3D Morphable Face Model,
a Unified Approach for Analysis and Synthesis of Images

Sami Romdhani Jean-Sébastien Pierrard Thomas Vetter
Computer Science Dept., University of Basel, Switzerland.
{sami.romdhani, jean.pierrard, thomas.vetter}@unibas.ch

Abstract

Morphable face models constitute a unifying framework for the analysis and synthesis of images. In the field of Computer Graphics, they are applied to synthesize photo-realistic animation of face images; in the domain of Computer Vision, they are used in face recognition applications compensating variations across pose, illumination and facial expressions. Morphable face models draw on prior knowledge of human faces in the form of a general face model, learned from examples of other faces. Whether these examples are two-dimensional images or three-dimensional face representations, the Morphable Model derived thereon share a common structure. By exploiting the correspondences between all face examples, these models introduce a vector space structure on the examples that allows to synthesize novel photo-realistic images of faces. Face analysis can be performed by fitting such a flexible model to novel images. Then, the model parameters yielding the optimal reconstruction are used to code or analyze the face depicted.

In this chapter, we start with a motivation of the analysis by synthesis approach and a discussion on the advantages of three-dimensional versus image based image models for image analysis. Then, we will explain in detail the structure of the three-dimensional Morphable Face Model we used in our experiments. The formal description of the model is followed by a detailed comparison of various model fitting algorithms published recently. At the end we present face recognition results obtained on different data bases.

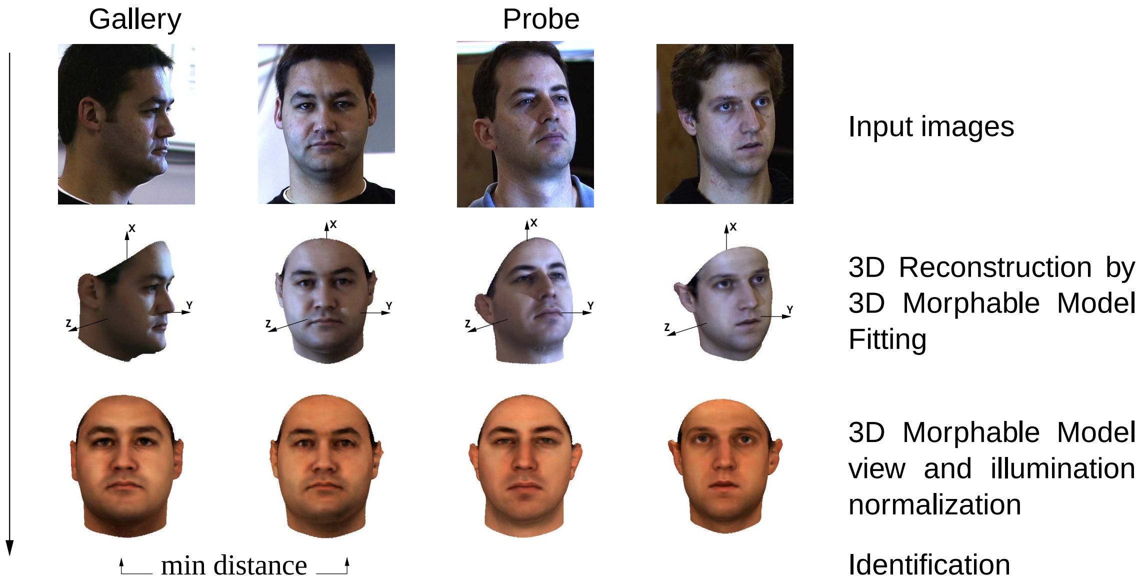
Gallery
Probe
Input images
3D Reconstruction by 3D Morphable Model Fitting
x
y
z
3D Morphable Model view and illumination normalization
min distance
Identification

Figure 1: After a pose and illumination normalization using the 3D Morphable Model, identification is performed. Identification can be performed by a comparison of either model parameters or normalized images.

0.1 Introduction

The requirement of pattern synthesis for pattern analysis has often been proposed within a Bayesian framework [22, 31] or has been formulated as an alignment technique [44]. This is in contrast to pure *bottom-up* techniques which have been advocated especially in the early stages of visual signal processing [28]. Here, a standard strategy is to reduce a signal to a lower dimensional feature vector and to compare this vector with those expected for signals in various categories. A crucial problem of these algorithms is that they cannot explicitly describe variations between or within the categories and therefore have difficulty separating unexpected noise from the variations within a particular category.

In contrast, Morphable Models, described in this paper, work by actively reconstructing the signal analyzed [5, 7, 39]. In an additional *top-down* path an estimated signal is generated and compared to the present input signal. Then by comparing the real signal with its reconstruction it is decided if the analysis is sufficient to explain the signal or not. Clearly, the crucial component in such an approach is the ability of the image model function to reconstruct the input signal.

For object classes, such as faces or cars, where all objects are similar, a model function can be learned from examples. That is, the general image model for the whole class of human faces is derived by exploiting some prototypical face examples.

Models developed for the analysis and synthesis of images of a specific class of objects must solve two problems simultaneously:

- The image model must be able to synthesize all images that cover the whole range of possible images of the class.

- It must be possible to fit the image model to a novel image. Formally, this leads to an optimization problem with all of the associated requirements that a global minimum can be found.

Solving the full problem.
The automated analysis of images of the human face has many aspects and the literature on the topic is enormous. Many of the proposed methods are driven by specific technical applications such as person identification or verification, usually under the aspect of real time performance. Especially the later constraint often requires a reduction of the generality such as the requirement of cooperative subjects, or a fixed perspective or a restriction on the number of subjects.

In contrast, the method proposed here tries to develop a unifying approach for all the different aspects of face image analysis, not trading generality or accuracy against speed. Simply, the system should have no requirement on the image to be analyzed. Before giving the details of our morphable face model approach we would like to describe three formals aspects for the comparison or the design of an automated face analysis system. These are the most relevant aspects for an understanding of the problem.

1. What are all the parameters of variation in face images, which a method can explicitly cope with.

2. What is the formal image model to represent and separate all these different parameters.

3. What is the fitting strategy for comparing a given image with the image model.

0.2 Parameters of variation in images of human faces

Human faces differ in shape and texture and, additionally, each individual face by itself can generate a variety of different images. This huge diversity in the appearance of face images makes the analysis difficult. Besides the general differences between individual faces, the appearance variations in images of a single face can be separated into the following four sources.

- Pose changes can result in dramatic changes in images showing different views of a face. Due to occlusions different parts become visible or non visible and, additionally, the parts seen in two views change their spatial configuration relative to each other.
- Lighting 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, constitute another source of variation of facial 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.
- Over a longer period of time, a face changes due to aging, to a changing hairstyle or according to makeup or accessories.

Without any requirement on the image to be analyzed, none of the parameters mentioned above can be assumed constant over a set of face images. The isolation and explicit description of all these different sources of variation must be the ultimate goal of a facial image analysis system. For example, it is desirable not to confuse the identification of a face with expression changes, or, vice versa, the recognition of the expression of a person might by eased by identifying her. This implies that an image model is required that accounts for each of these variations independently by explicit parameters. Image models not able to separate two of these parameters are not able to distinguish images varying along these parameters.

0.3 Two- or three-dimensional image models

In this section we discuss image representations that try to code explicitly all parameters of face variations as mentioned above.

The most elaborate techniques for modeling images of the human face have been developed in the field of Computer Graphics. The most general approach, in terms of modeling all parameters explicitly, consist of a 3D mesh describing the geometry of a face and bidirectional reflectance maps of the face surface that simulate the interaction of the human skin with light [33]. Geometric parameters for modeling the surface variations between individual humans as well as physical parameters for modeling the reflectance properties of human skin are derived from empirical studies on sample sets of humans. Reflectance and geometry variations are stored in an object centered coordinate system [24, 29]. For a long time, these *object-centered 3D* based techniques suffered from poor photo-realistic image quality rendering. In parallel, *image based rendering* techniques have been developed to close this gap [21, 27]. Here, light ray intensities (pixel intensities) are collected from large samples of real photographs taken from various directions around a specific object. Since all images are calibrated in 3D, according their position and viewing direction, each pixel intensity represents the light intensity along a certain ray in 3D space. For static objects, assuming a dense sampling of all possible ‘rays’, novel images from novel perspectives can be synthesized in perfect quality by reassembling the light rays for the novel perspective. Since the rays are defined in a 3D world coordinates and not in a coordinate system of the object depicted, modeling of the face shape for animation or for a different person is not obvious and has so far not been reported. A similar difficulty exists for modeling light variations, since all computations are done in world coordinates, modeling the surface light interaction can not be performed. For face animation or modeling different individuals, image based methods seem to be of little use and, in current graphics applications, the object centered three dimensional geometry and reflectance models seem to be superior.

