,mod}(x, y), I_{b,mod}(x, y))^T \quad (4)$$

are rendered using perspective projection and the Phong illumination model. The reconstructed image is supposed to be closest to the input image in terms of Euclidean distance

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

Matching a 3D surface to a given image is an ill-posed problem. Along with the desired solution, many non-face-like surfaces lead to the same image. It is therefore essential to impose constraints on the set of solutions. In our morphable model, shape and texture vectors are restricted to the vector space spanned by the database.

Within the vector space of faces, solutions can be further restricted by a tradeoff between matching quality and prior probabilities, using $P(\vec{\alpha})$, $P(\vec{\beta})$ from Section 3 and an ad-hoc estimate of $P(\vec{\rho})$. In terms of Bayes decision theory, the problem is to find the set of parameters $(\vec{\alpha}, \vec{\beta}, \vec{\rho})$ with maximum posterior probability, given an image $\mathbf{I}_{input}$. While $\vec{\alpha}$, $\vec{\beta}$, and rendering parameters $\vec{\rho}$ completely determine the predicted image I_{model}, the observed image I_{input} may vary due to noise. For Gaussian noise with a standard deviation σ_N, the likelihood to observe I_{input} is $p(\mathbf{I}_{input}|\vec{\alpha}, \vec{\beta}, \vec{\rho}) \sim \exp[\frac{-1}{2\sigma_N^2} \cdot E_I]$. Maximum posterior probability is then achieved by minimizing the cost function

$$E = \frac{1}{\sigma_N^2} E_I + \sum_{j=1}^{m-1} \frac{\alpha_j^2}{\sigma_{S,j}^2} + \sum_{j=1}^{m-1} \frac{\beta_j^2}{\sigma_{T,j}^2} + \sum_j \frac{(\rho_j - \bar{\rho}_j)^2}{\sigma_{\rho,j}^2} \quad (5)$$

The optimization algorithm described below uses an estimate of E based on a random selection of surface points. Predicted color values $\mathbf{I}_{model}$ are easiest to evaluate in the centers of triangles. In the center of triangle k, texture $(\bar{R}_k, \bar{G}_k, \bar{B}_k)^T$ and 3D location $(\bar{X}_k, \bar{Y}_k, \bar{Z}_k)^T$ are averages of the values at the corners. Perspective projection maps these points to image locations $(\bar{p}_{x,k}, \bar{p}_{y,k})^T$. Surface normals $\mathbf{n}_k$ of each triangle k are determined by the 3D locations of the corners. According to Phong illumination, the color components $I_{r,model}$, $I_{g,model}$ and $I_{b,model}$ take the form

$$I_{r,model,k} = (i_{r,amb} + i_{r,dir} \cdot (\mathbf{n}_k \mathbf{l}))\bar{R}_k + i_{r,dir} s \cdot (\mathbf{r}_k \mathbf{v}_k)^\nu \quad (6)$$

where $\mathbf{l}$ is the direction of illumination, $\mathbf{v}_k$ the normalized difference of camera position and the position of the triangle's center, and $\mathbf{r}_k = 2(\mathbf{nl})\mathbf{n} - \mathbf{l}$ the direction of the reflected ray. s denotes surface shininess, and ν controls the angular distribution of the specular reflection. Equation (6) reduces to $I_{r,model,k} = i_{r,amb}\bar{R}_k$ if a shadow is cast on the center of the triangle, which is tested in a method described below.

For high resolution 3D meshes, variations in $\mathbf{I}_{model}$ across each triangle $k \in \{1, ..., n_t\}$ are small, so E_I may be approximated by

$$E_I \approx \sum_{k=1}^{n_t} a_k \cdot \|\mathbf{I}_{input}(\bar{p}_{x,k}, \bar{p}_{y,k}) - \mathbf{I}_{model,k}\|^2,$$

where a_k is the image area covered by triangle k. If the triangle is occluded, $a_k = 0$.

