Regularized 3D Morphable Models

Curzio Basso, Thomas Vetter
University of Basel, Departement Informatik
Bernoullistrasse 16
4056 Basel, Switzerland
Curzio.Basso@unibas.ch

Volker Blanz
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
blanz@mpi-sb.mpg.de

Abstract

Three-dimensional morphable models of object classes are a powerful tool in modeling, animation and recognition. We introduce here the new concept of regularized *3D morphable models, along with an iterative learning algorithm, by adding in the statistical model a noise/regularization term which is estimated from the examples set. With regularized 3D morphable models we are able to handle missing information, as it often occurs with data obtained by 3D acquisition systems; additionally, the new models are less complex than, but as powerful as the non-regularized ones. We present the results obtained for a set of 3D face models and a comparison with the ones obtained by a traditional morphable model on the same data set.*

1. Introduction

Image morphing [16] consists in generating, given two input images, a sequence of new ones describing a smooth transition between the two inputs. Equally important is that at the same time it preserves all their *essential features*. Exploiting the fact that the image of an object depends on its *shape* and on its *appearance*, a correct morphing can be achieved by deforming the shapes of the input images in such a way that their features are aligned while their appearances are kept constant. A morphed image is obtained by interpolating with the desired coefficients the two deformed images, also known as *warped* or *shape-free* images [8]. Morphing among objects belonging to the same class is used in Computer Vision for implementing *analysis-by-synthesis* techniques, that use a statistical model of the images of a certain class in order to analyze them: instead of directly building a statistical model from the training images, one estimates their correspondence with a reference image and then build a linear model of the warped images [15, 3].

3D morphable models [5] are based on the same idea but use 3D meshes as training data rather than images, thus presenting the advantage of being independent from pose and illumination. The correspondence estimation in the case of 3D polyhedral surfaces [11, 12, 9, 1] consists in finding a mapping between the points of the input models. This is often performed interactively by the user, who selects a finite number of correspondences between the vertices of the two surfaces; from this sparse set, the correspondence for the entire surfaces is estimated. The reconstruction of the whole function can be more or less sophisticated, ranging from simple linear interpolation to RBF networks [17].The advantage of such an approach is that it can be as precise as one wants, by manually defining more and more correspondences; however, when morphing between more than two models (*polymorphs*) is required, manual selection of the desired features can prove impractical.

In the original paper the correspondences used in building the 3D morphable model were learnt via an automatic method [5]. By parameterizing in 2D the input 3D examples, the correspondences could be estimated using a modified version of the optical flow; the same algorithm had been in fact already employed for building morphable models of images [14]. In that context, the optical flow was wrapped into an iterative process estimating at the same iteration both the correspondences and the morphable model.

The positive effect on the reconstruction of partial data of the assumption of an additive Gaussian noise perturbing the complete data has been already clarified [6]. On the other hand, handling missing data is important not only in fitting 3D morphable models to novel examples, but in learning them as well, since acquisition systems do rarely provide 3D surface data without holes or other errors; also, the input data could present boundaries that have been artificially set, leading to errors in the correspondence estimation. We extend the 3D morphable models with the assumption that the training vectors with which they are built derive from a multivariate Gaussian distribution with additive Gaussian noise. We show how this assumption leads to an algorithm which

iteratively estimates the data distribution and the correspondence, and that compared to other iterative algorithms previously proposed ([14, 5, 4]) has the big advantage of being able to process partial information. We call the 3D morphable models derived in such a way *regularized*, because of the connection with regularization methods of statistical learning.

The paper is organized as follows:

- We first describe formally the polymorphs for 3D triangular meshes and the 3D correspondence problem (sec. 2); then we show how this problem can be rephrased as an image registration problem (sec. 2.1) and equipped with this equivalent formulation, we describe briefly a previous method used to solve the 3D problem by making use of optical flow (sec. 2.2).
- Assuming the 3D correspondence problem has been solved, in section 3 we summarize the statistical framework used to model the data distribution. This model is then used to give a definition of the regularized 3D morphable models.
- In our framework, the 3D correspondences and the statistical distribution of the data, as well as the reconstructions of missing information, are estimated together in an iterative process. This new learning algorithm is described in section 4.
- In section 5 we apply the learning algorithm to a training set of 200 3D models of human faces, showing the improvements both in term of quality of the 3D correspondence and of the resulting statistical model.

