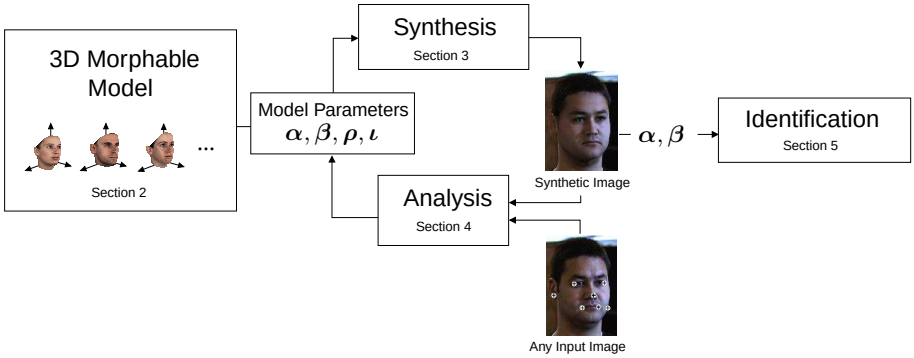
Chapter 10. Morphable Models of Faces

Sami Romdhani[1], Volker Blanz[2], Curzio Basso[1], and Thomas Vetter[1]

[1] University of Basel, Computer Science Department, Bernoullistrasse 16, CH - 4056 Basel, Switzerland {sami.romdhani, curzio.basso, thomas.vetter}@unibas.ch
[2] Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany blanz@mpi-sb.mpg.de



1 Morphable Model for Face Analysis

Our approach is based on an *Analysis by synthesis* framework. The idea of this framework is to synthesize an image of a face which resembles to the face in an input image. This framework requires a generative model able to accurately synthesize face images. Then the parameters of the image generated by the model are used for high-level tasks such as identification.

To be applicable on any input face image, a good model must be able to generate any face images. Face images vary widely with respect to the imaging conditions (illumination and angle from which the face is viewed, called pose) and with respect to the identity and the expression of the face. A generative model must not only allow for these variations but must also clearly separate the source of variations in order to make, say, identification tasks invariant to the other sources of variation.

We explain in this section that a 3D representation enables both the accurate modeling of any illumination and pose and also the separation of these variations from the rest (identity and expression). The generative model must be able to synthesize images from any individual. In a Morphable Model, the identity variation is modeled by making linear combinations of faces of a small set of persons. In this section, we show why linear

combinations yield a realistic face only if the set of example faces is in correspondence. A good generative model should be restrictive in the sense that unlikely faces should be rarely instantiated. To achieve this the probability density function of human faces must be learned and used as a prior to synthesize faces.

Based on these principles, we detail the construction of a 3D Morphable Face Model in Section 2. The main step of the model construction is the computation of the correspondences of a set of example 3D laser scan of faces with a reference 3D face laser scan. We also introduce the *Regularized* Morphable Model which improves the correspondences. The synthesis of a face image from the model is presented in Section 3.

The generative model is half of the work. The other half is the analyzing algorithm (a.k.a. the *fitting algorithm*). The fitting algorithm finds the parameters of the model that synthesize an image as close as possible to the input image. We detail two fitting algorithms in Section 4. The main difference between these two algorithms is their trade-off between accuracy and computational load. Based on these two fitting algorithms identification results are presented in Section 5 for face images varying in illumination and pose.

1.1 Three dimensional representation

Each individual face can generate a variety of different images. This huge diversity of face images makes the analysis of these difficult. Besides the general differences between individual faces the appearance variations in images of a single faces can be separated into the following four sources:

- Pose changes can result in dramatic changes in images. Due to occlusions different parts of the object become visible or invisible. Additionally, the parts seen in two views change their spatial configuration relative to each other.
- Illumination changes influence the appearance of a face even if the pose of the face is fixed. Positions and distribution of light sources around a face have the effect of changing the brightness distribution in the images, the locations of attached shadows and specular reflections. Additionally, cast shadows can generate prominent contours in facial images.
- Facial expressions an important tool in human communication are another source of variations in images. Only a few facial landmarks which are directly coupled with the bony structure of the skull like the interoccular distance or the general position of the ears are constant in a face. Most other features can change their spatial configuration or position due to the articulation of the jaw or to muscle action, like moving eyebrows, lips or cheeks.
- In the long term a face changes due to aging, to a changing hairstyle or according to makeup or accessories.

The isolation and explicit description of all these different sources of variations must be the ultimate goal of a face analysis system. For example it is desirable that the parameters which code the identity of a person are not perturbed by a modification of pose. In an analysis by synthesis framework this implies that a face model must account for each of these variations independently by explicit parameters.

The main challenge for the design of such systems is to find or choose a description of these parameters that allows both, the appropriate modeling of images on one side and gives a precise description of an image on the other.

Some of the sources of variations, such as illumination and pose, obey to the physical laws of nature. These laws reflect constraints derived from the three-dimensional geometry of faces and the interaction of their surfaces with light. They are optimally imposed by a 3D representation that was therefore chosen for the Morphable Model.

On the other hand there are additional regularities between faces that are not formulated as physical laws, but can be obtained by exploiting the general statistics of faces. These methods are also denoted as learning from examples. It is expected that learning schemes that conform or incorporate the physical constraints are more successful in tasks like generalizing from a single image of a face to novel views or to different illumination conditions.

As a result, the 3D Morphable Model uses physical laws to model pose and illumination and statistical methods to model identity and expression. As we see in the next two sections, these statistical methods require the faces to be put into correspondence.

1.2 Correspondence based representation

For ease of visualization, we motivate the correspondence based representation on a 2D image example, the same argument can be made on 3D faces. As seen in other chapters of this book, the crucial assumption of most of the model-based face analysis techniques is that any face image can be generated by linear combinations of few faces. However, linear operations like a simple addition of raw images, pixel by pixel, is not very meaningful as shown in Figure 1. Already the average of two images of a face does not result in the image of a face. Instead the average appears blurry with double contours. Hence, face images in their pixel representation do not form a vector space. For a correct image modeling and synthesis it is not sufficient to consider only image intensities, it is also necessary to consider the spatial locations of object features. That is, the correspondence between images has to be established. Only a separate addition of the shape and texture information satisfies the vector space requirements. Hence, shape alone form a vector space and texture alone form another vector space. The face vector space is the combination of these two vector spaces.

So, correspondence separates texture information from two-dimensional shape information in an image. Correspondence is the basic requirement for the modeling of face images in a vector space. The utilization of correspondence for image modeling was proposed by several authors [11, 5, 18, 15, 33] for a review see [4]. The common feature of these methods is that they all derive their face model, used for the analysis and synthesis of face images, from separate texture and shape vector spaces. It should be noted that some of these approaches do not extract the correspondence of *physical* points: [18], for instance, put into correspondence the 2D occluding contour, which varies from pose to pose. In contrast, our approach puts in correspondence 3D points that are equivalent across object instances.

Another representation, opposed to the correspondence based representation is the *Appearance based representation*. This representation is used by methods known as eigenface

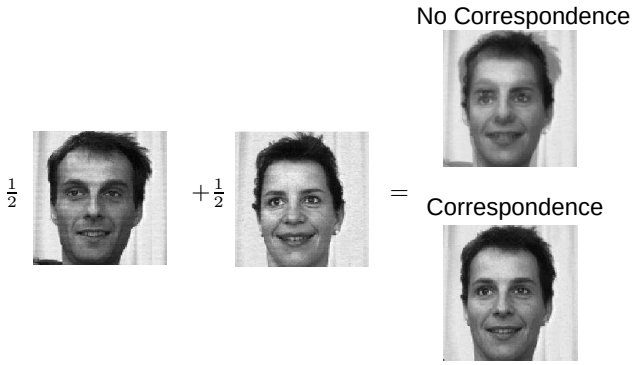


Fig. 1. Computing the average of two face images using different image representations. No correspondence information is used (Top right) and using correspondence (Bottom right).

techniques [28, 30] and their generalized version introduced as appearance based models [22]. These techniques have been demonstrated to be successful when applied to images of a set of very different objects or, as in the case of faces, when viewpoint or lighting conditions are kept close to constant. Contrasting with our approach these appearance based models do not rely on shape features, but rather only on pixel features and are also learned by exploring the statistics of example data represented in the pixel space of the images. However, none of these techniques uses correspondences that exploits an object centered representation.

In contrast to the techniques which are based on point correspondence between images or objects these appearance based models use linear combinations of images directly. This lack of correspondence information between the images is the crucial disadvantage of these methods. It disables the methods from any generalization abilities which go beyond a direct comparison of a novel image to stored images. Due to the truncated representation in a principal component space (see Section 1.3) and the missing correspondences these techniques have very limited image synthesis capabilities. Already the mixture of two different instances of the same object results in an extremely blurry image (see Figure 1). While we show the advantages of a correspondence based representation on a 2D image example, the validity of the argument extends to 3D faces.

1.3 Face Statistics

In the previous section, we explained that correspondences enable the generation of faces by simple linear combination of several faces. However, the coefficients of the linear combination do not have a uniform distribution. This distribution is learned from example faces using the currently widely accepted assumption that the face space is Gaussian. Under this assumption, PCA is used to learn a probability model of faces that is used as prior at the analysis step (see Section 4.1). More details about the face statistics of our model are given in Sections 2.3 and 2.4.

2 3D Morphable Model Construction

The construction of a 3D Morphable Model requires a set of examples 3D faces (e.g. laser scans). The results presented in this paper were obtained with a Morphable Model constructed with 200 laser scans acquired by a *Cyberware*$^{\mathrm{TM}}$ 3030PS laser scanner. The constructions is performed in three steps: First, the laser scans are preprocessed. This semi-automatic step aims to remove the scanning artifacts and to select the part of the head which is to be modeled (from one ear to the other and from the neck to the forehead). In the second step, the correspondences are computed between one scan, chosen as reference, and each of the other scans. Then a Principal Components Analysis is performed to learn statistics of the 3D shape and color of the faces.

In Section 2.4, a novel procedure to construct a *Regularized* Morphable Model is introduced which yields better correspondences. It will be shown in Section 5.1 that the Regularized Morphable Model improves the identification performance. The original Morphable Model computes correspondences between a pair of laser scans. However, the Regularized Morphable Model uses a prior derived from other face scans in order to constraint the correspondences. This can be seen as simultaneously putting a laser scan in correspondences with *a set* of laser scans.

2.1 Preprocessing of the laser scans

This semi-automatic step aims to remove the scanning artifacts and to select the part of the head which is to be modeled. It is performed by three consecutive stages:

1. Holes are filled and spikes (i.e. scan artifacts) removed using an interactive tool.
2. The faces are aligned in 3D using the 3D-3D Absolute Orientation method [16].
3. The parts of the head that we do not wish to model, such as the back of the head behind the ears, the hair area and the region underneath the throat are trimmed.