In gradient descent, contributions from different triangles of the mesh would be redundant. In each iteration, we therefore select a random subset $\mathcal{K} \subset \{1, ..., n_t\}$ of 40 triangles k and replace E_I by

$$E_{\mathcal{K}} = \sum_{k \in \mathcal{K}} \|\mathbf{I}_{input}(\bar{p}_{x,k}, \bar{p}_{y,k}) - \mathbf{I}_{model,k})\|^2. \quad (7)$$

The probability of selecting k is $p(k \in \mathcal{K}) \sim a_k$. This method of stochastic gradient descent [16] is not only more efficient computationally, but also helps to avoid local minima by adding noise to the gradient estimate.

Before the first iteration, and once every 1000 steps, the algorithm computes the full 3D shape of the current model, and 2D positions $(p_x, p_y)^T$ of all vertices. It then determines a_k, and detects hidden surfaces and cast shadows in a two-pass z-buffer technique. We assume that occlusions and cast shadows are constant during each subset of iterations.

Parameters are updated depending on analytical derivatives of the cost function E, using $\alpha_j \mapsto \alpha_j - \lambda_j \cdot \frac{\partial E}{\partial \alpha_j}$, and similarly for β_j and ρ_j, with suitable factors λ_j.

Derivatives of texture and shape (Equation 1) yield derivatives of 2D locations $(\bar{p}_{x,k}, \bar{p}_{y,k})^T$, surface normals $\mathbf{n}_k$, vectors $\mathbf{v}_k$ and $\mathbf{r}_k$, and $\mathbf{I}_{model,k}$ (Equation 6) using chain rule. From Equation (7), partial derivatives $\frac{\partial E_{\mathcal{K}}}{\partial \alpha_j}$, $\frac{\partial E_{\mathcal{K}}}{\partial \beta_j}$, and $\frac{\partial E_{\mathcal{K}}}{\partial \rho_j}$ can be obtained.

Coarse-to-Fine: In order to avoid local minima, the algorithm follows a coarse-to-fine strategy in several respects:

a) The first set of iterations is performed on a down-sampled version of the input image with a low resolution morphable model.

b) We start by optimizing only the first coefficients α_j and β_j controlling the first principal components, along with all parameters ρ_j. In subsequent iterations, more and more principal components are added.

c) Starting with a relatively large σ_N, which puts a strong weight on prior probability in equation (5) and ties the optimum towards the prior expectation value, we later reduce σ_N to obtain maximum matching quality.

d) In the last iterations, the face model is broken down into segments (Section 3). With parameters ρ_j fixed, coefficients α_j and β_j are optimized independently for each segment. This increased number of degrees of freedom significantly improves facial details.

Multiple Images: It is straightforward to extend this technique to the case where several images of a person are available (Figure 5). While shape and texture are still described by a common set of α_j and β_j, there is now a separate set of ρ_j for each input image. E_I is replaced by a sum of image distances for each pair of input and model images, and all parameters are optimized simultaneously.

Illumination-Corrected Texture Extraction: Specific features of individual faces that are not captured by the morphable model, such as blemishes, are extracted from the image in a subsequent texture adaptation process. Extracting texture from images is a technique widely used in constructing 3D models from images (e.g. [28]). However, in order to be able to change pose and illumination, it is important to separate pure albedo at any given point from the influence of shading and cast shadows in the image. In our approach, this can be achieved because our matching procedure provides an estimate of 3D shape, pose, and illumination conditions. Subsequent to matching, we compare the prediction $\mathbf{I}_{mod,i}$ for each vertex i with $\mathbf{I}_{input}(p_{x,i}, p_{y,i})$, and compute the change in texture (R_i, G_i, B_i) that accounts for the difference. In areas occluded in the image, we rely on the prediction made by the model. Data from multiple images can be blended using methods similar to [28].