2. Correspondence Estimation

In order to define the correspondence problem in 3D, we limit the discussion to 3D triangular meshes and introduce a formal definition of them, by making use of 2-dimensional simplicial complexes. Such a complex K, with n vertices, is a set whose elements can be vertices $\{i\}$, edges $\{i,j\}$ or triangles $\{i,j,k\}$, with the indices $i,j,k \in [1 \dots n]$: it defines the connectivity of a given mesh $\mathcal{M}$. The actual shape and appearance of $\mathcal{M}$ are defined by two vectors $\mathbf{S} \in I\!R^{3n}$ and $\mathbf{T} \in I\!R^{3n}$: if $(x_i, y_i, z_i) \in I\!R^3$ is the position of the vertex $\{i\}$, and $(R_i, G_i, B_i) \in I\!R^3$ its color, $\mathbf{S}$ and $\mathbf{T}$ are defined as [1]

$$\begin{aligned} \mathbf{S} &= (x_1, y_1, z_1, \dots, x_n, y_n, z_n) \in I\!R^{3n} \\ \mathbf{T} &= (R_1, G_1, B_1, \dots, R_n, G_n, B_n) \in I\!R^{3n} \end{aligned}$$

Then, a 3D triangular mesh with colored vertices can be represented as

$$\mathcal{M} = (K, \mathbf{S}, \mathbf{T})$$

1 Usually (e.g.[10]) $\mathbf{S}$ and $\mathbf{T}$ are defined as *sets* of n vectors (x_i, y_i, z_i) and (R_i, G_i, B_i), but we prefer to stack all these vectors together in order to allow us to combine them linearly.

The correpondence problem arises when we have two meshes, $\mathcal{M}_1$ and $\mathcal{M}_2$, with n_1 and n_2 vertices respectively, and we want to morph between them. To perform the morph, we need to transform one of the meshes in such a way that it uses the same connectivity of the other mesh and at the same time it approximates well the original surface:

$$\mathcal{M}_2' = (K_1, \mathbf{S}_2', \mathbf{T}_2') \approx \mathcal{M}_2$$

The transformed mesh $\mathcal{M}_2'$ is called *warped* mesh, and with it we can write the morph as a parameterized mesh:

$$\begin{aligned} \mathcal{M}(c) &= (1-c)\mathcal{M}_1 + c\mathcal{M}_2' \\ &= (K_1, (1-c)\mathbf{S}_1 + c\mathbf{S}_2', (1-c)\mathbf{T}_1 + c\mathbf{T}_2') \end{aligned}$$

For the essentail features of $\mathcal{M}(c)$ to be kept intact while varying c, not only has the warped mesh to approximates the original one, but its i-th vertex has to correspond to the i-th vertex of $\mathcal{M}_1$. We need therefore a correspondence between the vertices of $\mathcal{M}_1$ and the surface defined by $\mathcal{M}_2$, so that we can build $\mathcal{M}_2'$ accordingly.

In order to be able to define such a correspondence, we need a parameterization of the 3D mesh, so that we can identify its points by a vector of coefficients. A natural parameterization is given by barycentric coordinates; any point in the triangle $\{i,j,k\}$ can be represented with three coefficients, b_i, b_j and b_k, which define its position (colour):

$$\mathbf{x} = b_i\mathbf{x}_i + b_j\mathbf{x}_j + b_k\mathbf{x}_k$$

and with additional constraint

$$b_i + b_j + b_k = 1$$

Note that if the point is on an edge, then only the coefficients relative to the extremes of the edge will be different from zero; in case the point is one of the vertices the corresponding coefficient will be 1. In general, any point of a 3D mesh is defined in barycentric coordinates which cover the whole mesh, by means of a vector $\mathbf{b} = (b_1, \dots b_n)$ with $\|\mathbf{b}\|_1 = 1$ and at most three coordinates different from zero.