2.2 Dense correspondences computed by optical flow

To automatically compute dense point-to-point correspondences between two 3D laser scans of faces, we use optical flow. The scans are recorded by a laser scanner, measuring the radius (i.e. depth), r, along with the color R, G, B of faces. Optical flow is computed on a cylindrical representation, $\mathbf{I}(h, \phi)$, of the colored 3D scans:

$$\mathbf{I}(h, \phi) = (r(h, \phi), R(h, \phi), G(h, \phi), B(h, \phi)). \tag{1}$$

Correspondences are given by a dense vector field $\mathbf{v}(h, \phi) = (\Delta h(h, \phi), \Delta\phi(h, \phi))$, such that each point of the first scan, $\mathbf{I}_1(h, \phi)$, corresponds to the point $\mathbf{I}_2(h + \Delta h, \phi + \Delta\phi)$ on the second scan. A modified optical flow algorithm [7] is used to estimate this vector field.

Optical flow on gray-level images

Many optical flow algorithm (e.g. [17], [19], [3]) are based on the assumption that objects in an image sequence $I(x, y, t)$ conserve their brightness as they move across the images at a velocity $(v_x, v_y)^{\mathrm{T}}$:

$$\frac{dI}{dt} = v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0. \tag{2}$$

For a pair of images, I_1 and I_2, taken at two discrete moments, the temporal derivatives, v_x, v_y and $\frac{\partial I}{\partial t}$, 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, optical 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 [19, 3] is minimized at each point (x_0, y_0):

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

We used a 5×5 pixel neighborhood $R(x_0, y_0)$. In each point (x_0, y_0), $\mathbf{v}(x_0, y_0)$ can be found by solving a 2×2 linear system (see [8] for details). In order to deal with large displacements $\mathbf{v}$, the algorithm of [3] employs a coarse-to-fine strategy using a Gaussian pyramid 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 3D colored surfaces

For processing 3D laser scans $\mathbf{I}(h, \phi)$, Equation (3) is replaced by

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

$$\text{with a norm } \|\mathbf{I}\|^2 = w_r r^2 + w_R R^2 + w_G G^2 + w_B B^2. \tag{5}$$

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 2×2 linear system (see [8]).

Correspondences between scans of different individuals, who may differ in overall brightness and size, are improved by using Laplacian pyramids (band-pass filtering) rather than Gaussian pyramids (low-pass filtering). 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 [8] is added to the matching procedure on each level of resolution.

2.3 Face Space based on Principal Components Analysis

We mentioned that the correspondences enable the formulation of a face space. The face space is constructed by putting a set of M examples 3D laser scans into correspondences with a reference laser scan. This introduces a consistent labeling of all N_v 3D vertices across all the scans. The shape and texture surfaces are parameterized in the (u, v) reference frame where one pixel correspond to one 3D vertex (see Figure 2). The 3D position in Cartesian coordinates of the N_v vertices of a face scan are arranged in a shape matrix, $\mathbf{S}$, and their color in a texture matrix, $\mathbf{T}$:

$$\mathbf{S} = \begin{pmatrix} x_1 & x_2 & \cdots & x_{N_v} \\ y_1 & y_2 & \cdots & y_{N_v} \\ z_1 & z_2 & \cdots & z_{N_v} \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} r_1 & r_2 & \cdots & r_{N_v} \\ g_1 & g_2 & \cdots & g_{N_v} \\ b_1 & b_2 & \cdots & b_{N_v} \end{pmatrix}. \tag{6}$$

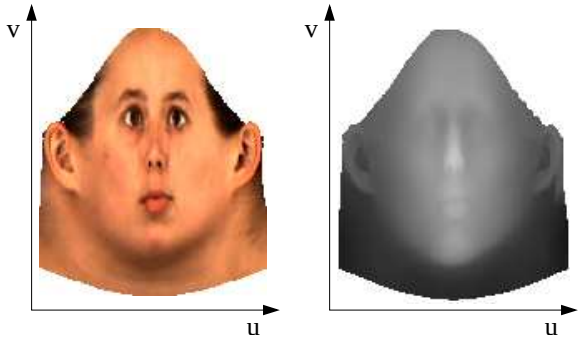


Fig. 2. Texture and shape in the reference space (u, v).

Having constructed a linear face space, we can make linear combinations of the shapes, $\mathbf{S}_i$, and of the textures, $\mathbf{T}_i$ of M example individuals to produce faces of new individuals:

$$\mathbf{S} = \sum_{i=1}^{M} \alpha_i \cdot \mathbf{S}_i, \qquad \mathbf{T} = \sum_{i=1}^{M} \beta_i \cdot \mathbf{T}_i. \tag{7}$$

Equation (7) assumes a uniform distribution of the shapes and the textures. We know that this distribution yields a model that is not restrictive enough: if some α_i or β_i are $\gg 1$, the face produced is unlikely. Therefore, we assume that the shape and the texture spaces have a Gaussian probability distribution function. Principal Component Analysis (PCA) is a statistical tool which transforms the space such that the covariance matrix is diagonal (i.e. it de-correlates the data). PCA is applied separately on the shape and texture spaces, thereby ignoring the correlation between shape and texture as opposed to other techniques (see Section 2.3 of Chapter 3). We describe the application of PCA to shapes, its application to textures is straightforward. After subtracting their average, $\overline{\mathbf{S}}$, the exemplars are arranged in a data matrix $\mathbf{A}$ and the eigenvectors of its covariance matrix $\mathbf{C}$ are computed using the Singular Value Decomposition [25] of $\mathbf{A}$:

$$\overline{\mathbf{S}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{S}_i, \qquad \mathbf{a}_i = \mathrm{vec}(\mathbf{S}_i - \overline{\mathbf{S}}), \qquad \mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M) = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}},$$
$$\mathbf{C} = \frac{1}{M}\mathbf{A}\mathbf{A}^{\mathrm{T}} = \frac{1}{M}\mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^{\mathrm{T}} \tag{8}$$

$\mathrm{vec}(\mathbf{S})$ vectorizes $\mathbf{S}$ by stacking its columns. The M columns of the orthogonal matrix $\mathbf{U}$ are the eigenvectors of the covariance matrix $\mathbf{C}$, and $\sigma_i^2 = \frac{\lambda_i^2}{M}$ are its eigenvalues, where the λ_i are the elements of the diagonal matrix $\mathbf{\Lambda}$, arranged in decreasing order. Let us denote $\mathbf{U}_{\cdot,i}$, the column i of $\mathbf{U}$, and the principal component i, reshaped into a $3 \times N_v$ matrix, by $\mathbf{S}^i = \mathbf{U}^{(3)}_{\cdot,i}$. The notation $\mathbf{a}^{(n)}_{m\times 1}$ (see [21]) folds the $m \times 1$ vector $\mathbf{a}$ into an $n \times (m/n)$ matrix.

Now, instead of describing a novel shape and texture as a linear combination of examples, as in Equation 7, we express them as a linear combination of N_S shape and N_T texture principal components:

$$\mathbf{S} = \overline{\mathbf{S}} + \sum_{i=1}^{N_S}\alpha_i \cdot \mathbf{S}^i, \qquad \mathbf{T} = \overline{\mathbf{T}} + \sum_{i=1}^{N_T}\beta_i \cdot \mathbf{T}^i, \tag{9}$$

The advantage of this formulation is that the probabilities of a shape and a texture are readily available:

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

2.4 Regularized Morphable Model

The correspondence estimation computed by optical flow, detailed in Section 2.2, may, for some laser scan, be wrong in some regions. These bad correspondences are visualized on shape caricatures (i.e. faces for which the shape is obtained by multiplying the flow field by a coefficient higher than 1), as seen on the top row of Figure 3. These problems arise because the scans' boundaries are artificially set and sometimes some of these boundaries do not correspond with the boundaries set on other scans. This lead to errors in correspondences estimated by optical flow. In this section, we present a scheme aiming to improve the correspondences by regularizing them using statistics derived from scans that do not present correspondence errors. This is achieved by modifying the model construction in two ways: First, Probabilistic PCA [29] is used instead of PCA, which regularizes the model by allowing the exemplars to be noisy. Second, a bootstrapping technique is used whereby the model is iteratively estimated.

Probabilistic PCA

Instead of assuming a linear model for the shape, as in previous section, we assume a linear Gaussian model:

$$\mathrm{vec}\,\mathbf{S} = \mathrm{vec}\,\overline{\mathbf{S}} + \mathbf{C}_S \cdot \boldsymbol{\alpha} + \boldsymbol{\epsilon} \tag{11}$$

where $\mathbf{C}_S$, whose columns are the regularized shape principal components, has dimensions $3N_v \times N_S$, and the shape coefficients $\boldsymbol{\alpha}$ and the noise $\boldsymbol{\epsilon}$ have a Gaussian distribution with zero mean and covariance $\mathbf{I}$ and $\sigma^2\mathbf{I}$, respectively.

Fig. 3. Correspondences artifacts are visualized by making shape caricatures (i.e. extending the shape deviations from the average). In the top row, three caricatures yielded by the original Morphable Model are shown. In the bottom row, caricatures of the same scans obtained by the Regularized Morphable Model are presented. Clearly, the bottom row is less perturbed by correspondences artifacts.

[29] use the EM-Algorithm [12] to iteratively estimate $\mathbf{C}_S$ and the projection of the example vectors to the model, $\mathbf{K} = [\boldsymbol{\alpha}_1 \boldsymbol{\alpha}_2 \dots \boldsymbol{\alpha}_M]$. The algorithm starts with $\mathbf{C}_S = \mathbf{A}$, and then at each iteration it computes a new estimate of the shape coefficients $\mathbf{K}$ (expectation step, or *e-step*) and of the regularized principal components $\mathbf{C}_S$ (maximization step, or *m-step*). The coefficients of the example shapes, the unobserved variables, are estimated at the *e-step*:

$$\mathbf{K} = \mathbf{B}^{-1}\mathbf{C}_S^{\mathrm{T}}\mathbf{A} \quad \text{with} \quad \mathbf{B} = \mathbf{C}_S^{\mathrm{T}}\mathbf{C}_S + \sigma^2\mathbf{I} \tag{12}$$

This is the maximum a posteriori estimator of $\mathbf{K}$, that is, the expected value of $\mathbf{K}$ given the posterior distribution $p(\boldsymbol{\alpha}|\mathbf{C}_S)$. At the *m-step*, the model is estimated by computing the $\mathbf{C}_S$ which maximizes the likelihood of the data, given the current estimate of $\mathbf{K}$ and $\mathbf{B}$:

$$\mathbf{C}_S = \mathbf{A} \cdot \mathbf{K}^{\mathrm{T}} \cdot \left(\sigma^2 \cdot M \cdot \mathbf{B}^{-1} + \mathbf{K} \cdot \mathbf{K}^{\mathrm{T}}\right)^{-1} \tag{13}$$

These two steps are iterated in sequence until the algorithm is judged to have converged.