For image analysis in the field of Computer Vision, a third type of face image models has been developed known as *linear object classes* [46, 47], as *AAMs* [26] and as an extension of deformable model [23] . Similar to the image based methods in Computer Graphics, they also start from large collections of example images of human individuals. However, instead of modeling the pixels intensities in world coordinates, a two-dimensional object centered reference system is chosen by registering all images to a single reference face image. The model parameters are then derived by statistical analysis of the variation of the pixel intensities and the variations of the correspondence fields. Since no three dimensional information is directly available in this process, on one hand, illumination and individual reflectance properties are mixed and, on the other hand, pose and individual face shape variations are not separable. This convolution of external imaging parameters with individual face characteristics limits the ability to explicitly model pose or illumination and can easily result in non realistic face images and, hence, reduce the value of these model for graphics applications. This problem was recognized for image analysis, also, and different methods for separating the external parameters have been proposed. One way is to train different models, each tuned to a specific parameter setting[14, 40, 45]. That is, for each pose a separate model is built that accounts for the head variations of different persons. Additionally, to separate illumination variations from variations of human skin color a separate image model should be developed for each illumination condition. These methods assume often that some external parameters are either constant or known beforehand. For the general case, when no prior knowledge is available, a huge number of different models would be required. To escape from this combinatorial explosion, two directions have been pursued. Interpolation between a few discrete image models can be applied to approximate the full range of images variations. However, handling several models simultaneously is still difficult with current computing technology. The other direction is to add explicit information on the three-dimensional nature of human faces. With a three-dimensional geometric object representation, changes in viewing direction and perspective can be directly computed by artificially rotating the object. Self occlusion can be easily treated by using a z-buffer approach. Surface normals also can be directly derived from such a representation. Assuming additionally some information on the light sources around the face the surface normals can be used to physically simulate the results of illumination variations. How can we obtain such a convenient three-dimension representation from a set of faces images? Implicitly, according to theory, having the positions of corresponding points in several images of one face, the three-dimensional structure of these points can be computed. However, it is proofed to be difficult to extract the tree-dimensional face surface from the images using techniques such shape from shading, structure from motion or from corresponding points visible in several images. Recently, exploiting structure from motion techniques from video streams of moving faces, three-dimensional image model could be built [8, 17, 48]. However, the geometric representations is still coarse, and hence a detailed evaluation of the surface normals for illumination modeling was not pursued.

Morphable Models, described in this paper [2, 5, 7, 39], integrate information on the three-dimensional nature of faces into the image model by taking a different approach in the model building step, without changing the basic structure of the image based models. The Morphable Models are derived from a large set of 3D textured laser scans instead of photographs or video images.

Therefore, the difficult step of model building from raw images is eased by technical means. A direct modeling of the influence of illumination must be performed, since illumination conditions in the images to be analyzed can rarely match the standard conditions for model formation. While, in image based models, a direct illumination modeling is impossible, it is quite straightforward using 3D morphable models. In 3D, surface normals can be directly computed and the interaction with light can be simulated. Here the morphable model incorporates the technique that was developed in Computer Graphics for photo-realistic rendering. Morphable Models combine the high quality rendering techniques developed in Computer Graphics with the techniques developed in Computer Vision for the modeling of object classes. Morphable face models constitute a unified framework for the analysis and synthesis of facial images. Current applications lie in the field of Computer Graphics, for photo-realistic animation of face images, and in the domain of Computer Vision, for face recognition applications compensating variations across pose, illumination and expressions.

From our current view, the question “Two- or three-dimensional image models?” has two aspects that should be considered separately. These are the process of model formation and the internal structure of the models for representing the three-dimensional nature of faces. For the second aspect, we do not see any simpler approach for handling self-occlusion, perspective or illumination variation than using a three-dimensional representation that includes surface normals. For the model formation, we think that the 3D laser scans, as used in the our morphable model, are not a must. In the near future, improved shape from shading and structure from motion techniques could transform sample sets of face images into a three-dimensional representation.

0.4 Image Analysis by Model Fitting

For the remainder of this chapter, we assume that the image to be analyzed can be instantiated by the model. Therefore, the analysis problem boils down to finding the internal model parameters that reconstruct the image. That is, we have to solve an inverse problem. Since the problem is ill-posed, the inverse of the image modeling function can not be computed analytically. It is a common strategy to apply some search techniques to find the parameters that best reconstruct the image. Instead of searching the parameter space exhaustively in a brute force approach, we would like to use some gradient based optimization methods that guides to the solution. It is clear that not all image models are equally suited. For most image models used in Graphics, it is almost impossible to tune a model to a given image, as the domain of validity of the model parameters is often disconnected and, hence, many steps require manual interactions. In Computer Vision, on the other hand, the structures of the different approaches are quite similar and all image models, based on 2D or on 3D examples, lead to similar least squares problems. First and second order derivatives of the image model can be computed and the parameter domain is often modeled as a convex domain from a multivariate normal distribution obtained by a Principal Component Analysis. The main differences are in the strategy chosen to solve the non linear optimization problem. As discussed later in detail, the different methods vary from linearizing the problem to applying Newton method exploiting second order derivatives. The computation time as well as the accuracy in terms of convergence of the different methods vary tremendously. Simpler strategies for solving the problem also tend to handle less model parameters. The morphable model approach presented in this paper does not trade efficiency against accuracy or problem simplification. We demonstrate that a high quality image model with many parameters accounting for person identity, and modeling explicitly the pose and illumination variations, can be successfully fitted to arbitrary face images.

0.5 Morphable Face Model

As mentioned in the previous sections, the 3D Morphable Model (3DMM) is based on a series of example 3D scans represented in an object centered system and registered to a single reference scan. A detailed description of the generation of a 3D Morphable Model is available in Basso *et al.* [2]. Briefly, to construct a 3DMM, a set of M example 3D laser scans are put into correspondence with a reference laser scan (in our case $M = 200$). 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 corresponds to one 3D vertex (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, $\mathbf{T}$.

$$\mathbf{S} = \begin{pmatrix} x_1 & x_2 & \cdots & x_{N_v} \\ y_1 & y_2 & \cdots & y_{N_v} \\ z_1 & z_2 & \cdots & z_{N_v} \end{pmatrix}, \qquad \mathbf{T} = \begin{pmatrix} r_1 & r_2 & \cdots & r_{N_v} \\ g_1 & g_2 & \cdots & g_{N_v} \\ b_1 & b_2 & \cdots & b_{N_v} \end{pmatrix} \tag{1}$$

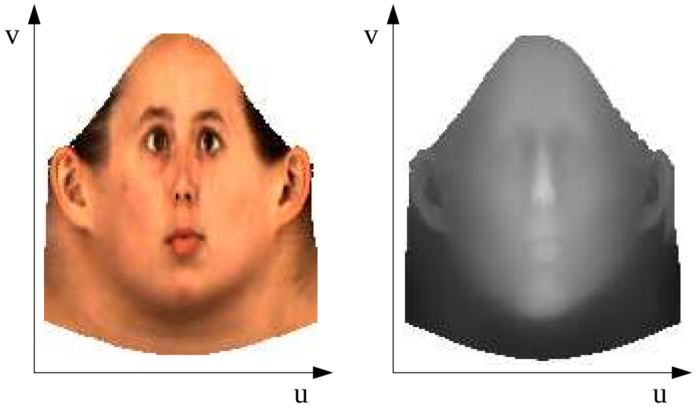

Figure 2: Texture and shape in the reference space (u, v).

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

$$\mathbf{S} = \sum_{i=1}^{M} \alpha_i \cdot \mathbf{S}_i, \qquad \mathbf{T} = \sum_{i=1}^{M} \beta_i \cdot \mathbf{T}_i \tag{2}$$

Equation (2) assumes a uniform distribution of the shapes and the textures. We know that this distribution yields a model that is not restrictive enough: For instance, 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 that transforms the space such that the covariance matrix is diagonal (i.e., it de-correlates the data). PCA is applied separately to the shape and texture spaces. 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 [36] of $\mathbf{A}$.

$$\begin{gathered} \overline{\mathbf{S}} = \tfrac{1}{M}\textstyle\sum_{i=1}^{M} \mathbf{S}_i, \quad \mathbf{a}_i = \mathrm{vec}(\mathbf{S}_i - \overline{\mathbf{S}}), \quad \mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M] = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}} \\ \mathbf{C}_{\mathbf{A}} = \tfrac{1}{M}\mathbf{A}\mathbf{A}^{\mathrm{T}} = \tfrac{1}{M}\mathbf{U}\boldsymbol{\Lambda}^2\mathbf{U}^{\mathrm{T}} \end{gathered} \tag{3}$$

The component $\mathrm{vec}(\mathbf{S})$ vectorized $\mathbf{S}$ by stacking its columns. The M columns of the orthogonal matrix $\mathbf{U}$ are the eigenvectors of the covariance matrix $\mathbf{C_A}$, 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. Now, instead of representing the data matrix $\mathbf{A}$ in its original space, it can be projected to the space spanned by the eigenvectors of its covariance matrix. Let us denote by the matrix $\mathbf{B}$ this new representation of the data, and by $\mathbf{C_B}$ its covariance matrix.

$$\mathbf{B} = \mathbf{U}^{\mathrm{T}}\mathbf{A} \qquad \mathbf{C_B} = \frac{1}{M}\mathbf{B}\mathbf{B}^{\mathrm{T}} = \frac{1}{M}\mathbf{\Lambda}^2 \tag{4}$$

The second equality of the last equation is obtained because $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix with the appropriate number of elements. Hence the projected data are de-correlated, as they have a diagonal covariance matrix. Hereafter, we denote by $\sigma_{S,i}$ and $\sigma_{T,i}$ the variances of, respectively, the shape and the texture vectors.

There is a second advantage in expressing a shape (or texture) as a linear combination of shape principal components, namely dimensionality reduction. It is demonstrated in [16], that the subspace, spanned by the columns of an orthogonal matrix $\mathbf{X}$, with N dimensions, which minimizes the mean squared difference between a data vector $\mathbf{a}$, sampled from the same population as the column vectors of the matrix $\mathbf{A}$, and its reconstruction, $\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{a}$, is the one formed by the N eigenvectors having largest eigenvalues: $\mathbf{X} = [\mathbf{U}_{\cdot,1}, \ldots, \mathbf{U}_{\cdot,N}]$, where $\mathbf{U}_{\cdot,i}$ denotes the i^{th} column of $\mathbf{U}$.