Denoting by $\mathbf{b}$ a point of $\mathcal{M}_1$, and by $\mathbf{q}$ a point of $\mathcal{M}_2$, both in barycentric coordinates, a 3D *dense* correspondence is defined by a 1-1 continuous function

$$\mathbf{q} = \varphi(\mathbf{b})$$

Given the 3D correspondence, and denoting by $\mathbf{b}^i$ the barycentric coordinates of the i-th vertex of $\mathcal{M}_1$ and by $\mathbf{q}^i = \varphi(\mathbf{b}^i)$ its corresponding point in $\mathcal{M}_2$, the warp is a transformation where $\mathbf{S}_2'$ and $\mathbf{T}_2'$ now stack n_1 vectors (rather than n_2) defined by

$$\begin{aligned} (x_i, y_i, z_i)' &= \sum_{j=1}^{n_2} q_j^i (x_j, y_j, z_j) \\ (R_i, G_i, B_i)' &= \sum_{j=1}^{n_2} q_j^i (R_j, G_j, B_j) \end{aligned}$$

In fact, the warping is nothing else than a resampling of the original mesh $\mathcal{M}_2$, so that the sampling points have now the same number and the same positions (with respect to the features) as in $\mathcal{M}_1$.

The correspondence problem consists in choosing the optimal φ, so that the resulting morph $\mathcal{M}(c)$ is smooth over the parameters domain and preserves all the essential features. Although these 3D correspondences can be defined manually, an alternative strategy is to turn the 3D problem into an image registration problem, for which many automatic algorithms exist.

The scheme described above can be extended to a set of m meshes $\mathcal{M}_i$ to obtain a polymorph, by defining a common connectivity K and for each mesh a 3D correspondence between K and K_i. Then, denoting by $\mathbf{S}'_i$ and $\mathbf{T}'_i$ the shape and texture vectors of the warped mesh $\mathcal{M}'_i$, we can write the result of a polymorph as

$$\mathcal{M}(\mathbf{a},\mathbf{b}) = (K, \sum_{i=1}^{m} a_i \mathbf{S}'_i, \sum_{i=1}^{m} b_i \mathbf{T}'_i) \tag{1}$$

with the constraints $0 \leq a_i, b_i \leq 1$ (that is we stay in the convex hull of the examples). One should be aware however, that within this setting K must be limited to a set of vertices for which the correspondences can be defined to *all* the meshes $\mathcal{M}_i$, otherwise some of the vectors $\mathbf{S}'_i$ and $\mathbf{T}'_i$ could be incomplete and the equation (1) would be meaningless. We will see in section 3 how the morphable model, making use of the statistical information, allows to relax this restriction.

2.1. 3D Correspondence as 2D Registration

Two-dimensional parameterizations of the 3D meshes allow to represent them as images: then, their features are turned into image features and automated image registration algorithms can be applied.

Given a set $\mathbf{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^n\}$, where each $\mathbf{p}^i$ defines a position in $\mathbb{R}^2$ for the i-th vertex of the mesh, and a 1-1 mapping

$$\psi_{\mathbf{P}}(\mathbf{b}) \equiv \sum_{i=1}^{n} b_i \mathbf{p}^i$$

which maps each point of $\mathcal{M}$ to a point in the plane we have a 2D parameterization of the mesh defined by the inverse $\psi_{\mathbf{P}}^{-1}$. Using the 2D parameterization one can represent the 3D mesh with an image $\mathbf{I} : \mathbb{R}^2 \rightarrow \mathbb{R}^6$ (that is, the image has 6 channels) defined in such a way that

$$\mathbf{I} \circ \psi_{\mathbf{P}}(\mathbf{b}) = \sum_{i=1}^{n} b_i (x_i, y_i, z_i, R_i, G_i, B_i)$$

which means that the position and color of a point of $\mathcal{M}$ is the linear interpolation of (at most) three pixels of $\mathbf{I}$. Accordingly, all the 3D features of $\mathcal{M}$ are projected to features in the image, and the 3D correspondence between points of two meshes becomes a mapping $\mathbf{v} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, sometimes known as *flow field*. Given $\mathbf{v}$, the 3D correspondence φ from $\mathcal{M}_1$ to $\mathcal{M}_2$ can be found as

$$\varphi(\mathbf{b}) \equiv \psi_{\mathbf{P}_2}^{-1} \circ \mathbf{v} \circ \psi_{\mathbf{P}_1}(\mathbf{b})$$

The point $\mathbf{b}$ is first mapped to its 2D projection in $\mathbf{I}_1$ by $\psi_{\mathbf{P}_1}$; from there it is mapped by $\mathbf{v}$ to a corresponding position in $\mathbf{I}_2$, and it is finally back-mapped to a point in $\mathcal{M}_2$ by the inverse $\psi_{\mathbf{P}_2}^{-1}$.