In the original algorithm the value of σ^2 is also estimated at the m-step as

$$\sigma^2 = \frac{1}{3N_v \cdot M} \operatorname{tr}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}} - \mathbf{C}_s\mathbf{K}\mathbf{A}^{\mathrm{T}}\right) \tag{14}$$

but in our case, with $M \ll 3N_v$, this would yield an estimated value of zero. Therefore, we prefer to estimate σ^2 by replacing $\mathbf{A}$ and $\mathbf{K}$ in the Equation (14) with a data matrix of test vectors (vectors not used in estimating $\mathbf{C}_s$) and its corresponding coefficients matrix obtained via Equation (12). If a test set is not available, we can still get an estimate of σ^2 by cross validation.

Bootstrapping

The bootstrapping aims to constraint the correspondences using face statistics. The idea, which originated in [32] and which was applied to 3D Morphable Model in [7], is to put some of the 3D scans in correspondence with optical flow as it is explained in Section 2.2. Then a Morphable Model is constructed using only the subset of the laser scans for which the correspondences are deemed to be correct by a human expert. The rest of the laser scans (those for which the correspondences are not good enough) are fitted to the model in a similar method as the one explained in Section 4. The method is different though, as here the Morphable Model is fitted to a 3D laser scan and not to a 2D image. The fitting produces, for each laser scan, a face which is in correspondences with the reference face and which more resembles to the original scan than the reference does. This face is called the approximation (see Figure 4). Then the correspondences between the approximations and its original face are computed by optical flow. These correspondences are more likely to be good than the one computed between the reference face and the original. This procedure is applied iteratively. At each iterations the faces for which correspondences are judged adequate are added to the morphable model. The process stops when all the correspondences are judge satisfactory.

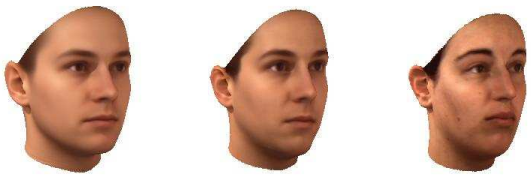


Fig. 4. Images of rendering of 3D lasers scans. The first is the reference with which correspondences are to be estimated. The second is the fitting of the original with a Morphable Model constructed with a subset of the scan for which correspondences were correct. The last one is the original scan.

2.5 Segmented Morphable Model

As mentioned, our Morphable Model is derived from statistics computed on 200 example faces. As a result, the dimensions of the shape and texture spaces, N_S and N_T, are limited to 199. This might not be enough to account for the rich variations of individualities present in mankind. Naturally, one way to augment the dimension of the face space would be to use 3D scans of more persons. However these are not available. Hence we resort to another scheme: We segment the face in four regions (nose, eyes, mouth and the rest) and use separate set of shape and texture coefficients to code them (see [7]). This method multiplies by four the expressiveness of the morphable model. The fitting results in Section 4 and the identification results in Section 5 are based on a segmented Morphable Model with

$N_S = N_T = 100$ for all segments. In the rest of the chapter, we denote the shape and texture parameters by $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ when they can be used interchangeably for the global and the segmented parts of the model. When we want to distinguish them, we use, for the shape parameters, $\boldsymbol{\alpha}^g$ for the global model (full face) and $\boldsymbol{\alpha}^{s_1}$ to $\boldsymbol{\alpha}^{s_4}$ for the segmented parts (the same notation is used for the texture parameters).

3 A Morphable Model to Synthesize Images

One part of the analysis by synthesis loop is the synthesis, i.e. the generation of accurate face images viewed from any pose and illuminated by any condition. This process is explained in this section.

3.1 Shape Projection

To render the image of a face, the 3D shape must be projected to the 2D image frame. This is performed in two steps. First a 3D rotation and translation (i.e. a rigid transformation) maps the object-centered coordinates, $\mathbf{S}$, to a position relative to the camera:

$$\mathbf{W} = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \mathbf{S} + \mathbf{t_w} \mathbf{1}_{1\times N_v} \tag{15}$$

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 3D translation. A projection then maps a vertex k to the image plane in (x_j, y_j). We typically use two types of projections, the perspective and the weak perspective projection:

$$\text{perspective}: \begin{cases} x_j = t_x + f\frac{\mathbf{W}_{1,j}}{\mathbf{W}_{3,j}} \\ y_j = t_y + f\frac{\mathbf{W}_{2,j}}{\mathbf{W}_{3,j}} \end{cases} \qquad \text{weak perspective}: \begin{cases} x_j = t_x + f\mathbf{W}_{1,j} \\ y_j = t_y + f\mathbf{W}_{2,j} \end{cases} \tag{16}$$

f is the focal length of the camera which is located in the origin, and (t_x, t_y) defines the image-plane position of the optical axis.

For ease of explanation, the shape transformation parameters are denoted by the vector $\boldsymbol{\rho} = [f\ \phi\ \theta\ \gamma\ t_x\ t_y\ \mathbf{t}_\mathbf{w}^\mathsf{T}]^\mathsf{T}$, and $\boldsymbol{\alpha}$ is the vector whose elements are the $\boldsymbol{\alpha}_i$. In the remaining of the paper, the projection of the vertex i to the image frame (x, y) is denoted by the vector valued function $\mathbf{p}(u_i, v_i; \boldsymbol{\alpha}, \boldsymbol{\rho})$. This function is clearly continuous in $\boldsymbol{\alpha}$, and $\boldsymbol{\rho}$. To provide continuity in the (u, v) space as well, we use a triangle list and interpolate between neighboring vertices as is common in Computer Graphics. Note that only N_{vv} vertices, a subset of the N_v vertices, are visible after the 2D projection (the remaining vertices are hidden by self-occlusion). We call this subset the domain of the shape projection $\mathbf{p}(u_i, v_i; \boldsymbol{\alpha}, \boldsymbol{\rho})$ and denote it by $\Omega(\boldsymbol{\alpha}, \boldsymbol{\rho}) \in (u, v)$.

In conclusion, the shape modeling and its projection provides a mapping from the parameter space $\boldsymbol{\alpha}, \boldsymbol{\rho}$ to the image frame (x, y) via the reference frame (u, v). However to synthesize an image, we need the inverse of this mapping, detailed in the next section.

3.2 Inverse Shape Projection

The shape projection aforementioned maps a (u, v) point from the reference space to the image frame. To synthesize an image, we need the inverse mapping: generating an image is performed by looping on the pixels (x, y). In order to know which color must be drawn on that pixel, it is needed to know where this pixel is mapped into the reference frame. This is the aim of the inverse shape mapping explained in this section.

The inverse shape projection, $\mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho})$, maps an image point (x, y) to the reference frame (u, v). Let us denote the composition of a shape projection and its inverse by the symbol $\circ$, hence, $\mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho}) \circ \mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho})$ is equal to $\mathbf{p}(\mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho}); \boldsymbol{\alpha}, \boldsymbol{\rho})$, but we prefer the former notation for clarity. The inverse shape projection is defined by the following equation which specifies that under the same set of parameters the shape projection composed with its inverse is equal to the identity:

$$\begin{aligned} \mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho}) \circ \mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho}) &= (x, y), \\ \mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho}) \circ \mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho}) &= (u, v). \end{aligned} \tag{17}$$

Due to the discretization of the shape, it is not easy to analytically express $\mathbf{p}^{-1}$ as a function of $\mathbf{p}$, but it can be computed using the triangle list: The domain of the plane (x, y) for which there exists an inverse under the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$, denoted by $\Psi(\boldsymbol{\alpha}, \boldsymbol{\rho})$, is the range of $\mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho})$. Such a point of (x, y) lies in a single visible triangle under the projection $\mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho})$. So, the point in (u, v) under the inverse projection has the same relative position in this triangle in the (u, v) space. This process is depicted in Figure 5.

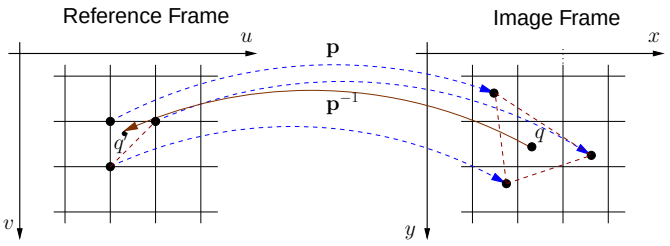


Fig. 5. The inverse shape function $\mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho})$ maps the point q (defined in the (x, y) coordinate system), onto the point q' in (u, v). This is done by recovering the triangle which would contain the pixel q under the mapping $\mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho})$. Then the relative position of q in that triangle is the same as the relative position of q' in the same triangle in the (u, v) space.

3.3 Illumination and Color Transformation

Ambient and directed light

We simulate the illumination of a face using an ambient light and a directed light. The effects of the illumination are obtained using the standard Phong model, which approximately describes the diffuse and specular reflection on a surface ([14], see [8] for further details). The parameters of this model are the intensity of the ambient light ($L_{r,amb}$,

$L_{g,amb}$, $L_{b,amb}$), the intensity of the directed light ($L_{r,dir}$, $L_{g,dir}$, $L_{b,dir}$), its direction (θ_l and ϕ_l), the specular reflectance of human skin (k_s) and the angular distribution of the specular reflections of human skin (ν).

Color transformation

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 [7]. This is a linear transformation which multiplies the RGB color of a vertex (after it has been illuminated) by the matrix $\mathbf{M}$ and adds the vector $\mathbf{o} = [o_r,\ o_g,\ o_b]^{\mathrm{T}}$, where:

$$\mathbf{M}(c, g_r, g_g, g_b) = \begin{pmatrix} g_r & 0 & 0 \\ 0 & g_g & 0 \\ 0 & 0 & g_b \end{pmatrix} \cdot \left[\mathbf{I} + (1-c) \begin{pmatrix} 0.3 & 0.59 & 0.11 \\ 0.3 & 0.59 & 0.11 \\ 0.3 & 0.59 & 0.11 \end{pmatrix} \right] \tag{18}$$

For brevity, the illumination and color transformation parameters are regrouped in the vector $\boldsymbol{\iota}$. Hence the illuminated texture depends on the coefficients of the linear combination regrouped in $\boldsymbol{\beta}$, on the light parameters, $\boldsymbol{\iota}$, and on $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ used to compute the normals and the viewing direction of the vertices required for the Phong illumination model. Similarly to the shape, we denote the color of a vertex i by the vector valued function $\mathbf{t}(u_i, v_i; \boldsymbol{\beta}, \boldsymbol{\iota}, \boldsymbol{\alpha}, \boldsymbol{\rho})$, which is extended to the continuous function $\mathbf{t}(u, v; \boldsymbol{\beta}, \boldsymbol{\iota}, \boldsymbol{\alpha}, \boldsymbol{\rho})$ by using the triangle list and interpolating.