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 $\mathbf{S}^i = \mathbf{U}_{\cdot,i}^{(3)}$. The notation $\mathbf{a}_{m\times 1}^{(n)}$ [30] 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 2, we express them as a linear combination of N_S shape and N_T texture principal components.

$$\mathbf{S} = \overline{\mathbf{S}} + \sum_{i=1}^{N_S} \alpha_i \cdot \mathbf{S}^i, \qquad \mathbf{T} = \overline{\mathbf{T}} + \sum_{i=1}^{N_T} \beta_i \cdot \mathbf{T}^i \tag{5}$$

The third advantage of this formulation is that the probabilities of a shape and a texture are readily available from their parameters.

$$p(\mathbf{S}) \sim e^{-\frac{1}{2}\sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2}}, \qquad p(\mathbf{T}) \sim e^{-\frac{1}{2}\sum_i \frac{\beta_i^2}{\sigma_{T,i}^2}} \tag{6}$$

0.5.1 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 individuals present in mankind. Naturally, one way to augment the dimension of the face space would be to use 3D scans of more persons, but they were not available in our experiments. Hence, we resort to another scheme: We segment the face into four regions (nose, eyes, mouth and the rest) and use a separate set of shape and texture coefficients to code them [7]. This method multiplies by four the expressiveness of the morphable model. 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).

0.5.2 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.

Shape Projection

To render the image of a face, the 3D shape is 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 in world coordinates.

$$\mathbf{W} = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \mathbf{S} + \mathbf{t_w} \mathbf{1}_{1\times N_v} \tag{7}$$

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 perspective projection then maps a vertex i to the image plane in (x_i, y_i):

$$x_i = t_x + f\frac{\mathbf{W}_{1,i}}{\mathbf{W}_{3,i}} \qquad y_i = t_y + f\frac{\mathbf{W}_{2,i}}{\mathbf{W}_{3,i}} \tag{8}$$

where f is the focal length of the camera (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 $\boldsymbol{\rho} = [f, \phi, \theta, \gamma, t_x, t_y, \mathbf{t}_\mathbf{w}^\mathtt{T}]^\mathtt{T}$, and $\boldsymbol{\alpha}$ is the vector whose elements are the $\boldsymbol{\alpha}_i$. 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 $\boldsymbol{\alpha}$, $\boldsymbol{\rho}$. 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 $\mathit{\Omega}(\boldsymbol{\alpha}, \boldsymbol{\rho}) \in (u, v)$.

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 synthesis an image, we need the inverse of this mapping, detailed in the next section.

Inverse Shape Projection

The shape projection aforementioned maps a (u, v) point from the reference space to the image frame. To synthesis an image, we need the inverse mapping: An image is generated by looping on the pixels (x, y). To know which color must be drawn on that pixel, we must 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.

$$\begin{aligned} \mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho}) \circ \mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho}) &= (x, y) \\ \mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho}) \circ \mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho}) &= (u, v) \end{aligned} \tag{9}$$

Because the shape is discrete, it is not easy to express $\mathbf{p}^{-1}(\cdot)$ analytically as a function of $\mathbf{p}(\cdot)$, 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 $\mathit{\Psi}(\boldsymbol{\alpha}, \boldsymbol{\rho})$, 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 $\mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho})$. Therefore, 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 3.

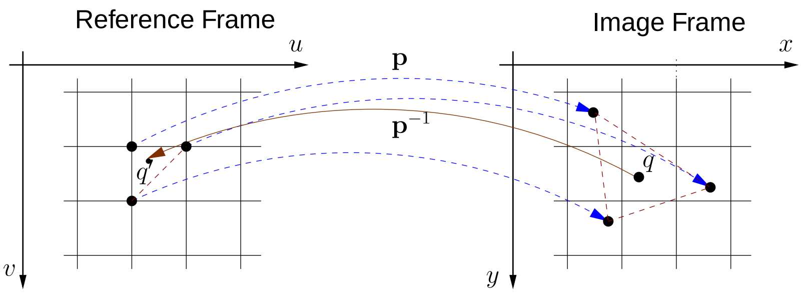

Figure 3: Inverse shape function $\mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}, \boldsymbol{\rho})$ maps the point q (defined in the (x, y) coordinate system), onto the point q' in (u, v). This is done by recovering the triangle that would contain the pixel q under the mapping $\mathbf{p}(u, v; \boldsymbol{\alpha}, \boldsymbol{\rho})$. 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.

Illumination Modeling and Color Transformation

Ambient and Directed Light We simulate the illumination of a face using an ambient light and a directed light. We use the standard Phong reflectance model that accounts for the diffuse and a specular reflection on a surface [18]. The parameters of this model are the intensity of the ambient light (L^a_r, L^a_g, L^a_b), the intensity of the directed light (L^d_r, L^d_g, L^d_b), 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). For clarity, we denote by the vector $\mathbf{t}_i$ the i^{th} column of the matrix $\mathbf{T}$, representing the RGB color of the vertex i. When illuminated, the color of this vertex is transformed to $\mathbf{t}^I_i$.

$$\mathbf{t}^I_i = \begin{pmatrix} L^a_r & 0 & 0 \\ 0 & L^a_g & 0 \\ 0 & 0 & L^a_b \end{pmatrix} \cdot \mathbf{t}_i + \begin{pmatrix} L^d_r & 0 & 0 \\ 0 & L^d_g & 0 \\ 0 & 0 & L^d_b \end{pmatrix} \cdot \left(\langle \mathbf{n}^{v,w}_i, \mathbf{d} \rangle \cdot \mathbf{t}_i + k_s \cdot \langle \mathbf{r}_i, \mathbf{v}_i \rangle^\nu \cdot \mathbf{1}_{3\times 1} \right) \tag{10}$$

The first term of this equation is the contribution of the ambient light. The first term of the last parenthesis is the diffuse component of the directed light and the second term is its specular component. To take account of the attached shadows, these two scalar products are lower bounded to zero. To take account of the cast shadows, a shadow map is computed, using standard computer graphics techniques [18]. The vertices in shadows are illuminated by the ambient light only.

In Equation 10, $\mathbf{d}$ is the unit-length light direction in Cartesian coordinates, which can be computed from its spherical coordinates by:

$$\mathbf{d} = \begin{pmatrix} \cos(\theta_l) \cdot \sin(\phi_l) \\ \sin(\theta_l) \\ \cos(\theta_l) \cdot \cos(\phi_l) \end{pmatrix} \tag{11}$$

The normal of the vertex i, $\mathbf{n}^{v,w}_i$, is expressed in world coordinates. World coordinates of a normal are obtained by rotating the normal from the object centered coordinates to the world coordinates.

$$\mathbf{n}^{v,w}_i = \mathbf{R}_\gamma \mathbf{R}_\theta \mathbf{R}_\phi \mathbf{n}^v_i \tag{12}$$

The normal of a vertex in object centered coordinates, $\mathbf{n}^v_i$, is defined as the unit-length mean of the normals of the triangles connected to this vertex (i.e. the triangles for which this vertex is one of the three corners).

$$\mathbf{n}^v_i = \frac{\sum_{j \in \mathcal{T}_i} \mathbf{n}^t_j}{\| \sum_{j \in \mathcal{T}_i} \mathbf{n}^t_j \|} \tag{13}$$

where $\mathcal{T}_i$ is the set of triangle indexes connected to the vertex i. The normal of the triangle j, denoted by $\mathbf{n}_j^t$, is determined by the unit-length cross product of the vectors formed by two of its edges. If $\mathbf{s}_{i_1}$, $\mathbf{s}_{i_2}$, and $\mathbf{s}_{i_3}$ are the Cartesian object centered-coordinates of the three corners of the triangle j (these indexes are given by the triangle list), then its normal is:

$$\mathbf{n}_j^t = \frac{(\mathbf{s}_{i_1} - \mathbf{s}_{i_2}) \times (\mathbf{s}_{i_1} - \mathbf{s}_{i_3})}{\|(\mathbf{s}_{i_1} - \mathbf{s}_{i_2}) \times (\mathbf{s}_{i_1} - \mathbf{s}_{i_3})\|} \tag{14}$$

In the Equation (10), $\mathbf{v}_i$ is the viewing direction of the vertex i, which is the unit-length direction connection the vertex i to the camera centerer. The camera centerer is at the origin of the world coordinates.

$$\mathbf{v}_i = -\frac{\mathbf{W}_{\cdot,i}}{\|\mathbf{W}_{\cdot,i}\|} \tag{15}$$

The vector $\mathbf{r}_i$ in Equation (10) is the direction of the reflection of the light coming from the direction $\mathbf{d}$, computed as follows:

$$\mathbf{r}_i = 2 \cdot \langle \mathbf{n}_i^{v,w}, \mathbf{d} \rangle \mathbf{n}_i^{v,w} - \mathbf{d} \tag{16}$$

Color Transformation Input images may vary a lot with respect to the overall tone of color. To be able to handle a variety of color images as well as gray level images and even paintings, we apply gains $g_r,\ g_g,\ g_b$, offsets $o_r,\ o_g,\ o_b$, and a color contrast c to each channel [7]. This is a linear transformation that yields the definitive color of a vertex, denoted by $\mathbf{t}_i^C$. It is obtained by multiplying the RGB color of a vertex after it has been illuminated, $\mathbf{t}_i^I$, by the matrix $\mathbf{M}$ and adding the vector $\mathbf{o} = [o_r,\ o_g,\ o_b]^{\mathtt{T}}$.

$$\mathbf{t}_i^C = \mathbf{M} \cdot \mathbf{t}_i^I + \mathbf{o}, \text{ where} \tag{17}$$

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

For brevity, the illumination and color transformation parameters are regrouped in the vector $\boldsymbol{\iota}$. Hence the illuminated and color corrected texture depends on the coefficients of the texture 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.