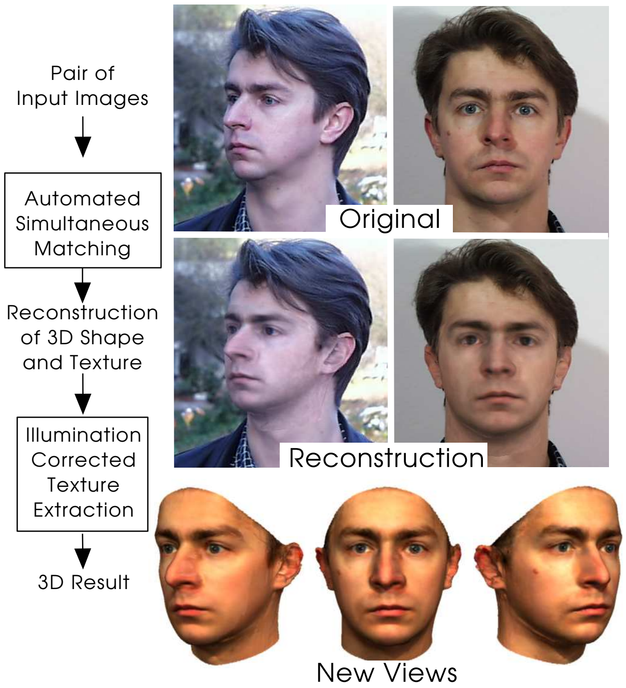

Figure 5: Simultaneous reconstruction of 3D shape and texture of a new face from two images taken under different conditions. In the center row, the 3D face is rendered on top of the input images.

4.1 Matching a morphable model to 3D scans

The method described above can also be applied to register new 3D faces. Analogous to images, where perspective projection

$P : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ and an illumination model define a colored image $\mathbf{I}(x, y) = (R(x, y), G(x, y), B(x, y))^T$, laser scans provide a two-dimensional cylindrical parameterization of the surface by means of a mapping $C : \mathcal{R}^3 \rightarrow \mathcal{R}^2, \quad (x, y, z) \mapsto (h, \phi)$. Hence, a scan can be represented as

$$\mathbf{I}(h, \phi) = (R(h, \phi), G(h, \phi), B(h, \phi), r(h, \phi))^T. \tag{8}$$

In a face (S,T), defined by shape and texture coefficients α_j and β_j (Equation 1), vertex i with texture values (R_i, G_i, B_i) and cylindrical coordinates (r_i, h_i, ϕ_i) is mapped to $\mathbf{I}_{model}(h_i, \phi_i) = (R_i, G_i, B_i, r_i)^T$. The matching algorithm from the previous section now determines α_j and β_j minimizing

$$E = \sum_{h,\phi} \|\mathbf{I}_{input}(h, \phi) - \mathbf{I}_{model}(h, \phi)\|^2.$$

5 Building a morphable model

In this section, we describe how to build the morphable model from a set of unregistered 3D prototypes, and to add a new face to the existing morphable model, increasing its dimensionality.

The key problem is to compute a dense point-to-point correspondence between the vertices of the faces. Since the method described in Section 4.1 finds the best match of a given face only within the range of the morphable model, it cannot add new dimensions to the vector space of faces. To determine residual deviations between a novel face and the best match within the model, as well as to set unregistered prototypes in correspondence, we use an optic flow algorithm that computes correspondence between two faces without the need of a morphable model [35]. The following section summarizes this technique.

5.1 3D Correspondence using Optic Flow

Initially designed to find corresponding points in grey-level images $I(x, y)$, a gradient-based optic flow algorithm [2] is modified to establish correspondence between a pair of 3D scans $\mathbf{I}(h, \phi)$ (Equation 8), taking into account color and radius values simultaneously [35]. The algorithm computes a flow field $(\delta h(h, \phi), \delta\phi(h, \phi))$ that minimizes differences of $\|\mathbf{I}_1(h, \phi) - \mathbf{I}_2(h + \delta h, \phi + \delta\phi)\|$ in a norm that weights variations in texture and shape equally. Surface properties from differential geometry, such as mean curvature, may be used as additional components in $\mathbf{I}(h, \phi)$.

On facial regions with little structure in texture and shape, such as forehead and cheeks, the results of the optic flow algorithm are sometimes spurious. We therefore perform a smooth interpolation based on simulated relaxation of a system of flow vectors that are coupled with their neighbors. The quadratic coupling potential is equal for all flow vectors. On high-contrast areas, components of flow vectors orthogonal to edges are bound to the result of the previous optic flow computation. The system is otherwise free to take on a smooth minimum-energy arrangement. Unlike simple filtering routines, our technique fully retains matching quality wherever the flow field is reliable. Optic flow and smooth interpolation are computed on several consecutive levels of resolution.