3.4 Image Synthesis

Synthesizing the image of a face is performed by mapping a texture from the reference to the image frame using an inverse shape projection:

$$\mathbf{I}(x_j, y_j; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\iota}) = \mathbf{t}(u, v; \boldsymbol{\beta}, \boldsymbol{\iota}, \boldsymbol{\alpha}, \boldsymbol{\rho}) \circ \mathbf{p}^{-1}(x_j, y_j; \boldsymbol{\alpha}, \boldsymbol{\rho}) \tag{19}$$

where j runs over the pixels which belongs to $\Psi(\boldsymbol{\alpha}, \boldsymbol{\rho})$, i.e. the pixels for which a shape inverse exist as defined in Section 3.2.

4 Image Analysis with a 3D Morphable Model

In the analysis by synthesis framework, an algorithm seeks the parameters of the model which render a face as close to the input image as possible. These parameters explain the image and can be used for high-level task such as identification. This algorithm is called a *fitting algorithm*. It is characterized by the following four features:

- **Efficient**: The computational load allowed for the fitting algorithm is clearly dependent on applications. Security applications, for instance, requires fast algorithms (i.e. near real time).

- **Robust** against non-Gaussian noise. The assumption of normality of the difference between the image synthesized by the model and the input image is generally violated due to the presence of accessories or artifacts (glasses, hair, specular highlight).
- **Accurate**, as we have already pointed out, the accuracy is crucial for the application which is to use the fitting results (and, generally, the level of accuracy required depends thereon.)
- **Automatic**: The fitting should require as little human intervention as possible, optimally, no initialization.

An algorithm capable of any of the four aforementioned features is difficult to set up. An algorithm capable of *all* four features is the holy grail of model based computer vision. In this chapter we present two fitting algorithms. The first one, called *Stochastic Newton Optimization* (SNO) is accurate but computationally expensive: a fitting takes 4.5 min. on a 2GHz Pentium IV. SNO is detailed in [8]. The second fitting algorithm is a 3D extension of the *Inverse Compositional Image Alignment* (ICIA) algorithm introduced by [1]. It is more efficient than SNO, a fitting requires 30s on the same machine. Our ICIA algorithm was introduced in [27].

As initialization, the algorithms require the correspondences between some of the model vertices (typically 8) and the input image. These correspondences are set manually . They are required to obtain a good initial condition for the iterative algorithm. The 2D positions in the image of these N_l points are set in the matrix $\mathbf{L}_{2\times N_l}$. They are in correspondences with the vertex indices set in the vector $\mathbf{v}_{N_l\times 1}$. The positions of these landmarks for three views are shown in Figure 6.

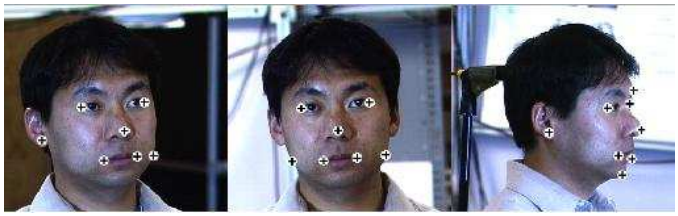


Fig. 6. Initialization: Seven landmarks for front and side views and eight for the profile view are manually labeled for each input image.

Both algorithms use optimization techniques to minimize a cost function. The main difference between them is that SNO is an *additive* algorithm whereby the update is added to the current parameters, whereas ICIA is an *inverse compositional* algorithm (see Section 4.3). The advantage of ICIA is that it uses derivatives computed in one point of the parameter space. Hence the derivatives must not be re-computed at each iteration as is the case with SNO. This is the reason of the efficiency of ICIA.

4.1 Maximum A Posteriori estimation of the parameters

Both algorithms aim to find the model parameters $\boldsymbol{\alpha}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\iota}$ that explain an input image. To increase the robustness of the algorithms, these parameters are estimated by a *maximum*

a posteriori (MAP) estimator, which maximizes $p(\boldsymbol{\alpha}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\iota} \,|\, \mathbf{I}_{\text{input}}, \mathbf{L})$ (see [7]). Applying the Bayes rule and neglecting the dependence between parameters yield:

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\iota} \,|\, \mathbf{I}_{\text{input}}, \mathbf{L}) \sim p(\mathbf{I}_{\text{input}} \,|\, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\iota}) \cdot p(\mathbf{L} \,|\, \boldsymbol{\alpha}, \boldsymbol{\rho}) \cdot p(\boldsymbol{\alpha}) \cdot p(\boldsymbol{\beta}) \cdot p(\boldsymbol{\rho}) \cdot p(\boldsymbol{\iota}) \quad (20)$$

The expression of the priors $p(\boldsymbol{\alpha})$ and $p(\boldsymbol{\beta})$, is given by Equation (10). For each shape projection and illumination parameters, we assume a Gaussian probability distribution with mean $\overline{\rho}_i$ and $\overline{\iota}_i$ and with variances $\sigma^2_{\boldsymbol{\rho},i}$ and $\sigma^2_{\boldsymbol{\iota},i}$. These values are set manually.

Assuming that the x and y coordinates of the landmark points are independent and that they have the same Gaussian distribution, with variance σ_L^2, gives:

$$E_L = -2 \log p(\mathbf{L} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}) = \frac{1}{\sigma_L^2} \sum_j^{N_l} \| \mathbf{L}_{\cdot,j} - \begin{pmatrix} x_{\mathbf{v}_j} \\ y_{\mathbf{v}_j} \end{pmatrix} \|^2 \quad (21)$$

The difference between our two algorithms lies in the formulation of the likelihood $p(\mathbf{I}_{\text{input}} \mid \boldsymbol{\alpha}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\iota})$ which results in two different maximization schemes.

4.2 Stochastic Newton Optimization

The likelihood of the input image given the model parameters is expressed in the image frame. Assuming that all the pixels are independent and that they have the same Gaussian distribution with variance σ_I^2, gives:

$$E_I = -2 \log p(\mathbf{I}_{\text{input}} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\iota}) = \frac{1}{\sigma_I^2} \sum_{x,y} \| \mathbf{I}_{\text{input}}(x,y) - \mathbf{I}(x,y; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\iota}) \|^2 . \quad (22)$$

The sum is carried out over the pixels that are projected from the vertices in $\Omega(\boldsymbol{\alpha}, \boldsymbol{\rho})$. At one pixel location, the norm is computed over the three color channels. The overall energy to be minimized is then:

$$E = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_L^2} E_L + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\boldsymbol{\rho}_i - \overline{\rho}_i)^2}{\sigma_{\boldsymbol{\rho},i}^2} + \sum_i \frac{(\boldsymbol{\iota}_i - \overline{\iota}_i)^2}{\sigma_{\boldsymbol{\iota},i}^2}. \quad (23)$$

This log-likelihood is iteratively minimized by performing a Taylor expansion up to the second order (i.e. approximating the log-likelihood by a quadratic function) and computing the update that minimizes the quadratic approximation. The update is added to the current parameter to obtain the new parameters.

We use a stochastic minimization to decrease the odds of getting trapped in a local minima and to decrease the computational time: Instead of computing E_I and its derivatives on all pixels of $\Psi(\boldsymbol{\alpha}, \boldsymbol{\rho})$, it is computed only on a subset of 40 pixels thereof. These pixels are randomly chosen at each iteration. The first derivatives are computed analytically using the chain rule. The Hessian is approximated by a diagonal matrix computed by numeric differentiation every 1000 iterations. This algorithm is further detailed in [8].

SNO is very accurate (see the experiments in Section 5). However, its main drawback is its poor efficiency due to the fact that the derivatives are computed at each iteration. In the next section we present a fitting algorithm that uses constant derivatives. It is hence faster.

Fitting Results

In Figure 7, several fitting results and reconstructions are shown. These were obtained with the SNO algorithm on some of the PIE images (see Section 13.1.4 of this book). These images are illuminated with ambient light and one directed light source. The algorithm was initialized with 7 to 8 landmark points (depending on the pose of the input image, see Figure 6). In the third column, the separation between the albedo of the face and the illumination is not optimal: part of the specular reflections were attributed to the texture by the algorithm. This may be due to shortcomings of the Phong illumination model for reflections at grazing angles or to a prior probability inappropriate for this illumination condition (The prior probabilities of the illumination and rigid parameters, $\sigma^2_{\boldsymbol{\rho},i}$ and $\sigma^2_{\boldsymbol{\iota},i}$, are kept constant for the fitting of the 4488 PIE images).

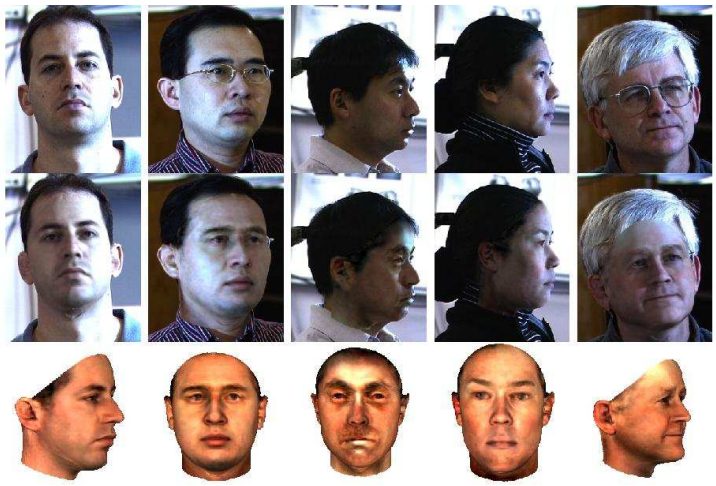


Fig. 7. SNO Fitting Results: Three-dimensional reconstruction from CMU-PIE images using the SNO fitting algorithm. Top: originals, middle: reconstructions rendered into original, bottom: novel views. The pictures shown here are difficult due to harsh illumination, profile views, or eye glasses. Illumination in the third image is not fully recovered, so part of the reflections are attributed to texture.

4.3 Inverse Compositional Image Alignment

In this Section, we present a second fitting algorithm whose major feature compared to the previous Stochastic Newton Optimization algorithm is its improved efficiency. The efficiency comes from the fact that throughout the algorithm, the derivatives needed are always computed at the same point in the parameter space. Hence the derivatives do not change across iterations and can be pre-computed. We introduced this algorithm in [27]. The original Inverse Compositional Image Alignment (ICIA) algorithm was introduced by [1] as an efficient algorithm for fitting 2D correspondence-based models. They derived

a first order approximation thereof to fit Flexible Appearance Models (i.e. sparse correspondence based 2D models). In this section, we extend this algorithm to 3D Morphable Models.