Similar to the shape, the color of a vertex i, $\mathbf{t}_i^C$, is represented on the (u, v) reference frame by the vector valued function $\mathbf{t}^C(u_i, v_i; \boldsymbol{\beta}, \boldsymbol{\iota}, \boldsymbol{\alpha}, \boldsymbol{\rho})$, which is extended to the continuous function $\mathbf{t}^C(u, v; \boldsymbol{\beta}, \boldsymbol{\iota}, \boldsymbol{\alpha}, \boldsymbol{\rho})$ by using the triangle list and interpolating.

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.

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

where j runs over the pixels that belong to $\Psi(\boldsymbol{\alpha}, \boldsymbol{\rho})$ (i.e., the pixels for which a shape inverse exist, as defined in Section 0.5.2).

0.6 Comparison of Fitting Algorithms

The previous section detailed the 3D Morphable Model, a mathematical formulation of the full image formation process. It takes into account most of the sources of facial image variations (flexible shape deformation, varying albedo, 3D rotation, and directed lights). A face recognition algorithm is intrinsically a method that is inverting this image formation process, i.e. inverting Equation (19), thereby separating the identity from the imaging parameters. The first algorithm that inverted the full image formation process, and that made the least assumptions, treating the problem in its full complexity, is the Stochastic Newton Optimization (SNO) [4, 6, 7]. It casted the task as an optimization problem estimating all the model parameters (shape, texture, rigid transformation and illumination). The only assumption made is that the pixels are independent and identically distributed with a residual normally distributed with a variance equal to σ_I^2. This is performed by maximizing the posterior of the parameters given the image, thereby minimizing the following energy function:

$$E = \min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho},\boldsymbol{\iota}} \frac{1}{\sigma_I^2} \sum_{x,y} \| I^m(x,y;\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho},\boldsymbol{\iota}) - I(x,y) \|^2 + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} \qquad (20)$$

This is a difficult and computationally expensive optimization problem, as the energy function is non convex. To avoid the local minima problem, a stochastic optimization algorithm is used: At each iteration of the fitting algorithm, the energy function and its derivatives are evaluated on a very small set of points (40 points) that are randomly chosen. This stochasticity introduces a perturbation on the derivatives that minimizes the risk of locking on a local minimum. The price to pay is a computationally expensive algorithm with several thousands iterations.

Over the past years, many fitting algorithms have been presented. Having computational efficiency and tractability as main goal, these methods restrain the domain of applicability and make several assumptions. It is interesting to analyze the assumptions made by each of these methods and their limitations in light of the 3DMM.

Point Distribution Model and Active Shape Model Craw and Cameron [15] were the first to align some landmark pixels on a set of face images. Then they applied PCA on the textures put in coarse correspondence by sampling them on a reference frame defined by a few landmarks. Cootes *et al.* [9, 13] represented 2D face shapes as a sparse set of feature points that corresponded to one another across a face image ensemble. Applying PCA on this set of points yielded a Point Distribution Model (PDM).

Though it can be argued that there is no major conceptual difference between the PDM's of the early nineties and the 3D shape model of the 3DMM, as both models compute statistics on a set of points in correspondence, yet there are three major differences between the two models: (i) The use of 3D rather than 2D enabling accurate out of the image plane rotation and illumination modeling. (ii) The 3DMM is a dense model including several thousands vertices whereas the PDM uses a few tens. (iii) As implemented by Cootes *et al.*, the PDM includes landmarks on the contour between the face and the background. This contour depends on the pose of the face on the image and on the 3D shape of the individual. Thus the landmarks on it do not represent the same physical points, and hence should not be put in correspondence across individuals. The authors of the PDM argue that it is capable of handling $\pm 20^\circ$ out of the image plane rotation. However, the 2D shape variation induced by this 3D rigid transformation is encoded in the 2D shape parameters, resulting in shape parameters not independent of the face pose. As explained in the first sections of this chapter, this reduces the identification capabilities of this model.

The first algorithm used to fit a PDM to an image is called the Active Shape Model (ASM) and its first version appeared in Cootes and Taylor [11]. This version used the edge evidence of the input image to adjust the 2D translation, 2D scale, image plane rotation and shape parameters of the model. Each iteration of this fitting algorithm includes two steps: First, each model point is displaced in the direction normal to the shape contour, toward the strongest edge, and with an amplitude proportional to the edge strength at the point. This yields a position for each model point that would fit better the image. The second step is to update the 2D pose and shape parameters reflecting the new position of the model points. To increase the likeliness of the resulting shape, hard limits are put on the shape parameters. This method proved itself not robust enough to deal with complex objects, where the model points do not necessarily lie on strong edges. Therefore, the second version of the ASM [12] modeled the gray-levels along the normal of the contour at each model point. Then, during model search, a better position for a model point was given by the image point, along the normal of the contour, that minimized the distance to its local gray-level model. An exhaustive search for this minimizer was performed that evaluated all the points along the normal within a given distance of the model point. A PCA model was used to model a gray-level profile along the normal of a contour point. Then the distance minimized during fitting is the norm of the reconstruction error of a gray-level profile obtained from a point along the normal evaluated as a potential minimizer. A predicament of the local gray-level models, as they were implemented in the ASM, is the fact that one half of the pixels modeled are outside the face area, in the background area, and hence could change randomly from image to image.

The first identification experiment on facial images fitted by an ASM was made by Lanitis *et al.* [25]. After fitting an image by the ASM, the gray texture enclosed withing the face contour was sampled on the reference frame defined by the mean shape (and called shape-free texture) and modeled by a PCA model. The features used for identification were the shape and texture coefficients of the shape and texture models.

Active Appearance Model Continuing in this direction, researchers started to use not only the pixels along the normal of the landmark points, but rather the full texture in the face area to drive the parameters fitting algorithm. The main motivation was that, as the algorithm would use more information, the fitting would converge faster to a more accurate minimum more robustly. First, Gleicher [20], with the Image Difference Decomposition (IDD) algorithm, then Sclaroff and Isidoro [42], with the Active Blobs, and Cootes *et al.* [10], with the Active Appearance Model (AAM), used the full texture error to compute, linearly, an update of the model parameters. The texture error, $\delta\mathbf{t}$, is the difference between the texture extracted from the input image and sampled using the shape parameters and the model texture.

$$\delta\mathbf{t} = I(x,y) \circ \mathbf{p}(u,v;\boldsymbol{\alpha}) - \mathbf{t}(u,v;\boldsymbol{\beta}) \tag{21}$$

As this algorithm is applied to the AAM that does neither model out of the image plane rotation, nor illumination, this last equation does neither depend on $\boldsymbol{\rho}$, nor on $\boldsymbol{\iota}$. This equation defining the texture error is the same as the term of the SNO energy function, inside the norm, with the difference that the texture error is sampled in the reference frame, not in the image frame.

The aim of the fitting algorithm is to estimate the shape and texture model parameters that minimize the square of the norm of the texture error.

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \|\delta\mathbf{t}\|^2 \tag{22}$$

For efficiency reasons, the texture difference is projected onto a constant matrix, which yields the shape and texture model parameters update.

$$\begin{pmatrix} \delta\boldsymbol{\alpha} \\ \delta\boldsymbol{\beta} \end{pmatrix} = \mathbf{A} \cdot \delta\mathbf{t} \tag{23}$$

Then the next estimate is obtained by adding the update to the current estimate.

$$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \delta\boldsymbol{\alpha} \quad \text{and} \quad \boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \delta\boldsymbol{\beta} \tag{24}$$

This algorithm would be a gradient descent optimization algorithm, if the matrix relating the texture error to the model update, the matrix $\mathbf{A}$, was the inverse of the Jacobi matrix. Assuming $\mathbf{A}$ to be constant, is equivalent to assuming that the model Jacobi matrix is constant and hence that the rendering model is linear. However, the sources of nonlinearities of the rendering model are multiple: (i) The out of the image plane rotation and the perspective effects induce a nonlinear variation of the shape points projected onto the image plane. (ii) The modification of the light source direction produce a nonlinear variation in pixel intensities. (iii) The warping of the texture using the shape parameters is nonlinear as well. As the face model, on which this fitting algorithm is applied, is 2D, allow only small out of the image plane rotation, and does not model directed light sources, the first two sources of nonlinearities could be limited. Thus, the authors showed that this fitting was effective on facial images with no directed light and with small pose variations. However, it does not produce satisfactory results on the full 3D problem addressed in this chapter.

The constant matrix relating the texture error to the model update, the matrix $\mathbf{A}$, was first computed by a regression using texture errors generated from training images and random model parameters displacement. Then, it was estimated by averaging Jacobian matrix obtained by numerical differentiation on typical facial images. A problem with these two approaches is that not only the pixels inside the face area, but also the ones outside, on the background area, are sampled to form the training texture error. Thus, the quality of the estimate obtained on an input image depends on the resemblance of the background of this image to the one of the images used for computing the matrix $\mathbf{A}$.

As mentioned in the introduction of this chapter, the AAM algorithm is an instance of a fitting algorithm that favors efficiency over accuracy and generality.