Constructing a morphable face model from a set of unregistered 3D scans requires the computation of the flow fields between each face and an arbitrary reference face. Given a definition of shape and texture vectors S_{ref} and T_{ref} for the reference face, S and T for each face in the database can be obtained by means of the point-to-point correspondence provided by $(\delta h(h, \phi), \delta\phi(h, \phi))$.

5.2 Bootstrapping the model

Because the optic flow algorithm does not incorporate any constraints on the set of solutions, it fails on some of the more unusual faces in the database. Therefore, we modified a bootstrapping algorithm to iteratively improve correspondence, a method that has been used previously to build linear image models [36].

The basic recursive step: Suppose that an existing morphable model is not powerful enough to match a new face and thereby find correspondence with it. The idea is first to find rough correspondences to the novel face using the (inadequate) morphable model and then to improve these correspondences by using an optic flow algorithm.

Starting from an arbitrary face as the temporary reference, preliminary correspondence between all other faces and this reference is computed using the optic flow algorithm. On the basis of these correspondences, shape and texture vectors S and T can be computed. Their average serves as a new reference face. The first morphable model is then formed by the most significant components as provided by a standard PCA decomposition. The current morphable model is now matched to each of the 3D faces according to the method described in Section 4.1. Then, the optic flow algorithm computes correspondence between the 3D face and the approximation provided by the morphable model. Combined with the correspondence implied by the matched model, this defines a new correspondence between the reference face and the example.

Iterating this procedure with increasing expressive power of the model (by increasing the number of principal components) leads to reliable correspondences between the reference face and the examples, and finally to a complete morphable face model.

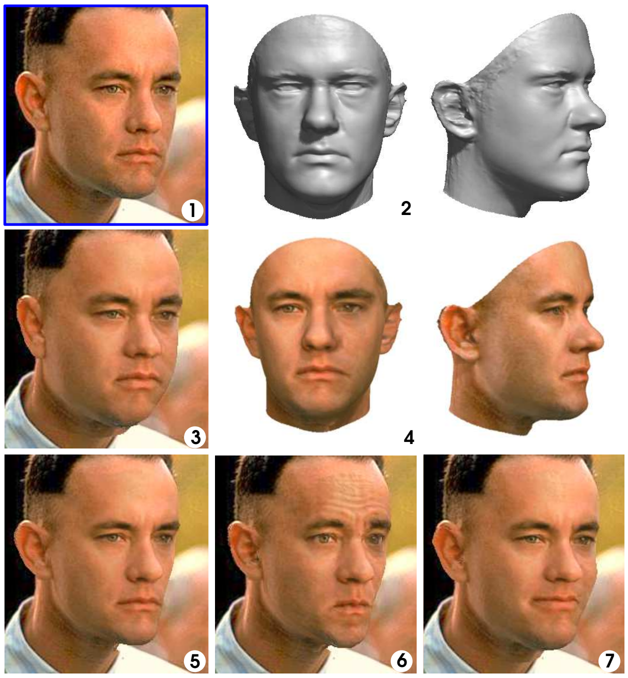

Figure 6: Matching a morphable model to a single image (1) of a face results in a 3D shape (2) and a texture map estimate. The texture estimate can be improved by additional texture extraction (4). The 3D model is rendered back into the image after changing facial attributes, such as gaining (3) and loosing weight (5), frowning (6), or being forced to smile (7).

6 Results

We built a morphable face model by automatically establishing correspondence between all of our 200 exemplar faces. Our interactive