ICIA log-likelihood

As opposed to the SNO algorithm, the log-likelihood of the input image is computed in the reference frame (u, v). Another difference is that the parameters' update is inverted and composed with the current estimate instead of simply added to the current estimate.

We derived this algorithm for the case where the face is illuminated by ambient light only, i.e. without directed light. Then the texture depends on $\boldsymbol{\beta}$ and on $\boldsymbol{\iota}$. In order to formulate the log-likelihood, we introduce the inverse texture mapping, using the color transformation matrices $\mathbf{M}$ and $\mathbf{o}$ defined in Equation (18):

$$\mathbf{t}^{-1}(\mathbf{t}(u_i, v_i); \boldsymbol{\beta}, \boldsymbol{\iota}) = \mathbf{t}(u_i, v_i) - \mathbf{M}^{-1} \cdot \sum_{k=1}^{N_t} \beta_k \cdot \mathbf{T}^k_{\cdot,i} - \mathbf{o} \cdot \mathbf{1}_{1 \times N_v}, \tag{24}$$

This definition is chosen for the texture inverse because then a texture composed with its inverse, under the same set of parameters is equal to the mean texture: $\mathbf{t}^{-1}(\mathbf{t}(u_i, v_i; \boldsymbol{\beta}, \boldsymbol{\iota}); \boldsymbol{\beta}, \boldsymbol{\iota}) = \mathbf{T}^0_{\cdot,i}$. Denoting by $\boldsymbol{\gamma} = [\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\rho}^{\mathrm{T}}]^{\mathrm{T}}$ the shape and projection parameters, the log-likelihood is expressed as:

$$E_I = \frac{1}{2\sigma_I^2} \cdot \sum_{u_i, v_i \in \Omega(\boldsymbol{\gamma}^{\boldsymbol{d}})} \big(\mathbf{t}(u, v; \Delta\boldsymbol{\beta}, \Delta\boldsymbol{\iota}) \circ \mathbf{p}^{-1}(x, y; \boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}(u_i, v_i; \boldsymbol{\gamma}^{\boldsymbol{d}} + \Delta\boldsymbol{\gamma}) - \mathbf{t}^{-1}(\mathbf{I}_{\text{input}}(x, y) \circ \mathbf{p}(u_i, v_i; \boldsymbol{\gamma}^{\boldsymbol{c}}); \boldsymbol{\beta}^{\boldsymbol{c}}, \boldsymbol{\iota}^{\boldsymbol{c}})\big)^2 \tag{25}$$

where the parameters superscripted by $^{\boldsymbol{d}}$ refer to the parameters at which the derivatives are computed and the parameters superscripted by $^{\boldsymbol{c}}$ refer to the current parameters. The log-likelihood is to be minimized with respect to the model parameters update $\Delta\boldsymbol{\gamma}$, $\Delta\boldsymbol{\beta}$ and $\Delta\boldsymbol{\iota}$. The current estimate of the model parameters are $\boldsymbol{\gamma}^{\boldsymbol{c}}$, $\boldsymbol{\beta}^{\boldsymbol{c}}$ and $\boldsymbol{\iota}^{\boldsymbol{c}}$. An simple example of this log-likelihood is given in Figure 8. The first row shows the mean texture in the reference space which is to be put in correspondence with an input image (displayed on its right). The input image is mapped to the reference frame using the current shape parameters (second row): For a point (u_i, v_i) in $\Omega(\boldsymbol{\gamma}^{\boldsymbol{d}})$, the image is sampled at the pixel $\mathbf{p}(u_i, v_i; \boldsymbol{\gamma}^{\boldsymbol{c}})$. Then, the update which minimizes E_I is a transformation of the reference space given by $\mathbf{p}^{-1}(x, y; \boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}(u_i, v_i; \boldsymbol{\gamma}^{\boldsymbol{d}} + \Delta\boldsymbol{\gamma})$ (third row).

There are two novelties with respect to the original ICIA [1] formulation: The first is the presence of the inverse shape projection $\mathbf{p}^{-1}(x, y; \boldsymbol{\gamma}^{\boldsymbol{d}})$ between the shape update and the texture update. This inverse shape projection must be present because it is not possible to compose a shape (projecting to the image frame) with a texture (whose domain is the reference frame). The second novelty is the update of the texture and illumination parameters as well, $\Delta\boldsymbol{\beta}$ and $\Delta\boldsymbol{\iota}$.

Let us now compute the derivative of the ICIA log-likelihood (Equation (25)) with respect to the shape and projection parameter update $\Delta\boldsymbol{\gamma}$, at the point $\Delta\boldsymbol{\gamma} = 0, \Delta\boldsymbol{\beta} =$

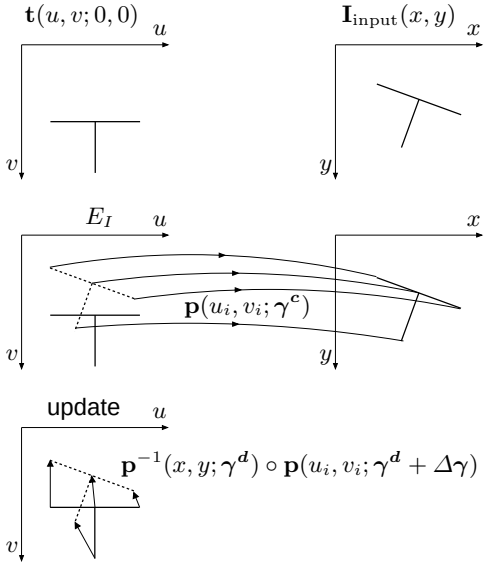


Fig. 8. Example of the computation of the log-likelihood for a reference texture and an input image given at the first row. On the second row, the image is sampled at the position given by the current shape projection estimate. On the third row, the parameters $\Delta\gamma$ that minimizes E_I are computed. This update is a transformation of the reference frame.

$0, \Delta\boldsymbol{\iota} = 0$. In the remaining, we will omit the dependent variables, assuming that it is clear that the dependents of $\mathbf{p}$ and $\mathbf{t}$ are (u, v) and the dependents of $\mathbf{p}^{-1}$ and I are (x, y).

$$\frac{\partial E_I}{\partial \Delta\boldsymbol{\gamma}_k} = \sum_i \frac{\partial (t(0,0) \circ \mathbf{p}^{-1}(\boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}_i(\boldsymbol{\gamma}^{\boldsymbol{d}} + \Delta\boldsymbol{\gamma}))}{\partial \Delta\boldsymbol{\gamma}_k}\Bigg|_{\Delta\boldsymbol{\gamma}=0}^{\mathsf{T}} \cdot [\mathbf{t}(0,0) \circ \mathbf{p}^{-1}(\boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}_i(\boldsymbol{\gamma}^{\boldsymbol{d}}) - \mathbf{t}^{-1}(I \circ \mathbf{p}_i(\boldsymbol{\gamma}^{\boldsymbol{c}}); \boldsymbol{\beta}^{\boldsymbol{c}}, \boldsymbol{\iota}^{\boldsymbol{c}})] \quad (26)$$

Note that, using Equation (19) and the chain rule:

$$\mathbf{t}(0,0) \circ \mathbf{p}^{-1}(\boldsymbol{\gamma}^{\boldsymbol{d}}) = \mathbf{I}^{\mathbf{d}}(x, y; \boldsymbol{\alpha}^{\boldsymbol{d}}, 0, \boldsymbol{\rho}^{\boldsymbol{d}}, 0) \quad (27)$$

$$\frac{\partial (t(0,0) \circ \mathbf{p}^{-1}(\boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}_i(\boldsymbol{\gamma}))}{\partial \boldsymbol{\gamma}_k} = \nabla \mathbf{I}^{\mathbf{d}} \cdot \frac{\partial \mathbf{p}_i(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}_k} \quad (28)$$

We refer to the second factor of the right member of Equation (26) in squared brackets as the *texture error* at the vertex i, $\mathbf{e}_i$. The texture error, $\mathbf{e}$, is a column vector of length $3N_{vv}$; it is a difference of two terms: The first one is the mean texture (the projection and inverse projection cancel each other using Equation 17). The second term is the image to be fitted mapped to the reference frame (u, v) using the current shape and projection parameters and inverse-texture mapped with the current texture and illumination parameters. At the optimum (and if the face image can be fully explained by the model), this term is equal to the mean texture, and hence the texture error is null. The first factor of Equation (26) is the

element of the shape Jacobian, $\mathbf{J^s}$, at the row i and column k. The dimensions of the shape Jacobian matrix are $3N_{vv} \times N_s$. As the parameters' update is composed with the current estimate, the Taylor expansion of Equation (25) is always performed in $\Delta\boldsymbol{\gamma} = 0, \Delta\boldsymbol{\beta} = 0, \Delta\boldsymbol{\iota} = 0$. Hence, the Jacobian depends only on $\boldsymbol{\gamma^d}$ which is constant across iterations. This is the key feature of the ICIA algorithm: the Jacobian and the Gauss approximation of the Hessian can be pre-computed, as opposed to those of the SNO algorithm that depends on the current parameters. As a result, what is to be computed at each iteration is a matrix-vector product and a shape projection composition (explained in the next Section).

The derivatives with respect to the texture parameters update $\Delta\boldsymbol{\beta}$ and illumination parameters $\Delta\boldsymbol{\iota}$ take a similar form to those of the shape parameters. The combined Jacobian of the shape and texture model is then: $\mathbf{J} = [\mathbf{J^s}\mathbf{J^t}]$. The Gauss approximation of the Hessian is $\mathbf{H} = \mathbf{J^T}\mathbf{J}$, leading to the Gauss-Newton update:

$$\begin{pmatrix} \Delta\boldsymbol{\gamma} \\ \Delta\boldsymbol{\beta} \\ \Delta\boldsymbol{\iota} \end{pmatrix} = -\mathbf{H}^{-1} \cdot \mathbf{J^T} \cdot \mathbf{e} \tag{29}$$

Parameters at which the derivatives are computed

The derivatives of Equations (28) are computed using an image of the mean texture $\mathbf{I^d}$. This image is rendered in a particular image frame set by ϕ^d and θ^d. The gradient of this image is multiplied with the derivative of the 3D shape model projected in the same image frame. As the shape model is in 3D, in different image frames, the vertices move in different directions with $\boldsymbol{\alpha}$. This means that the Jacobian depends on the image frame at which it is computed. For the second order Taylor expansion (performed to obtain Equation (29)) to be valid, this image frame must be close to the image frame of the optimum, i.e. ϕ^d and θ^d must be close to the optimum ϕ and θ. We conducted synthetic experiments to understand the impact of poor ϕ^d and θ^d on the fitting result. Figure 9 shows a plot of the error in correspondence obtained after fitting synthetic face images with various ϕ^d different from the optimal ϕ. The synthetic face image is generated with one Morphable Model and fitted with another Morphable Model. As the optimum face image is synthetic, we know exactly its pose and the 2D position of its vertices. The figure shows that if a set of derivatives is pre-computed at 20° interval, the error in correspondences is less than a pixel. As a result, a set of Jacobians is computed for a series of different ϕ^d, θ^d. During the iterative fitting, the derivatives used are the ones closest to the current estimation of ϕ, θ. Note that, at first, this approach might seem very close to the View-based approach [23, 22, 10]. The difference is, however, fundamental. In this approach, the extraneous (rotation) parameters are clearly separated from the intrinsic (identity, i.e. $\boldsymbol{\alpha}, \boldsymbol{\beta}$) parameters. They are, however, convolved with one another in the View-based approach.