To enlarge the domain of application of the Active Appearance Model to faces viewed from any azimuth angle, Cootes *et al.* [14] introduced the multi-view AAM. It is constituted of five AAMs, each trained on facial images at different pose (front, left and right side views and left and right profile). So, the fitting was also constituted of five constant Jacobi matrices. This is an ad-hoc solution addressing one of the limitations of a 2D model that was not pursued afterward.

Inverse Compositional Image Alignment algorithm As aforementioned, for efficiency reasons, the AAM treats the matrix relating the texture error to the model parameter update as constant. This is based on the assumption that the Jacobi matrix (that should be recomputed at each iteration) is well approximated by a constant matrix. However, this matrix is not constant owing to the warping of the texture by the shape. Baker and Matthews [1] introduced the Inverse Compositional Image Alignment (ICIA) algorithm, that also uses a constant Jacobi matrix, but, here, the matrix is shown, to a first order, to be constant. The fixedness of the updating matrix is not assumed anymore. This was achieved by a modification of the cost function to minimize. Instead of optimizing Equation (22), the following cost function is minimized.

$$\min_{\delta\boldsymbol{\alpha}} \|\mathbf{t}(u,v;0) \circ \mathbf{p}(u,v;\delta\boldsymbol{\alpha}) - I(x,y) \circ \mathbf{p}(u,v;\boldsymbol{\alpha})\|^2 \tag{25}$$

To clarify the notations, in what follows, the dependency on the frame coordinates (x,y) and (u,v) is not explicit anymore; only the dependency on the model parameters is left. A first Taylor expansion of the term inside the norm of the cost function yields:

$$\min_{\delta\boldsymbol{\alpha}} \|\mathbf{t}(0) \circ \mathbf{p}(0) + \mathbf{t}(0) \left.\frac{\partial \mathbf{p}}{\partial \boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}=0} \delta\boldsymbol{\alpha} - I \circ \mathbf{p}(\boldsymbol{\alpha})\|^2 \tag{26}$$

Differentiating this cost function with respect to the shape parameter update, equating to zero, and rearranging the terms, yields the expression of the parameter update:

$$\delta\boldsymbol{\alpha} = \left[\mathbf{t}(0) \left. \frac{\partial \mathbf{p}}{\partial \boldsymbol{\alpha}} \right|_{\boldsymbol{\alpha}=0} \right]^{\dagger} (I \circ \mathbf{p}(\boldsymbol{\alpha}) - \mathbf{t}(0) \circ \mathbf{p}(0)) \tag{27}$$

where the notation † denotes the pseudo-inverse of a matrix. This derivation leads to a Gauss-Newton optimization [19]. In a least squares optimization, the Hessian is a sum of two terms: the Jacobi matrix transposed and multiplied by itself and the purely second derivative terms. In a Gauss-Newton optimization, the Hessian is approximated by its first term only. The reason is that in a least square optimization (such as $\min_x \sum_i e_i^2$), the second derivative term of the Hessian of the cost function is the sum of the Hessian matrices of the elements, $\frac{\partial^2 e_i}{\partial x^2}$, multiplied by their residual, e_i. Near to the minimum, the residuals e_i are small, and the approximation is adequate. A Gauss-Newton optimization algorithm is more efficient than a Newton algorithm, as the second derivatives are not computed. It is not surprising that the ICIA algorithm is equivalent to a Gauss-Newton optimization, as it was derived using a first order Taylor approximation.

The essence of the ICIA algorithm, is that the Jacobi matrix, $\mathbf{t}(0) \left. \frac{\partial \mathbf{p}}{\partial \boldsymbol{\alpha}} \right|_{\boldsymbol{\alpha}=0}$ does not depend on the current estimate of the model parameters, neither on the input image. It is then constant throughout the fitting algorithm. This constancy is not assumed but derived from a first order expansion. The update is, then, not added to the current estimate as in the case of the AAM fitting algorithm, but composed with the current estimate.

$$\mathbf{p}(u, v; \boldsymbol{\alpha}) \leftarrow \mathbf{p}(u, v; \boldsymbol{\alpha}) \circ \mathbf{p}^{-1}(u, v; \delta\boldsymbol{\alpha}) \tag{28}$$

Baker *et al.*, in [1], do not make the distinction between the reference and the image frames. A consequence of this, is that they require the set of warps to be closed under inversion. This leads them to a first order approximation of the inverse shape projection (called inverse warping in their nomenclature): $\mathbf{p}^{-1}(x, y; \boldsymbol{\alpha}) = \mathbf{p}(u, v; -\boldsymbol{\alpha})$. This does not agree with the identity defining the inverse shape projection of Equation (9): As shown in Figure 4, a point from (u, v), q', is mapped under $\mathbf{p}(u, v; \boldsymbol{\alpha})$ to q in (x, y). Hence, to agree with the identity, this point q must be warped back to q' under $\mathbf{p}^{-1}(x, y; \boldsymbol{\alpha})$. So, the displacement in q which should be inverted is the one from q'. However, in Baker *et al.* [1], the displacement function $\mathbf{p}$ is inverted at the point q, leading to the point b, instead of q'. This is due to the fact that the distinction between the two coordinates systems is not made. This approximation is less problematic for a sparse-correspondence model as used by Baker for which the triangles are quite large (see Image (b) of Figure 2 of [1]), because the chances that both q and q' fall in the same triangle are much higher than in our dense correspondence model for which the triangles are much tinier. When q and q' fall in the same triangle, then their displacements are similar to a first order approximation, due to the linear interpolation inside triangles, and the error made during composition is small.

The improvement of ICIA over AAM is that, as the updating matrix is derived mathematically from the cost function and not learned over a finite set of examples, the algorithm is more accurate and requires less iterations to converge. The fact that the Jacobi matrix is constant is the first factor of the efficiency of the ICIA algorithm. The second factor is the fact that only the shape parameters are iteratively updated. The texture parameters are estimated in a single step after the recovery of the shape parameters. This is achieved by making the shape Jacobi matrix $\mathbf{t}(0) \left. \frac{\partial \mathbf{p}}{\partial \boldsymbol{\alpha}} \right|_{\boldsymbol{\alpha}=0}$ orthogonal to the texture Jacobi matrix, by projecting out the shape Jacobi matrix onto the texture Jacobi matrix. It is therefore called the *project out* method. This induces a perturbation on the shape Jacobi matrix. However, if the texture model has few components (Baker and Matthews use less than ten components), then the error is small.

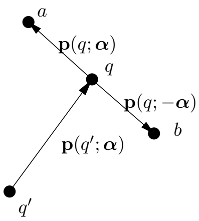

Figure 4: First order approximation of the inverse shape projection defined by Baker and Matthews in [1].

ICIA is an efficient algorithm. However, its domain of application is limited. It is a fitting algorithm for 2D AAM, hence cannot handle out of the image plane rotation and directed light. After discussion with Simon Baker, it also appears that it achieves its best performance when fitting images of individuals used for training the model. On novel subjects, the performance and accuracy are comparable to the AAM fitting algorithm. The ICIA is hence a person-specific fitting algorithm.

ICIA applied to the 3DMM The ICIA fitting algorithm was adapted to use the 3DMM in [39]. Several modifications of the original algorithm had to be made in order to obtain accurate results. The first one was to use the precise inverse shape projection defined in Section 0.5.2, which makes the distinction between the reference and the image frames. The use of this inverse shape projection leads to the definition of the following cost function:

$$\|\mathbf{t}(u,v;\Delta\boldsymbol{\beta}) \circ \mathbf{p}^{-1}(x,y;\boldsymbol{\gamma}^{\boldsymbol{d}}) \circ \mathbf{p}(u,v;\boldsymbol{\gamma}^{\boldsymbol{d}}+\Delta\boldsymbol{\gamma}) - \mathbf{t}^{-1}(I(x,y)\circ \mathbf{p}(u,v;\boldsymbol{\gamma});\boldsymbol{\beta})\|^2 \tag{29}$$

where $\boldsymbol{\gamma}$ is the vector formed by the concatenation of the shape parameters, $\boldsymbol{\alpha}$, and the rigid transformation parameters, $\boldsymbol{\rho}$. The derivatives are precomputed at the parameters $\boldsymbol{\gamma}^{\boldsymbol{d}}$. Projecting out the shape Jacobi matrix onto the texture Jacobi matrix would significantly perturb the shape update, hence the project out method was not used. Thus, the texture parameters, $\boldsymbol{\beta}$, are also iteratively updated. The texture parameter update is denoted by the vector $\Delta\boldsymbol{\beta}$. This required the definition of the inverse texture transformation:

$$\mathbf{t}^{-1}(\mathbf{t}(u_i,v_i);\boldsymbol{\beta}) = \mathbf{t}(u_i,v_i) - \sum_{k=1}^{N_T} \beta_k \cdot \mathbf{T}^k_{\cdot,i}, \tag{30}$$

where $\mathbf{T}^k_{\cdot,i}$ denotes the i^{th} column of the matrix $\mathbf{T}^k$, i.e. the deviation of the RGB color of vertex i along the principal component k. This definition was 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: $\mathbf{t}^{-1}(\mathbf{t}(u_i,v_i;\boldsymbol{\beta});\boldsymbol{\beta}) = \overline{\mathbf{T}}_{\cdot,i}$, see Equation (5 on page 7).