The optimal $\mathbf{v}$ is, as explained in the introduction, not easy to define formally. In practice, one minimizes the cost function

$$E_{\mathbf{I}_1,\mathbf{I}_2}(\mathbf{v}) = \sum_{u,v} d(\mathbf{I}_2 \circ \mathbf{v}, \mathbf{I}_1) \tag{2}$$

taking care of choosing a distance function $d(\cdot,\cdot)$ which is appropriate (that is, it yields satisfying results) for the given problem setting.

$\mathbf{P}$ is derived by converting the vertex positions to cylindrical coordinates (ϕ, h, r):

$$(\phi_i, h_i, r_i) = (\tan^{-1}(x_i/y_i), z_i, \sqrt{x_i^2 + y_i^2})$$

and then setting a vertex 2D position to its azimuth and height values:

$$\mathbf{p}^i \equiv (\phi_i, h_i)$$

Note that with this parameterization, we can reduce the dimensionality of $\mathbf{I}$ since two of the three geometry coefficients (ϕ and h) are already given by the position in the image:

$$\mathbf{I} \circ \psi_{\mathbf{P}}(\mathbf{b}) = \sum_{i=1}^{n} b_i (r_i, R_i, G_i, B_i)$$

As distance measure we use a weighted L_2-norm of the form

$$d(A,B) = \sum_{i \in \{r,R,G,B\}} w_i \|A_i - B_i\|^2 \tag{3}$$

where the weights w_i compensate for the dishomogeneity between the different dimensions.

The problem of finding the optimal $\mathbf{v}$ is what is known as image registration, and it is a fairly standard problem in computer vision, which however does not mean that it has been solved. The optical flow is one of the algorithms used to tackle the problem, and a modified version has been used in previous works in order to automatically solve the correspondence problem in 3D [5].

2.2. 3D Correspondence via Optical Flow

In its traditional form, the optical flow takes as input two gray-level images $I_1(x,y)$ and $I_2(x,y)$, assumed to be snapshots taken at different times of the same sequence

$I(x, y, t)$. The optical flow equation is derived from the fundamental assumption that objects in the sequence maintain their brightnesses:

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

Assuming that $I_1 = I(x, y, t_1)$ and $I_2 = I(x, y, t_1 + 1)$, and that in a neighborhood $R(x_0, y_0)$ of (x_0, y_0) the vector field $\mathbf{v}$ is approximately constant, a local solution for (x_0, y_0) of the above equation can be approximated by minimizing the cost function

$$E(x_0, y_0) = \sum_{R(x_0, y_0)} (\nabla I_2 \cdot \mathbf{v} + \Delta I)^2 \qquad (4)$$

where $\Delta I(x, y) = I_2(x, y) - I_1(x, y)$. Since the assumption of local constancy of $\mathbf{v}$ is in general not satisfied, it is advisable to use a coarse-to-fine strategy [2]: two Gaussian pyramids of the images are built, and then the algorithm is applied from the coarsest to the finest resolution, using the result of a coarser resolution to pre-warp the level currently processed. The rationale is that at the coarsest level of the Gaussian pyramid one expect the assumption of local constancy of the flow to be approximately exact, and at finer levels the pre-warping should compensate for the biggest variations on the field.

In the case of the image representations $\mathbf{I}_1$ and $\mathbf{I}_2$ of two 3D meshes $\mathcal{M}_1$ and $\mathcal{M}_2$, equation (4) is replaced by

$$E(h_0, \phi_0) = \sum_{h,\phi \in R(h_0, \phi_0)} \|\nabla \mathbf{I}_2(h, \phi) \cdot \mathbf{v}(h, \phi) + \Delta \mathbf{I}(h, \phi)\|^2$$

Note that this last equation can be derived from (2) by approximating $\mathbf{I}_2 \circ \mathbf{v}$ with a first order Taylor expansion and by rescaling the values of the images $\mathbf{I}_1$ and $\mathbf{I}_2$ with the weights w_i.