Shape projection composition

In the ICIA algorithm the shape projection update is *composed* with the current shape projection estimate. This composition, detailed in this section, is performed in two steps. The

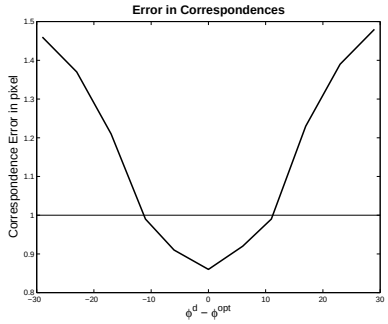


Fig. 9. Error in correspondence using derivatives computed at different azimuths than the optimum.

first step computes the correspondences after composition between the model vertices and the image pixels, i.e. it yields a mapping from (u_i, v_i) to (x, y), denoted by $\mathbf{p}^*(u_i, v_i)$. The second step maps this set of vertices-pixels correspondence to the shape model, yielding the model parameters after composition, $\boldsymbol{\alpha}^*$ and $\boldsymbol{\rho}^*$.

Correspondences after composition

The update obtained after an iteration is a transformation of the reference frame (u, v): $\mathbf{p}^{-1}(x, y; \boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}(u_i, v_i; \boldsymbol{\gamma}^{\boldsymbol{d}} + \Delta\boldsymbol{\gamma})$ (see Equation (25)). It is this transformation that must be composed with the current shape projection to obtain the new correspondences: The result of the shape projection composition is a shape projection, $\mathbf{p}^*(u_i, v_i)$, mapping the points of the reference frame (u_i, v_i) to the image frame, (x_i^*, y_i^*), equal to:

$$\mathbf{p}^*(u_i, v_i) = \mathbf{p}(u, v; \boldsymbol{\gamma}^{\boldsymbol{c}}) \circ \mathbf{p}^{-1}(x, y; \boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}(u_i, v_i; \boldsymbol{\gamma}^{\boldsymbol{d}} + \Delta\boldsymbol{\gamma}) \tag{30}$$

Taking the example of Figure 8, the composition of its update with the current parameter is shown on Figure 10. Under the first shape projection, the rightmost on the above equation, the vertex i is mapped to the image frame, say the point (x_i^+, y_i^+), under the parameters $\boldsymbol{\gamma}^{\boldsymbol{d}} + \Delta\boldsymbol{\gamma}$ using Equations (9), (15) and (16). Then, under the inverse shape projection, this point (x_i^+, y_i^+) is mapped to the reference frame, say the point (u_i^+, v_i^+), using the procedure described in Section 3.2. Finally, (u_i^+, v_i^+) is mapped to (x_i^*, y_i^*) using the shape projection with the parameters $\boldsymbol{\gamma}^{\boldsymbol{c}}$.

Correspondences mapping to the shape model

The first step of the composition yields a set of correspondences between the model vertices (u_i, v_i) and the points in the image frame (x_i^*, y_i^*). To recover the model parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$ explaining these correspondences, Equations (9), (15) and (16) must be inverted. In the case of perspective projection, the equation governing the projection is non-linear and hence the parameters are recovered by minimizing a non-linear function. Alternatively, if a weak-perspective projection is used, the equation is bilinear. A closed form solution to this problem is presented in [26]. This algorithm is a novel *selective* approach addressing

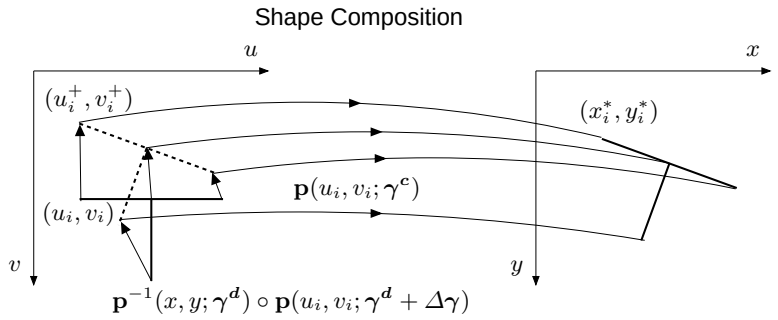


Fig. 10. Composition of the shape projection update with the current shape projection.

this problem more accurately than the former method based on SVD factorization [2]. The rotation matrix and the focal length are recovered using Equation (15) of [26], the 2D translation using Equation (16) of [26], and then the shape parameters $\boldsymbol{\alpha}$ are recovered by inverting the linear system of equations using the estimated rotation matrix and focal length.

It should be noted that neither with the weak perspective nor with the perspective projections, can these equation be fully inverted: there will remain a small residual, called the *correspondences mapping error*. This is due to the fact that the shape model is not closed under composition.

Deferred Correspondence Mapping

It is to be noted that the correspondences are not required to be projected to the shape model, thereby extracting the shape projection parameters explaining a shape composition, at each iteration. As seen on the second row of Figure 8, the inputs of an iteration are the current correspondences (needed to sample the image at the correct location), not the current shape parameters, $\boldsymbol{\gamma}^c$. Then, at the end of the fitting, the correspondences are mapped to retrieve the final shape parameters. This scheme, called Deferred Correspondence Mapping, has two advantages: First, there is no correspondence mapping error introduced at each iteration. Second, the algorithm is more efficient, as the Equations (9), (15) and (16) are not inverted at each iteration.

The ICIA algorithm applied to 3D Morphable Models is detailed in [27], which also presents a robust version, using a Talwar function instead of the sum of square, thereby alleviating the Gaussian assumption of the image formation and allowing the fitting to be independent of attributes that are not modeled such as glasses.

Comparison with the Active Appearance Model Fitting

The Active Appearance Model Search algorithm is presented in the Section 4 of Chapter 3. The Active Appearance Model is a 2D correspondence based model. Its fitting algorithm has a similar feature to the ICIA algorithm presented here: Similarly to Equation (29), it uses a constant and linear relationship between the texture error and the model update. However, as opposed to the ICIA algorithm, the update is not composed with the current

parameters, but added to them. [20] shows why this additive update is inappropriate and therefore has a limited applicability.

Fitting Results

Some fitting results and reconstructions obtained with the ICIA algorithm are presented on Figure 11. The input images are part of the PIE face database (ambient light at 13 different poses). Similarly to the SNO algorithm, the initialization is provided by landmark points.

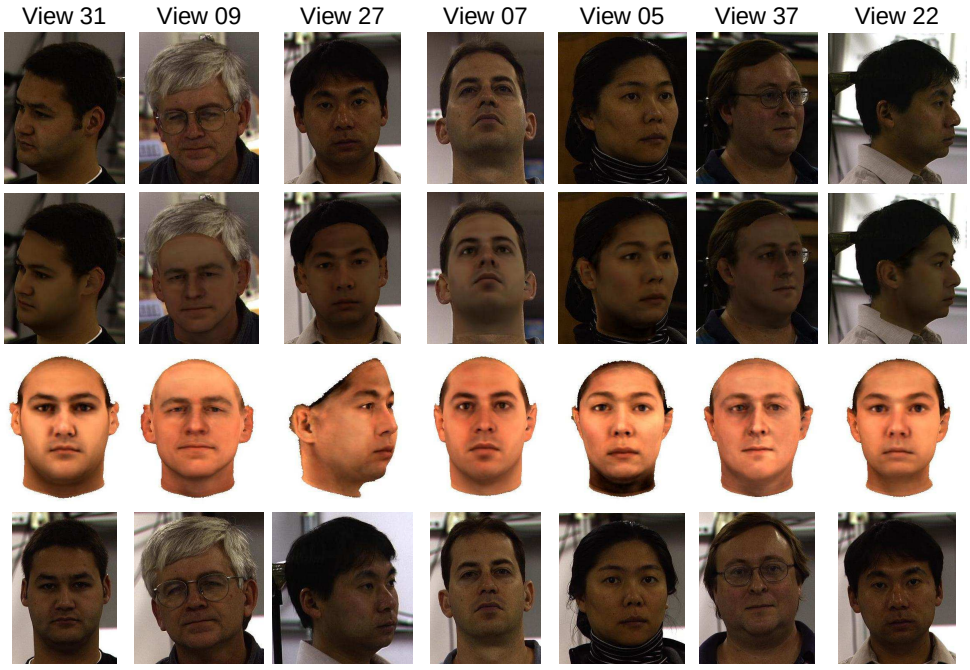


Fig. 11. ICIA Fitting Results: Three-dimensional reconstruction from CMU-PIE images using the ICIA fitting algorithm. Top: originals, second row: reconstructions rendered into originals, third row: novel views, bottom: images at approximately the same pose as the third row provided for comparison.

5 Identification and Verification

We evaluate the 3D Morphable Model and the fitting algorithms on two applications: identification and verification. In the identification task, an image of an unknown person is provided to our system. The unknown face image is then compared to a database of known people, called the gallery set. The ensemble of unknown images is called the probe set.

In the identification task, it is assumed that the individual in the unknown image is in the gallery. In a verification task, the individual in the unknown image claims an identity. The system must then accept or reject the claimed identity. Verification performance is characterized by two statistics: The verification rate is the rate at which legitimate users are granted access. The false alarm rate is the rate at which impostors are granted access. See section 2 of the Chapter 14 for more detailed explanations about these two tasks.