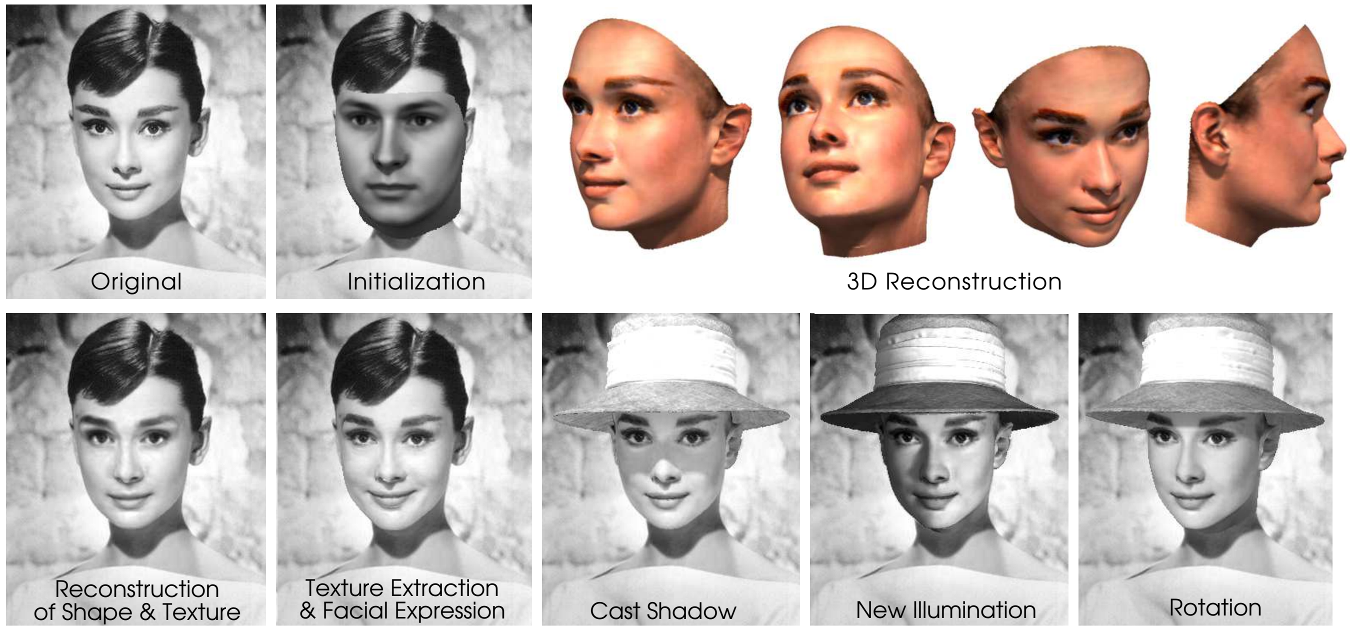

Figure 7: After manual initialization, the algorithm automatically matches a colored morphable model (color contrast set to zero) to the image. Rendering the inner part of the 3D face on top of the image, new shadows, facial expressions and poses can be generated.

face modeling system enables human users to create new characters and to modify facial attributes by varying the model coefficients. Within the constraints imposed by prior probability, there is a large variability of possible faces, and all linear combinations of the exemplar faces look natural.

We tested the expressive power of our morphable model by automatically reconstructing 3D faces from photographs of arbitrary Caucasian faces of middle age that were not in the database. The images were either taken by us using a digital camera (Figures 4, 5), or taken under arbitrary unknown conditions (Figures 6, 7).

In all examples, we matched a morphable model built from the first 100 shape and the first 100 texture principal components that were derived from the whole dataset of 200 faces. Each component was additionally segmented in 4 parts (see Figure 2). The whole matching procedure was performed in 10^5 iterations. On an SGI R10000 processor, computation time was 50 minutes.

Reconstructing the true 3D shape and texture of a face from a single image is an ill-posed problem. However, to human observers who also know only the input image, the results obtained with our method look correct. When compared with a real image of the rotated face, differences usually become only visible for large rotations of more than 60°.

There is a wide variety of applications for 3D face reconstruction from 2D images. As demonstrated in Figures 6 and 7, the results can be used for automatic post-processing of a face within the original picture or movie sequence.

Knowing the 3D shape of a face in an image provides a segmentation of the image into face area and background. The face can be combined with other 3D graphic objects, such as glasses or hats, and then be rendered in front of the background, computing cast shadows or new illumination conditions (Fig. 7). Furthermore, we can change the appearance of the face by adding or subtracting specific attributes. If previously unseen backgrounds become visible, we fill the holes with neighboring background pixels (Fig. 6).