This algorithm is not as efficient as the original ICIA, but it is more accurate and its domain of applicability is also wider: It is able to fit input facial images at any pose and of any individual. This was demonstrated in the identification experiments reported in [37]. However, this algorithm does not handle directed light sources. There is a second predicament of this algorithm. In an implementation of ICIA, the parameters at which the derivatives are computed $\boldsymbol{\gamma}^{\boldsymbol{d}} = [\boldsymbol{\alpha}^{\boldsymbol{d}^{\mathrm{T}}} \boldsymbol{\rho}^{\boldsymbol{d}^{\mathrm{T}}}]^{\mathrm{T}}$ must be selected. A natural choice for the shape parameters is $\boldsymbol{\alpha}^{\boldsymbol{d}} = 0$. The selection of $\boldsymbol{\rho}^{\boldsymbol{d}}$ is not as trivial, because the derivatives of the shape projections are computed in a particular image frame set by θ^d and by ϕ^d. Therefore, the two rotation angles should be close to their optimal values (depending on the input image). Hence, a set of Jacobian's is computed for a series of different directions. During the iterative fitting, the derivatives used are the ones closest to the current estimation of the angles. Note that, at first, this approach might seem very close to the View-based approach [14, 32, 34]. 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.

2D+3D Active Appearance Model Recently, Xiao *et al.* [48] extended the ICIA algorithm, originally developed for to the fitting of 2D AAM, to the fitting of a 2D+3D AAM. The aim of this fitting algorithm is to recover the 3D shape and appearance of a face very efficiently (more than 200 fps). They argued that the difference between a 2D AAM and the 3DMM is that the 3DMM codes, additionally to the (X,Y) coordinates, the Z coordinate for each shape vertex. We will see, shortly, that there is an additional major difference between these two models. Xiao *et al.* [48] showed that, in fact, any 2D projection of a 3D shape in the span of a 3DMM can also be instantiated by a 2D AAM, but at the expense of using more parameters. The 2D AAM requires up to 6 times more parameters than the 3DMM to model the same phenomenon. A weak perspective projection model was used to demonstrate this. This property would not hold for a perspective projection. A weak perspective projection is governed by the following equation:

$$x_i = t_x + f\mathbf{W}_{1,i} \qquad y_i = t_y + f\mathbf{W}_{2,i} \tag{31}$$

Xiao *et al.* also showed that such a 2D AAM would be capable of instantiating invalid shapes, which is natural, as 3D transformations projected to 2D are not linear in 2D. The conclusion is that it is possible to fit facial images with non frontal poses with a 2D AAM trained on frontal pose. To ensure the validity of the shape estimated and to increase the efficiency of the algorithm, Xiao *et al.* impose the constraint that the 2D shape is a legitimate projection of a 3D shape modeled by a 3DMM. Thus, 3DMM shape model parameters and weak projection parameters are required to exist such as they produce a 2D shape equal to the one estimated by the 2D fitting algorithm. This is implemented as a soft constraint by augmenting the original ICIA cost function by a term proportional to the discrepancy between the 2D AAM estimated shape and the projection of the 3DMM shape. This seems to be a rather inefficient solution as, here, two shapes have to be estimated: The 2D AAM shape as well as the 3DMM shape and the projection parameters. The 3DMM model is only used to ensure the validity of the 2D AAM shape. The 2D AAM shape is used to warp the shape-free texture onto the image frame.

Xiao *et al.* [48] argued that the difference between a 2D AAM and the 3DMM is that the 3DMM codes depth information for each vertex and the 2D AAM does not. There is another major difference. The 3DMM is a dense model whereas the AAM is a sparse shape model. 76000 vertices are modeled by the 3DMM and a few tens by the AAM. This dense sampling enables the 3DMM to separate the texture from the illumination, thereby estimating a texture free of illumination effects. This is because the shading of a point depends mostly on the normal of this point and on its reflectance properties. (The self-cast shadow term of the illumination depends also on the full 3D shape of the object.) Therefore, to accurately model the illumination, it is required to accurately model the normals. The normal of a point depends on the local 3D shape in a neighborhood of this point. Hence, the 3D surface is required to be densely sampled in order to permit an accurate computation of the normals. Thus, it is not possible for a sparse 3D shape model to accurately separate the shading from the texture. It would then be difficult to relight a facial image with a different lighting configuration, as it is possible with the dense 3DMM, or to obtain high identification rates on a face recognition application across illumination.

We have just seen that it would be difficult to estimate the illumination parameters from a coarse shape model, for the normal may not be computed accurately. It seems also unclear how to extend this algorithm to estimate the illumination while retaining the constancy of the Jacobi matrix property. This might be the reason why this algorithm has never been used to estimate the lighting and to compensate for it.

As demonstrated in [48], the 2D+3D ICIA algorithm recovers accurately the correspondence between the model 3D vertices and the image. These vertices are located on edges (eyebrows, eyes, nostrils, lips, contour). The edge features are, hence, implicitly used, through the input image gradient, to drive the fitting. Although, the correspondences are recovered, it does not imply that the estimated Z values of the landmarks is close to the Z value of the corresponding physical points on the face surface. In a single 2D image the only depth information is contained in the lighting. To estimate accurately the 3D shape of a surface, the lighting must be estimated and its 3D shape recovered using a reflectance model. For example, in a frontal image, the only clue about the distance between the nose tip and, say, the lips, is in the shading of the nose. Failing to take the shading into account in a fitting algorithm, as it is done by the 2D+3D AAM fitting algorithm, results in an imprecise 3D shape.

As the original ICIA fitting algorithm, the 2D+3D ICIA fitting is person-specific: it is able to fit accurately only individual within the training set. Its main domain of application is real-time 3D face tracking.

Linear Shape and Texture Fitting algorithm LiST [38] is a 3DMM fitting algorithm that addresses the same problem as the SNO algorithm in a more efficient manner by use of the linear parts of the model. (A fitting is 5 times faster than with the SNO algorithm.) It is based on the fact that if the correspondences between the model reference frame and the input image are estimated, then fitting the shape and rigid parameters is a simple bilinear optimization that can be solved accurately and efficiently. To obtain a bilinear relationship between the 2D vertices projection and the shape and rigid parameters, a weak-perspective projection is used (Equation (31). One way of estimating these correspondences is by the use of a 2D optical flow algorithm [3]. Hence an optical flow algorithm is applied to a model image synthesized using the current model parameters and the input image. These correspondences are then used to update the shape and rigid parameters. The texture and illumination parameters can also be recovered efficiently using the correspondences: The input image is sampled at the location given by the correspondences and first the illumination parameters are recovered using a Levenberg-Marquardt optimization [36], while keeping the texture parameters constant. This optimization is fast as only a few parameters need to be estimated. Then using the estimated light parameters, the light effect of the extracted texture is inverted, yielding an illumination-free texture used to estimate the texture parameters. The texture parameters are recovered by inverting a linear system of equations. It is hence efficient and provide an accurate estimate.

A drawback of this algorithm is that the 3D shape is estimated using correspondence information only, not using the shading, as it is done in the SNO algorithm.

0.7 Results

We explained in Section 2 that the Morphable Face Model is a representation of human face images in which the pose and illumination parameters are separated from the shape and texture parameters. Then, in Section 6 we outlined SNO, a 3D Morphable Model Fitting algorithm that estimate the 3D shape, the texture and the imaging parameters from a single facial image. Some example images obtained after fitting and pose and illumination normalization are displayed on Figure 1 on page 1. Examples of illumination normalization are shown on Figure 5. The images of the first row, illuminated from different directions, are fitted. Renderings of the fitting results are shown on the second row. The same renderings, but using the illumination parameters from the left-most input image, appear on the third row. The last row presents the input image with illumination normalized to the illumination of the left-most image.