We evaluate our approach on three datasets. **Set 1**: a portion of the FERET dataset containing images with different poses. In the FERET nomenclature these images correspond to the series ba through bk. We omitted the images bj as the subjects present an expression that is not accounted for by our 3D Morphable Model. This dataset includes 194 individual across 9 poses at constant lighting condition except for the series bk: frontal view at another illumination condition than the rest of the images. **Set 2**: a portion of the CMU–PIE dataset including images of 68 individuals at a neutral expression viewed from 13 different angles at ambient light. **Set 3**: another portion of the CMU–PIE dataset containing images of the same 68 individuals at 3 poses (frontal, side and profile) and illuminated by 21 different directions and by ambient light only. Among the 68 individuals in Set 2 and 3, 28 wear glasses, which are not modeled and could decrease the accuracy of the fitting. None of the individuals present in these 3 sets was used to construct the 3D Morphable Model. These sets cover a large ethnic variety, not present in the set of 3D scans used to build the model. Refer to Chapter 13 for a formal description of the FERET and PIE set of images.

Identification and verification are performed by fitting an input face image to the 3D Morphable Model, thereby extracting its identity parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Then, recognition tasks are achieved by comparing the identity parameters of the input image with those of the gallery images. We define the identity parameters of a face image, denoted by the vector $\mathbf{c}$, by stacking the shape and texture parameters of the global and segmented models (see Section 2.5) and rescaling them by their standard deviations:

$$\mathbf{c} = \left[\frac{\alpha_1^g}{\sigma_{S,1}}, \ldots, \frac{\alpha_{99}^g}{\sigma_{S,99}}, \frac{\beta_1^g}{\sigma_{T,1}}, \ldots, \frac{\beta_{99}^g}{\sigma_{T,99}}, \frac{\alpha_1^{s_1}}{\sigma_{S,1}}, \ldots, \frac{\alpha_{99}^{s_1}}{\sigma_{S,99}}, \ldots, \ldots, \frac{\beta_{99}^{s_4}}{\sigma_{T,99}} \right]^{\mathsf{T}} \tag{31}$$

We define two distance measures to compare two identity parameters $\mathbf{c}_1$ and $\mathbf{c}_2$. The first measure, d_A, is based on the angle between the two vectors (it can also be seen as a normalized correlation). This measure is insensitive to the norm of both vectors. This is favorable for recognition tasks as increasing the norm of $\mathbf{c}$ produces a caricature (see Section 2.4) which does not modify the perceived identity. The second distance [8], d_W, is based on Discriminant Analysis [13] and favors directions where identity variations occur. Denoting by $\mathbf{C}_W$ the pooled within-class covariance matrix, these two distances are defined by:

$$d_A = \frac{\mathbf{c}_1^{\mathsf{T}} \cdot \mathbf{c}_2}{\sqrt{(\mathbf{c}_1^{\mathsf{T}} \cdot \mathbf{c}_1)\,(\mathbf{c}_2^{\mathsf{T}} \cdot \mathbf{c}_2)}} \quad \text{and} \quad d_W = \frac{\mathbf{c}_1^{\mathsf{T}} \cdot \mathbf{C}_W \cdot \mathbf{c}_2}{\sqrt{(\mathbf{c}_1^{\mathsf{T}} \cdot \mathbf{C}_W \cdot \mathbf{c}_1)\,(\mathbf{c}_2^{\mathsf{T}} \cdot \mathbf{C}_W \cdot \mathbf{c}_2)}} \tag{32}$$

Results on Sets 1 and 3 use the distance d_W with, for Set 1, a within-class covariance matrix learned on Set 3, and vice-versa.

5.1 Pose Variation

In this section, we present identification and verification results for images of faces which vary in pose. Table 1 lists percentages of correct rank 1 identification obtained with the SNO fitting algorithm on Set 1 (FERET). The ten different poses were used to constitute gallery sets. The results are detailed for each probe pose. The results for the front view gallery (here in bold) were first published in [8]. The first plot of Figure 12 shows the ROC for a verification task for the front view gallery and the nine other poses in the probe set. The verification rate for a false alarm rate of 1% is 87.9%.

Probe View	*bi*	*bh*	*bg*	*bf*	*ba*	*be*	*bd*	*bc*	*bb*	*bk*	mean
ϕ	$-37.9°$	$-26.5°$	$-16.3°$	$-7.1°$	$1.1°$	$11.2°$	$18.9°$	$27.4°$	$38.9°$	$0.1°$	
Gallery View											
bi	-	98.5	94.8	87.6	85.6	87.1	87.1	84.0	77.3	76.8	86.5
bh	99.5	-	97.4	95.9	91.8	95.9	94.8	92.3	83.0	86.1	93.0
bg	97.9	99.0	-	99.0	95.4	96.9	96.9	91.2	81.4	89.2	94.1
bf	95.9	99.5	99.5	-	97.9	96.9	99.0	94.8	88.1	95.4	96.3
ba	**90.7**	**95.4**	**96.4**	**97.4**	**-**	**99.5**	**96.9**	**95.4**	**94.8**	**96.9**	**95.9**
be	91.2	95.9	96.4	97.4	100.0	-	99.5	99.0	96.4	94.3	96.7
bd	88.7	97.9	96.9	99.0	97.9	99.5	-	99.5	98.5	92.3	96.7
bc	87.1	90.7	91.2	94.3	96.4	99.0	99.5	-	99.0	87.6	93.9
bb	78.9	80.4	77.8	80.9	87.6	94.3	94.8	99.0	-	74.7	85.4
bk	83.0	88.1	92.3	95.4	96.9	94.3	93.8	88.7	79.4	-	90.2

Table 1. SNO Identification performances on Set 1 (FERET). The overall mean of the table is **92.9%**. 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. The row in bold is the front view gallery (condition *ba*).

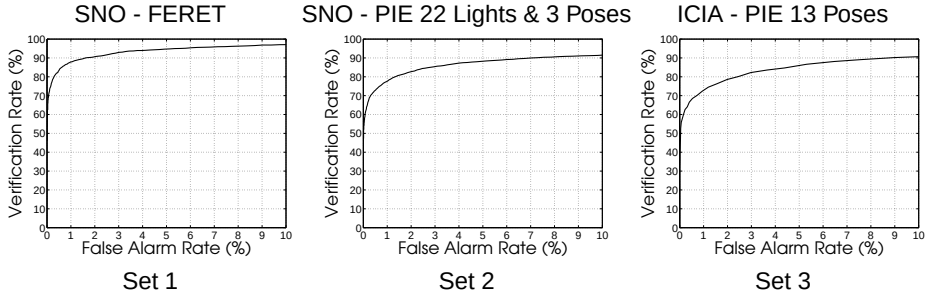


Fig. 12. ROC for a verification task obtained with the SNO and ICIA algorithms on different set of images. These plots should not be used as a comparison between SNO and ICIA as they reflect results obtained on different images.

The ICIA algorithm is evaluated on Set 2, images from 68 individuals at ambient illumination seen from 13 different poses. These images are part of the CMU–PIE face database. The rank 1 identification percentages are presented in Table 2 for each pose taken as gallery and probe sets. These results were obtained using the distance d_A. The third plot of Figure 12 is the ROC for a verification task obtained using the pose 11 as gallery set and all other poses as probe set. The verification rate for 1% false alarm rate is 72.9%. These results were obtained using a Regularized Morphable Model (see Section 2.4). They are on average 6% better than the one obtained using the original Morphable Model.

Note that the results presented here should not be used as a comparison between the two fitting algorithms as they were applied on different sets of images.

Probe View	34	31	14	11	29	9	27	7	5	37	25	2	22	mean
azimuth	-66	-47	-46	-32	-17	0	0	0	16	31	44	44	62	
altitude	3	13	2	2	2	15	2	2	2	2	2	13	3	
Gallery View														
34	-	100	100	100	94	74	75	72	71	69	66	71	75	81
31	99	-	100	100	100	99	96	85	91	91	88	87	78	93
14	99	100	-	100	100	97	97	94	93	90	88	91	76	94
11	96	100	100	-	100	100	100	99	96	90	85	84	72	93
29	88	97	100	100	-	100	100	100	99	96	88	84	71	94
9	79	97	94	97	100	-	100	100	100	94	84	93	68	92
27	76	93	99	99	100	100	-	99	100	96	85	85	79	93
7	71	91	96	99	100	100	100	-	100	97	85	88	74	92
5	88	99	97	99	100	99	100	100	-	100	100	99	90	97
37	81	91	96	97	94	99	97	96	100	-	100	100	99	96
25	81	96	97	93	96	96	87	87	97	100	-	100	100	94
2	76	82	93	90	84	90	85	87	94	100	100	-	100	90
22	85	84	87	87	76	81	85	76	90	99	100	100	-	88

Table 2. ICIA identification performances on the Set 2 (PIE dataset across 13 poses, illuminated by ambient light). The overall mean of the table is **91.9%**.

5.2 Pose and Illumination Variations

In this section we investigate the performance of our method in presence of combined pose and illumination variations. The SNO algorithm was applied to the images of Set 3, CMU–PIE images of 68 individuals varying with respect to 3 poses, 21 directed light and ambient light condition. Table 3 presents the rank 1 identification performance averaged over all lighting conditions for front, side and profile view galleries. Illumination 13 was selected for the galleries. The second plot of Figure 12 shows the ROC for a verification using as gallery a side view illuminated by light 13 and using all other images of the set as probes. The verification rate for a 1% false alarm rate is 77.5%. These results were first published in [8].

Gallery View	Probe View			mean
	front	side	profile	
front	99.8% (97.1–100)	97.8% (82.4–100)	79.5% (39.7–94.1)	92.3 %
side	99.5% (94.1–100)	99.9% (98.5–100)	85.7% (42.6–98.5)	95.0 %
profile	83.0% (72.1–94.1)	86.2% (61.8–95.6)	98.3% (83.8–100)	89.0 %

Table 3. Mean percentage of correct identification obtained after a SNO fitting on the Set 3, 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. The overall mean of the table is **92.1%**.

5.3 Identification Confidence

In this section, we present an automated technique for assessing the quality of the fitting in terms of a Fitting Score (FS). We show that the Fitting Score is correlated with identification performance and hence, may be used as an identification confidence measure. This method was first presented in [6]

A fitting score can be derived from the image error and from the model coefficients of each fitted segment from the average:

$$FS = f(\frac{E_I}{N_{vv}}, \boldsymbol{\alpha}_g, \boldsymbol{\beta}_g, \boldsymbol{\alpha}_{s_1}, \boldsymbol{\beta}_{s_1}, \ldots, \boldsymbol{\beta}_{s_4}) \tag{33}$$

Although FS can be derived by a Bayesian method, we learned it using a Support Vector Machine (see [31] for a general description of SVM and [6] for details about the FS learning).

Figure 13 shows the identification results for the PIE images varying in illumination across 3 poses, with respect to the FS for a gallery of side views. FS > 0 denotes good fittings and, FS < 0, poor ones. We divided the probe images into 8 bins of different FS and computed the percentage of correct rank 1 identification for each of these bins. There is a strong correlation between fitting score and identification performance, indicating that the fitting score is a good measure of identification confidence.