We also applied the method to paintings such as Leonardo's Mona Lisa (Figure 8). Due to unusual (maybe unrealistic) lighting, illumination-corrected texture extraction is difficult here. We therefore apply a different method for transferring all details of the painting to novel views. For new illumination, we render two images of the reconstructed 3D face with different illumination, and multiply relative changes in pixel values (Figure 8, bottom left) by the original values in the painting (bottom center). For a new pose (bottom right), differences in shading are transferred in a similar way, and the painting is then warped according to the 2D projections of 3D vertex displacements of the reconstructed shape.

7 Future work

Issues of implementation: We plan to speed up our matching algorithm by implementing a simplified Newton-method for minimizing the cost function (Equation 5). Instead of the time consuming computation of derivatives for each iteration step, a global mapping of the matching error into parameter space can be used [9].

Data reduction applied to shape and texture data will reduce redundancy of our representation, saving additional computation time.

Extending the database: While the current database is sufficient to model Caucasian faces of middle age, we would like to extend it to children, to elderly people as well as to other races.

We also plan to incorporate additional 3D face examples representing the time course of facial expressions and visemes, the face variations during speech.

The laser scanning technology we used, unfortunately, does not allow us to collect dynamical 3D face data, as each scanning cycle takes at least 10 seconds. Consequently, our current example set of facial expressions is restricted to those that can be kept static by the scanned subjects. However, the development of fast optical 3D digitizers [27] will allow us to apply our method to streams of 3D data during speech and facial expressions.

Extending the face model: Our current morphable model is restricted to the face area, because a sufficient 3D model of hair cannot be obtained with our laser scanner. For animation, the missing part of the head can be automatically replaced by a standard hair style or a hat, or by hair that is modeled using interactive manual segmentation and adaptation to a 3D model [30, 28]. Automated reconstruction of hair styles from images is one of the future challenges.

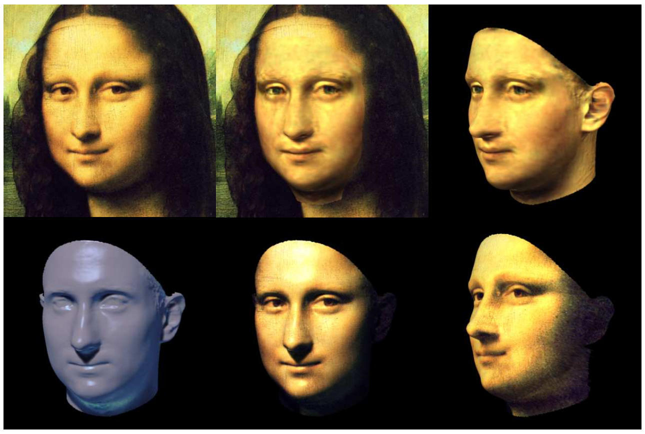

Figure 8: Reconstructed 3D face of Mona Lisa (top center and right). For modifying the illumination, relative changes in color (bottom left) are computed on the 3D face, and then multiplied by the color values in the painting (bottom center). Additional warping generates new orientations (bottom right, see text), while details of the painting, such as brush strokes or cracks, are retained.

8 Acknowledgment

We thank Michael Langer, Alice O'Toole, Tomaso Poggio, Heinrich Bülthoff and Wolfgang Straßer for reading the manuscript and for many insightful and constructive comments. In particular, we thank Marney Smyth and Alice O'Toole for their perseverance in helping us to obtain the following. **Photo Credits:** Original image in Fig. 6: Courtesy of Paramount/VIACOM. Original image in Fig. 7: MPTV/interTOPICS.

References

[1] T. Akimoto, Y. Suenaga, and R.S. Wallace. Automatic creation of 3D facial models. *IEEE Computer Graphics and Applications*, 13(3):16–22, 1993.

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

[3] P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements. In *Advanced Computer Animation, SIGGRAPH '85 Tutorials*, volume 2, pages 61–79, New York, 1985. ACM.

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

[5] D. Beymer, A. Shashua, and T. Poggio. Example-based image analysis and synthesis. A.I. Memo No. 1431, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[6] S. E. Brennan. The caricature generator. *Leonardo*, 18:170–178, 1985.