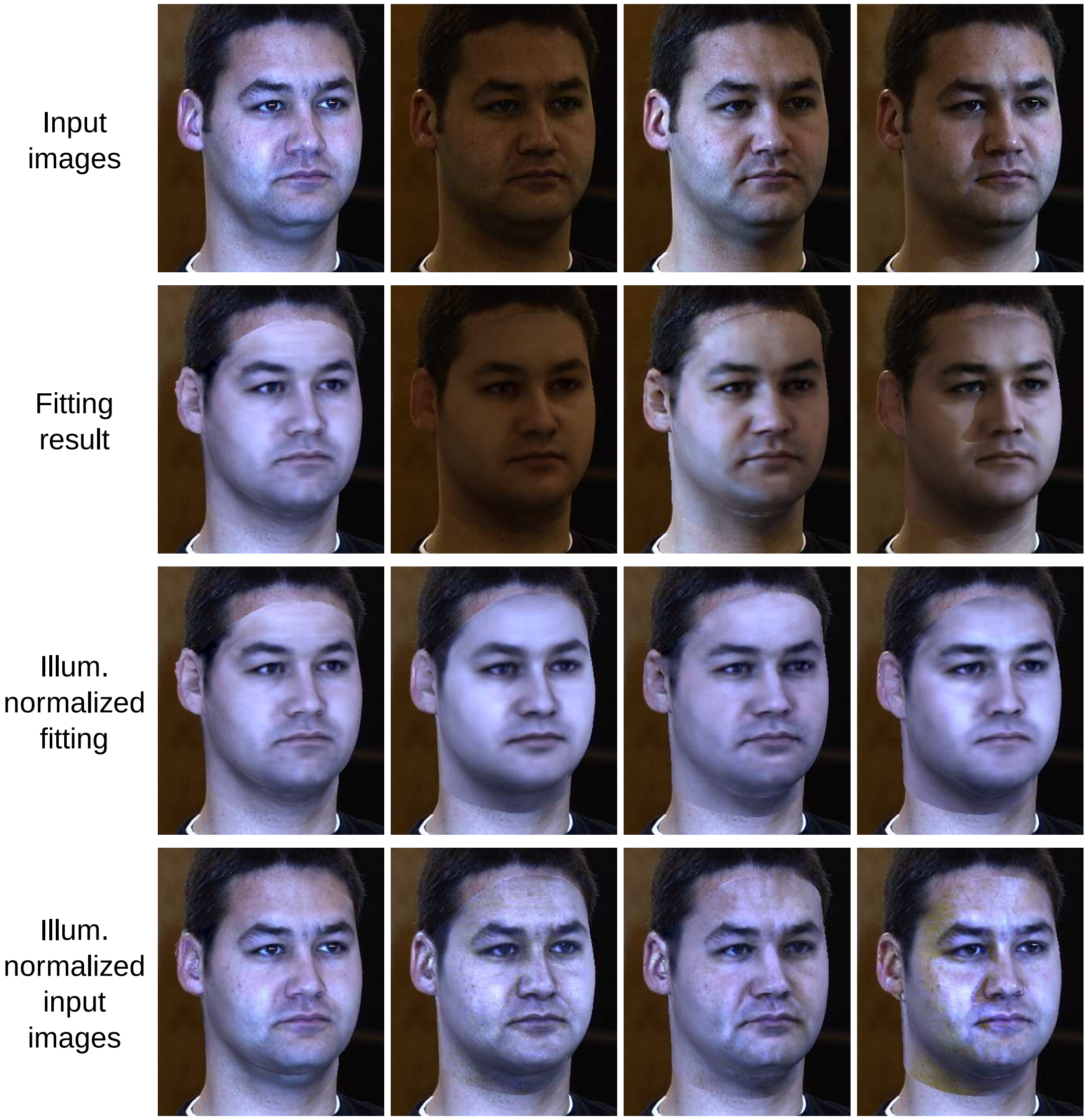

Figure 5: Demonstration of illumination normalization on a set of input images obtained using the SNO fitting algorithm. Renderings of the input image of the top row are shown on the second row. The same renderings with standard illumination, taken from the left-most input image, are displayed on the third row. Finally, the rendering using the extracted texture taken from the original images and again with the standard illumination appear on the bottom row.

0.7.1 Identification Results

In this section, the 3D Morphable Model and its fitting algorithm are evaluated on an identification application; these results were first published in [6]. In an identification task, an image of an unknown person is provided to the 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. It is assumed that the individual in the unknown image is in the gallery.

Dataset. We evaluate our approach on two datasets. *Set 1*: a portion of the FERET dataset [35] 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 a smiling expression that is not accounted for by the current 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 [43] containing images of 68 individuals at 3 poses (frontal, side and profile) and illuminated by 21 different directions and by ambient light only. Among the 68 individuals, 28 wear glasses, which are not modeled and could decrease the accuracy of the fitting. None of the individuals present in these sets were 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.

Distance Measure. 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 0.5.1 on page 7) and rescaling them by their standard deviations:

$$\mathbf{c} = \left[\frac{\alpha_1^g}{\sigma_{S,1}}, \ldots, \frac{\alpha_{99}^g}{\sigma_{S,99}}, \frac{\beta_1^g}{\sigma_{T,1}}, \ldots, \frac{\beta_{99}^g}{\sigma_{T,99}}, \frac{\alpha_1^{s_1}}{\sigma_{S,1}}, \ldots, \frac{\alpha_{99}^{s_1}}{\sigma_{S,99}}, \ldots, \ldots, \frac{\beta_{99}^{s_4}}{\sigma_{T,99}}\right]^{\mathtt{T}} \tag{32}$$

We define a distance measures to compare two identity parameters $\mathbf{c}_1$ and $\mathbf{c}_2$. The measure, d, 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 which does not modify the perceived identity:

$$d = \frac{\mathbf{c}_1^{\mathtt{T}} \cdot \mathbf{C}_W^{-1} \cdot \mathbf{c}_2}{\sqrt{\left(\mathbf{c}_1^{\mathtt{T}} \cdot \mathbf{C}_W^{-1} \cdot \mathbf{c}_1\right)\left(\mathbf{c}_2^{\mathtt{T}} \cdot \mathbf{C}_W^{-1} \cdot \mathbf{c}_2\right)}} \tag{33}$$

In this equation, $\mathbf{C}_W$ is the covariance matrix of the within-subject variations. It is computed on the FERET fitting results for the identification of the CMU-PIE dataset and vice-versa.

Table 1 lists percentages of correct rank 1 identification obtained on the FERET dataset. The pose chosen as gallery is the one with an average azimuth angle of $11.2°$, i.e. the condition *be*.

The CMU–PIE dataset is used to test the performance of this method in presence of combined pose and illumination variations. Table 2 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.

0.7.2 Improved Fitting using an Outlier Map

The major source of fitting inaccuracies is the presence of outlying pixels in a facial image. An outlying pixel, or outlier, is a pixel inside the face area of an image whose value cannot be predicted by the model nor by the noise distribution assumed by the fitting cost function[1]. Typical examples of outliers are glasses, specular highlight due to the presence of glasses, and occluding objects such as facial hair. Naturally, a black pixel due to facial hair may be predicted by the model, but doing so, would substantially modify the model parameters and deteriorate the fitting of the rest of the face. This is shown on the top row images of Figure 6. Thus, fitting an outlier is prejudicial

[1] We assumed that the noise is independent and identically distributed over all pixels of the image with a normal distribution.

Probe View	Pose ϕ	Correct Identification
bb	38.9°	94.8%
bc	27.4°	95.4%
bd	18.9°	96.9%
be	11.2°	99.5
ba	1.1°	*gallery*
bf	−7.1°	97.4%
bg	−16.3°	96.4%
bh	−26.5°	95.4%
bi	−37.9°	90.7%
bk	0.1°	96.9%
mean		*95.9%*

Table 1: Percentage of correct identification on the FERET dataset obtained using the SNO fitting algorithm. The gallery images are the view *be*. ϕ denotes the average estimated azimuth pose angle of the face. Ground truth for ϕ is not available. Condition *bk* has different illumination than the others. (Results from [6]).

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 2: Mean percentage of correct identification obtained on PIE images using the SNO fitting algorithm, 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%**. (Results from [6]).

not only because the fitted model parameter would account for an artifact such as hair but also because it would substantially decrease the overall quality of the fitting in the outlier-free region of the face. Discarding all pixels with a large residual is not a general solution to this problem: Some inliers may have a large fitting error and discarding them would jeopardize the quality of the fitting. This may happen at the beginning of the fitting algorithm when there is an important lack of correspondence. For instance, the model pupil may overlap the skinny area between the eye and the eyebrows inducing a large residual. It is this residual which is to drive the model parameters to improve the correspondence and, hence, it should not be down-weighted.

To appreciate the importance of this problem, the fitting of the same image was performed with the outlier pixels discarded. To do so, a outlier mask was automatically generated. The mask is shown on the first image of the bottom row of Figure 6: The brighter pixels are treated as outliers and are not sampled in the sum of the energy function of Equation (20). The visual quality of the reconstruction yielded by this fitting, shown on the middle of the bottom row of the figure, is clearly improved. The rendering of the novel view (last column) is also superior when the outlier are excluded.

To automatically produce an outlier mask, we use the following algorithm: First a coarse fitting is performed without outlier mask, but with a large weight on the prior probability (σ_I^2 is increased

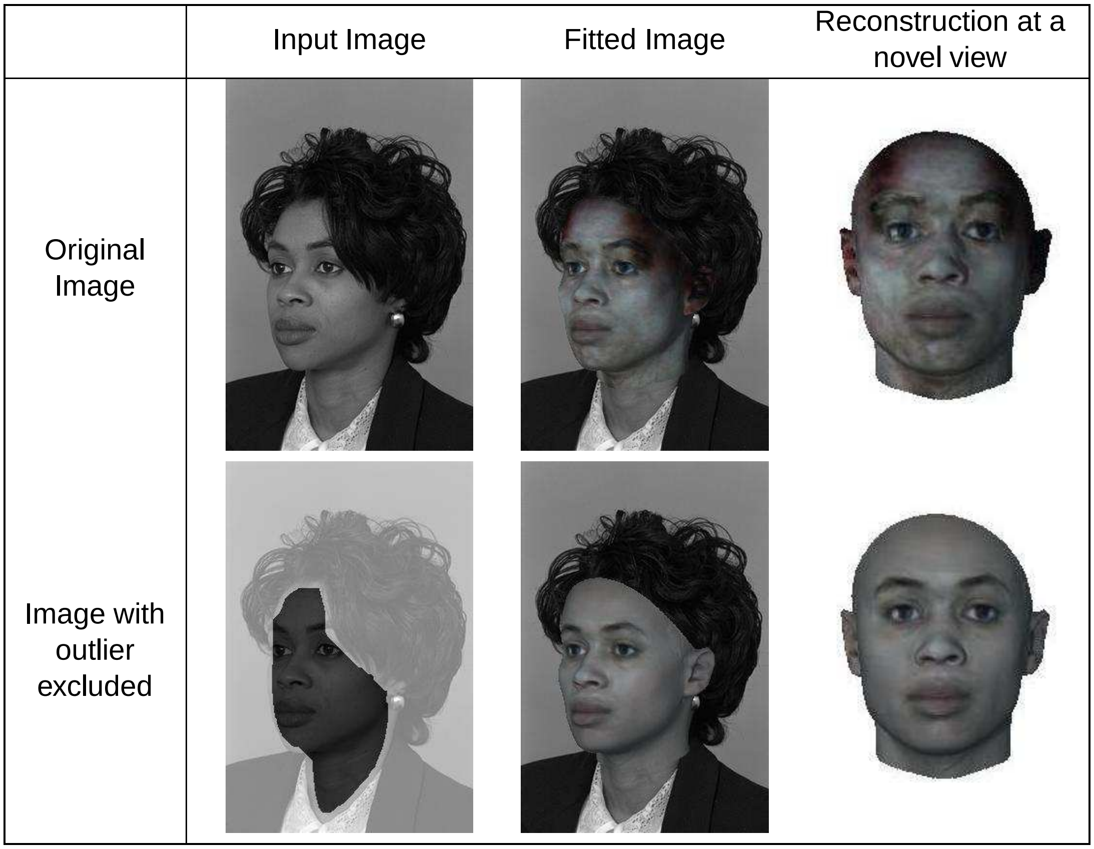

Figure 6: Example of the benefit of excluding the outlier part of a face image. Top row: The second image is a rendering of the fitting result obtained by fitting the first image. The right image is a rendering of the same fitting result at a frontal pose. Bottom row: The first image is the input image with the outlier region shown in bright. This region was excluded to produce the fitting shown on the second image. The third image is a rendering of this fitting result at a frontal pose.