5.4 Virtual views as an aid to standard face recognition algorithms

The Face Recognition Vendor Test (FRVT) 2002 [24] was an independently administered assessment, conducted by the U.S. Government, of the performance of commercially available automatic face recognition systems. The test is described in Section 3 of Chapter 14. It was realized that identification of face images significantly drops if the face image is non-frontal. Hence, one of the questions addressed by FRVT02 is: Do identification performances of non-frontal face images improve if the pose is normalized by our 3D Morphable Model? To answer this question, we normalized the pose of a series of images [9]. Normalizing the pose means to fit an input image where the face is non-frontal, thereby estimating its 3D structure, and to synthesize an image with a frontal view of the estimated face. Examples of pose normalized images are shown in Figure 14. As neither the hair nor

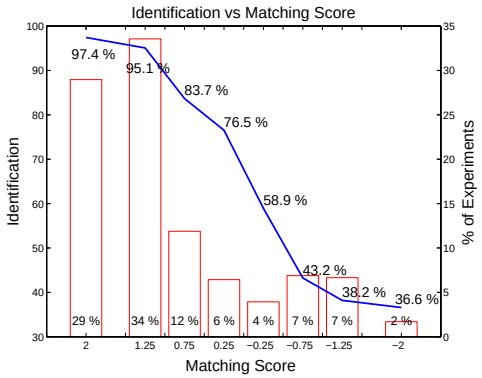


Fig. 13. Identification results as a function of the fitting score.

the shoulders are modeled, the synthetic images are rendered into a standard frontal face image of one person. This normalization is performed by the following steps:

1. Manually define up to eleven landmarks points on the input image to ensure optimal quality of the fitting.
2. Run the SNO Fitting algorithm described in Section 4.2 yielding a 3D estimation of the face in the input image.
3. Render the 3D face in front of the standard image, using the rigid parameters (position, orientation and size) and illumination parameters of the standard image. These parameters were estimated by fitting the standard face image.
4. Draw the hair of the standard face in front of the forehead of the synthetic image. This makes the transition between the standard image and the synthetic image smoother.

The normalization was applied to images of 87 individuals at five poses (frontal, two side views, one up and a down view). Identifications were performed by the ten participants to FRVT02 (see Pages 31 and 32 of [24]) using the frontal view images as gallery and nine probe sets: four probe sets with images at non-frontal views, four probe sets with the normalized images of the non-frontal views and one probe set with our pre-processing normalization applied to the front images. The comparison of performances between the normalized images (a.k.a morph images) and the raw images is presented on Figure 15 for a verification experiment (the hit-rate is plotted for a false alarm rate of 1%).

The frontal morph probe set provides a baseline for how the normalization affects an identification system. In the frontal morph probe set, the normalization is applied to the gallery images. The results on this probe set is shown on the first column of Figure 15. The verification rates would be 1.0, if a system were insensitive to the artifacts introduced by the Morphable Model and did not rely on the person's hairstyle, collar or other details which are exchanged by the normalization (which are, anyway, no reliable features to identify one person). The sensitivity to the Morphable Model of the ten participants ranges from 0.98 down to 0.45. The overall results show that with the exception of Iconquest, Morphable Models significantly improved (and usually doubled) performance.

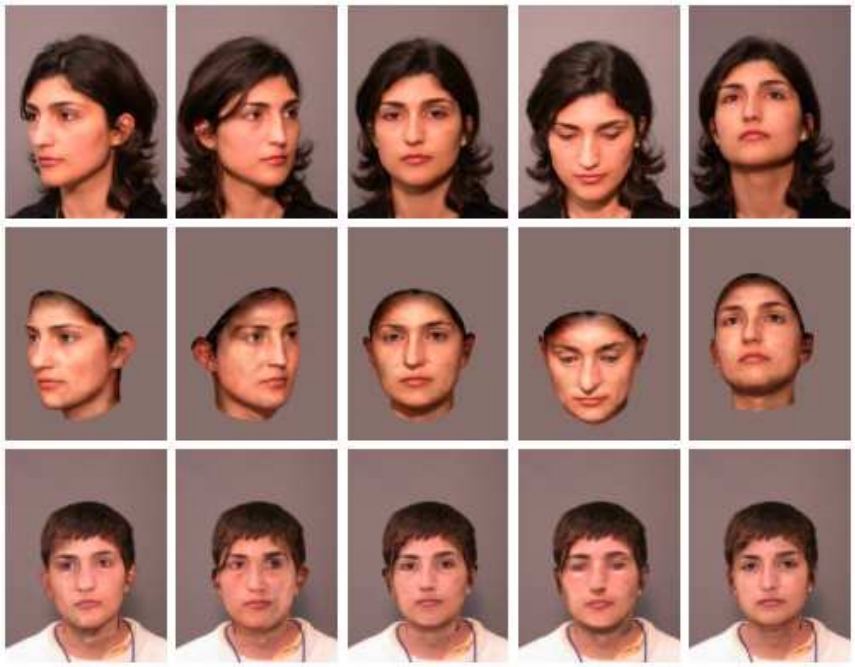


Fig. 14. From the original images (top row), we recover the 3D shape (middle row), by SNO fitting. Mapping the texture of visible face regions on the surface and rendering it into a standard background, which is a face image that we selected, produces virtual front views (bottom row). Note that the frontal-to-frontal mapping, which served as a baseline test, involves hairstyle replacement (bottom row, center).

6 Conclusion

We have shown that 3D Morphable Models can be an answer for challenging real world identification problems. They address in a natural way difficult problems such as combined variations of pose and illumination. Morphable Models can be extended, in a straightforward way, to cope with other sources of variation such as facial expression or age.

Our focus is mainly centered on improving the fitting algorithms with respect to accuracy and efficiency. We also investigated several methods for estimating identity from model coefficients. However, a more thorough understanding of the relation between these coefficients and identity might still improve recognition performance. The separation of identity from other attributes could be improved, for instance, by using other features made available by the fitting such as the texture extracted from the image (after correspondences are recovered by model fitting). Improving this separation might even be more crucial when facial expression or age variation are added to the model.

In order to model fine and identity-related details such as freckles, birthmarks and wrinkles, it might be helpful to extend our current framework for representing texture.

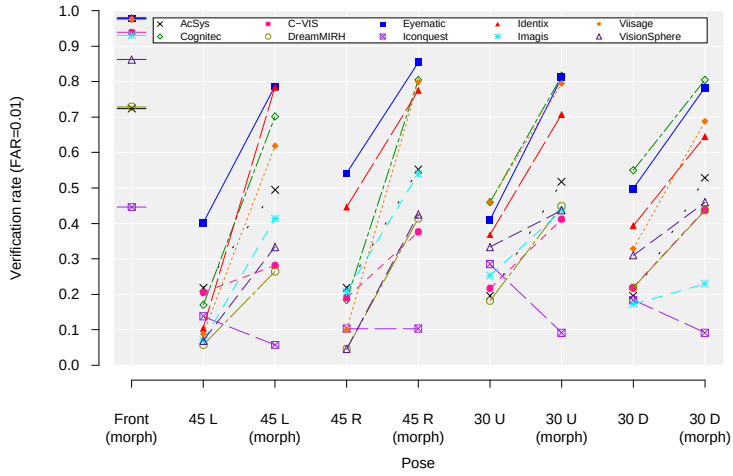


Fig. 15. The effect of the original images versus normalized images using the 3D Morphable Models. The verification rate at a false alarm rate of 1% is plotted. (Courtesy of Jonathon Phillips.)

Indeed, linear combination of textures is a rather simplifying choice, hence improving the texture model is subject to future research.

Currently our approach is clearly limited by its computational load. However this disadvantage will evaporate with time as computer increase their clock speed. Adding an automatic landmark detection will enable 3D Morphable Models to compete with state of the art commercial systems such as those that took part in the Face Recognition Vendor Test 2002 [24].

References

[1] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *CVPR*, 2001. http://www.ri.cmu.edu/projects/project_515.html.

[2] B. Bascle and A. Blake. Separability of pose and expression in facial tracking and animation. In *Sixth International Conference on Computer Vision*, 1998.

[3] J. 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. Image representations for visual learning. *Science*, 1996.

[5] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[6] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Auto. Face and Gesture Recognition*, 2002.

[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D-faces. In *SIGGRAPH 99*, 1999.
[8] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 2003.
[9] V. Blanz and T. Vetter. Generating frontal views from single, non-frontal images. in: Face recognition vendor test 2002: Technical appendix O. NISTIR 6965, Nat. Inst. of Standards and Technology, 2003.
[10] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *Automatic Face and Gesture Recognition*, 2000.
[11] I. Craw and P. Cameron. Parameterizing images for recognition and reconstruction. In *Proc. BMVC*, 1991.
[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B*, 1977.
[13] R. Duda, P. Hart, and D. Stork. *Pattern classification*. John Wiley & Sons, 2001.
[14] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, 1996.
[15] P. Hallinan. *A deformable model for the recognition of human faces under arbitrary illumination*. PhD thesis, Harvard University, 1995.
[16] R. Haralick and L. Shapiro. *Computer and robot vision*. Addison-Wesley, 1992.
[17] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.
[18] A. Lanitis, C. Taylor, and T. Cootes. An automatic face identification system using flexible appearance models. In *Proc. British Machine Vison Conference*, 1994.
[19] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Intl Joint Conf. Artificial Intelligence*, 1981.
[20] I. Matthews and S. Baker. Active appearance models revisited. Technical report, Robotics Institute, Carnegie Mellon University, 2003.
[21] T. P. Minka. Old and new matrix algebra useful for statistics. http://www.stat.cmu.edu/~minka/papers/matrix.html, 2000.
[22] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *IJCV*, 1995.
[23] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *CVPR*, 1994.
[24] P. Phillips, P. Grother, R. Michaels, D. Blackburn, E. Tabassi, and M. Bone. Face recognition vendor test 2002: Evaluation report. NISTIR 6965, Nat. Inst. of Standards and Technology, 2003.
[25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C : the art of scientific computing*. Cambridge University Press, 1992.
[26] S. Romdhani, N. Canterakis, and T. Vetter. Selective vs. global recovery of rigid and non-rigid motion. Technical report, CS Dept, University of Basel, 2003.
[27] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *IEEE International conference on Computer Vision*, 2003.
[28] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 1987.

[29] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 1999.

[30] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.

[31] V. N. Vapnik. *The Nature of Statistical Learning*. Springer-Verlag, 1995.

[32] 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*, 1997.

[33] T. Vetter and N. Troje. Separation of texture and shape in images of faces for image coding and synthesis. *Journal of the Optical Society of America*, 1997.

Index