[7] 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.

[8] 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 Syatems (ISCAS91)*, pages 2665–2668, 1991.

[9] T.F. Cootes, G.J. Edwards, and C.J. 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.

[10] D. DeCarlos, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 67–74, 1998.

[11] S. DiPaola. Extending the range of facial types. *Journal of Visualization and Computer Animation*, 2(4):129–131, 1991.

[12] G.J. Edwards, A. Lanitis, C.J. Taylor, and T.F. Cootes. Modelling the variability in face images. In *Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition*, IEEE Comp. Soc. Press, Los Alamitos, CA, 1996.

[13] L.G. Farkas. *Anthropometry of the Head and Face*. RavenPress, New York, 1994.

[14] B. Guenter, C. Grimm, D. Wolf, H. Malvar, and F. Pighin. Making faces. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 55–66, 1998.

[15] I.T. Jollife. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[16] M. Jones and T. Poggio. Multidimensional morphable models: A framework for representing and matching object classes. In *Proceedings of the Sixth International Conference on Computer Vision*, Bombay, India, 1998.

[17] R. M. Koch, M. H. Gross, and A. A. Bosshard. Emotion editing using finite elements. In *Proceedings of the Eurographics '98, COMPUTER GRAPHICS Forum, Vol. 17, No. 3*, pages C295–C302, Lisbon, Portugal, 1998.

[18] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.

[19] Y.C. Lee, D. Terzopoulos, and Keith Waters. Constructing physics-based facial models of individuals. *Visual Computer*, Proceedings of Graphics Interface '93:1–8, 1993.

[20] Y.C. Lee, D. Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *SIGGRAPH '95 Conference Proceedings*, pages 55–62, Los Angels, 1995. ACM.

[21] J. P. Lewis. Algorithms for solid noise synthesis. In *SIGGRAPH '89 Conference Proceedings*, pages 263–270. ACM, 1989.

[22] N. Magneneat-Thalmann, H. Minh, M. Angelis, and D. Thalmann. Design, transformation and animation of human faces. *Visual Computer*, 5:32–39, 1989.

[23] L. Moccozet and N. Magnenat-Thalmann. Dirichlet free-form deformation and their application to hand simulation. In *Computer Animation'97*, 1997.

[24] F. I. Parke and K. Waters. *Computer Facial Animation*. AKPeters, Wellesley, Massachusetts, 1996.

[25] F.I. Parke. Computer generated animation of faces. In *ACM National Conference*. ACM, November 1972.

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

[27] M. Petrow, A. Talapov, T. Robertson, A. Lebedev, A. Zhilyaev, and L. Polonskiy. Optical 3D digitizer: Bringing life to virtual world. *IEEE Computer Graphics and Applications*, 18(3):28–37, 1998.

[28] F. Pighin, J. Hecker, D. Lischinski, Szeliski R, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 75–84, 1998.

[29] S. Platt and N. Badler. Animating facial expression. *Computer Graphics*, 15(3):245–252, 1981.

[30] G. Sannier and N. Magnenat-Thalmann. A user-friendly texture-fitting methodology for virtual humans. In *Computer Graphics International'97*, 1997.

[31] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4:519–554, 1987.

[32] D. Terzopoulos and Keith Waters. Physically-based facial modeling, analysis, and animation. *Visualization and Computer Animation*, 1:73–80, 1990.

[33] Demetri Terzopoulos and Hong Qin. Dynamic NURBS with geometric constraints to interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, April 1994.

[34] J. T. Todd, S. M. Leonard, R. E. Shaw, and J. B. Pittenger. The perception of human growth. *Scientific American*, 1242:106–114, 1980.

[35] T. Vetter and V. Blanz. Estimating coloured 3d face models from single images: An example based approach. In Burkhardt and Neumann, editors, *Computer Vision – ECCV'98 Vol. II*, Freiburg, Germany, 1998. Springer, Lecture Notes in Computer Science 1407.

[36] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition – CVPR'97*, Puerto Rico, USA, 1997. IEEE Computer Society Press.

[37] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

[38] Keith Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics*, 22(4):17–24, 1987.