in Equation 20 on page 11). Only a few iterations of the fitting algorithm are necessary at this stage. A rendering of the result of this fitting is shown on the first image of Figure 7. Then five image patch are selected in the face area. These image patches are the one with minimum residual; they are shown on the second image of Figure 7. These image patches are used to initialize a *GrabCut* algorithm, developed by Rother *et al.* [41], that segments the skin region from the non-skin part of the image, thereby producing the outlier maks. GrabCut is an image segmentation algorithm that combines color and contrast information together with a strong prior on region coherence. Foreground and background regions are estimated by solving a graph cut algorithm.

0.8 Conclusion

Recognizing a signal by explaining it, is a standard strategy that has been used successfully. The two conditions to apply this technique is to obtain a model that can account for the input signals in their whole diversity and an algorithm to estimate the model parameters explaining a signal.

We showed in this chapter, that an accurate and general model is one that separates the sources of variations of the signals and represent them by independent parameters. As well as identity, the sources of variations of facial images include pose changes and the lighting changes. To accurately account for pose and illumination variations, Computer Graphics proposed a 3D object centered

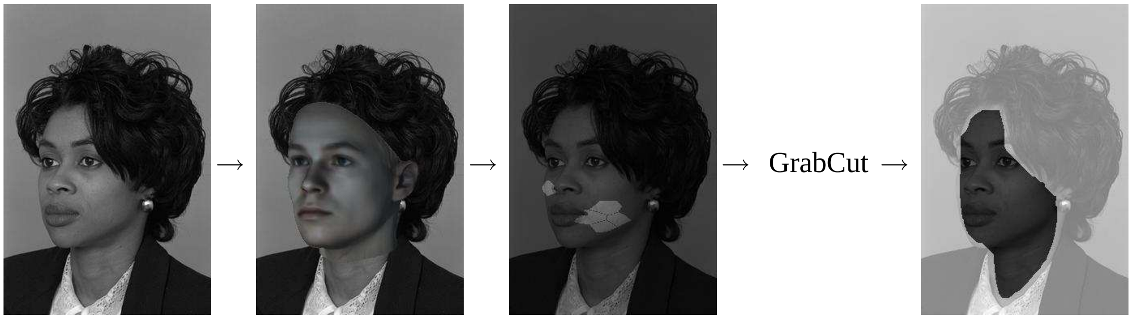

Figure 7: Automatic outlier mask generation. The outlier mask is produced by first performing a coarse fitting (second image) of the input image (first image). Then the five image patch with minimum residual error are selected (third image). The outlier mask is obtained by applying GrabCut using as foreground the five image patches selected.

representation. On the other hand, linear combinations of exemplar faces are used to produce face of novel individual. This produces a valid face, if the faces are represented on a common reference frame, on which all facial features are labeled.

We proposed, in this chapter, the 3D Morphable Model, a model that takes advantage of the 3D representation and of the correspondence principle, to account for any individual, viewed from any angles and under any light direction. The second ingredient of a analysis-by-synthesis loop is the model inversion, or analyzing, algorithm. The most general and accurate algorithm proposed so far is the Stochastic Newton Optimization whose update, at each iteration, is based on the first and second derivatives of a MAP energy function. The first derivatives are computed at each iteration, thereby favoring accuracy over efficiency. A stochastic optimization scheme was chosen to reduce the risk of locking into a local minimum.

Additionally to SNO that gives greater importance to accuracy and generality, there exists other fitting algorithm that favor efficiency at the expense of the domain of application and the precision. In the light of the 3D Morphable Model, we outlined the principles of the major fitting algorithms, and described their advantages and predicaments. The algorithm reviewed are the Active Shape Model, the Active Appearance Model, the Inverse Compositional Image Alignment (ICIA) algorithm, ICIA applied to 3DMM, 2D+3D Active Appearance Model, and the Linear Shape and Texture fitting algorithm.

Bibliography

[1] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[2] C. Basso, T. Vetter, and V. Blanz. Regularized 3d morphable models. In *Proc. of IEEE Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK 2003)*, 2003.

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

[4] V. Blanz. *Automatische Rekonstruction der dreidimensionalen Form von Gesichtern aus einem Einzelbild*. PhD thesis, Universität Tübingen, 2001.

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

[6] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003.

[7] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 187–194, Los Angeles, 1999. Addison Wesley Longman.

[8] M.E. Brand. Morphable 3D models from video. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[9] T. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. In *Proc. British Machine Vison Conference*, 1991.

[10] T. Cootes, G. Edwards, and C. Taylor. Active appearance model. In *Proc. European Conference on Computer Vision*, volume 2, pages 484–498. Springer, 1998.

[11] T. Cootes and C. Taylor. Active shape models- - smart snakes. In *Proc. British Machine Vision Conference*, 1992.

[12] T. Cootes, C. Taylor, A. Lanitis, D. Cooper, and J. Graham. Building and using flexible models incorporating grey-level information. In *Proceedings of the 4th International Conference on Computer Vision*, pages 355–365, 1993.

[13] T. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 266–275, Berlin, 1992. Springer.

[14] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *Fourth International Conference on Automatic Face and Gesture Recognition*, pages 227–232, 2000.

[15] I. Craw and P. Cameron. Parameterizing images for recognition and reconstruction. In Peter Mowforth, editor, *Proc. British Machine Vision Conference*, pages 367–370. Springer, 1991.

[16] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks*. Wiley, 1996.

[17] M. Dimitrijevic, S. Ilic, , and P. Fua. Accurate face models from uncalibrated and ill-lit video sequences. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

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

[19] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

[20] M. Gleicher. Projective registration with difference decomposition. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 331–337, 1997.

[21] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *Computer Graphics*, 30(Annual Conference Series):43–54, 1996.

[22] Ulf Grenander. *Pattern Analysis, Lectures in Pattern Theory*. Springer, New York, 1 edition, 1978.

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

[24] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proceedings of SIGGRAPH 2001*, pages 511–518, August 2001.

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

[26] 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.

[27] Marc Levoy and Pat Hanrahan. Light field rendering. *Computer Graphics*, 30(Annual Conference Series):31–42, 1996.

[28] D. Marr. *Vision,*. W. H. Freeman, San Fancisco, 1982.

[29] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, , Kenneth E. Torrance, and Donald P. Greenberg. Reflectance measurements of human skin. Technical Report PCG-99-2, 1999.

[30] T. P. Minka. Old and new matrix algebra useful for statistics. http://www.stat.cmu.edu/˜minka/papers/matrix.html, 2000.

[31] D. Mumford. Pattern theory: A unifying perspective. In D.C. Knill and W. Richards, editors, *Perception as Bayesian Inference*. Cambridge University Press, 1996.

[32] H. Murase and S.K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

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

[34] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 1994.

[35] P. J. Phillips, P. Rauss, and S.Der. Feret (face recognition technology) recognition algorithm development and test report. Technical report, U.S. Army Research Laboratory, 1996.

[36] Vetterling Press, Teukolsky and Flannery. *Numerical recipes in C : the art of scientific computing*. Cambridge University Press, Cambridge, 1992.

[37] S. Romdhani, V. Blanz, C. Basso, and T. Vetter. Morphable models of faces. In S. Z. Li and A. Jain, editors, *Handbook of Face Recognition*. Springer, 2005.

[38] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *Proc. European Conference on Computer Vision*, 2002.

[39] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *Proceedings of the International Conference on Computer Vision*, 2003.

[40] Sami Romdhani, Alexandra Psarrou, and Shaogang Gong. On utilising template and feature-based correspondence in multi-view appearance models. In *Proc. European Conference on Computer Vision*, 2000.

[41] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *Proc. ACM Siggraph*, 2004.

[42] S. Sclaroff and J. Isidoro. Active blobs. In *Proceedings of the 6th International Conference on Computer Vision*, 1998.

[43] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination and expression (pie) database of human faces. Technical report, CMU, 2000.

[44] S. Ullman. Aligning pictorial descriptions: An approach for object recognition. *Cognition*, 32:193–254, 1989.

[45] T. Vetter. Recognizing faces from a new viewpoint. In *ICASSP97 Int. Conf. Acoustics, Speech, and Signal Processing*, volume 1, pages 139–144, IEEE Comp. Soc. Press, Los Alamitos, CA, 1997.

[46] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):733–742, 1997.

[47] Thomas Vetter. Synthestis of novel views from a single face image. *International Journal of Computer Vision*, 28(2):103–116, 1998.

[48] J. Xiao, S. Baker, I. Matthews, R. Gross, and T. Kanade. Real-time combined 2d+3d active appearance model. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 535–542, 2004.