Although the optical flow should work in theory with the intensities of the images, experience shows that it is often better to use their gradient, and accordingly a Laplacian pyramid rather than a Gaussian one. This modification however makes the algorithm much more sensitive to the boundaries of the objects in the images, which can sometime lead to undesirable results. In fact, in case of projections of 3D meshes, missing data will lead to artificial boundaries which will introduce errors in the estimated correspondence; in section 4 we describe how to avoid this problem.

3. Regularized Morphable Models

Equation (1) defines a 3D polymorph, restricted by necessity to the minimum common domain of the different meshes between which we morph. We show now how we can overcome this restriction by exploiting the statistical information of a set of examples, building what we call a 3D morphable model.

Assume that both the geometry and texture *warped* vectors of the examples, $\mathbf{S}_i$ and $\mathbf{T}_i$, are obtained from a linear-Gaussian generative model. That is, we assume that all the shape vectors $\mathbf{S}$ (the argument is similar for texture vectors), once they are warped to a common space, are generated as

$$\mathbf{S} = \bar{\mathbf{s}} + \mathbf{C}\alpha + \epsilon \qquad (5)$$

where $\mathbf{C} \in I\!R^{3n \times k}$ with $k < 3n$ and both α and ϵ are random variables drawn from a Gaussian distribution:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0, \mathbf{I}) \\ \epsilon &\sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \end{aligned}$$

In case of $\sigma^2 = 0$, the relation between the observed variable $\mathbf{S}$ and the unobserved variable α would be completely determined by the mean $\bar{\mathbf{s}}$, which can be estimated with the sample average $(1/m) \sum_{i=1}^{m} \mathbf{S}_i$, and by the matrix $\mathbf{C}$. The model coefficients of a shape vector $\mathbf{S}$ are given by

$$\hat{\alpha} = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T (\mathbf{S} - \bar{\mathbf{s}})$$

and we look for the matrix $\mathbf{C}$ that minimizes the reconstruction error over the example set,

$$\|\mathbf{Y} - \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{Y}\|$$

where $\mathbf{Y}$ is the $3n \times m$ data matrix holding in its i-th column the vector $\mathbf{S}_i - \bar{\mathbf{s}}$. Principal Components Analysis (PCA) finds the solution by solving the eigenproblem of the data covariance matrix $\mathbf{Y}\mathbf{Y}^T$; without delving into details, it can be shown that by decomposing $\mathbf{Y}$ via Singular Value Decomposition (SVD),

$$\mathbf{Y} = \mathbf{U}\mathbf{W}\mathbf{V}^T$$

the optimal solution is given by $\mathbf{C} = \mathbf{U}\mathbf{W}$.

If $\sigma^2 > 0$, the solution is slightly more complex: the random variable $\mathbf{S}$ has the normal distribution

$$\mathbf{S} \sim \mathcal{N}(\bar{\mathbf{s}}, \mathbf{C}\mathbf{C}^T + \sigma^2 \mathbf{I})$$

and its model coefficients are

$$\begin{aligned} \hat{\alpha} &= (\mathbf{C}^T \mathbf{C} + \sigma^2 \mathbf{I})^{-1} \mathbf{C}^T (\mathbf{S} - \bar{\mathbf{s}}) \\ &= (\mathbf{W}^2 + \sigma^2 \mathbf{I})^{-1} \mathbf{W}\mathbf{U}^T (\mathbf{S} - \bar{\mathbf{s}}) \end{aligned} \qquad (6)$$

Although the mean can be still estimated via the sample average, PCA estimate of the matrix $\mathbf{C}$ is no longer optimal; however, an iterative EM-algorithm can be used to estimate both the generative matrix and the noise σ^2[13]. It is interesting to note that σ^2 is related to the *effective degrees of freedom* of the ridge regression fit

$$df(\sigma^2) = \sum_{i=1}^{k} \frac{w_i^2}{w_i^2 + \sigma^2} \qquad (7)$$

where w_i^2 is the variance of the i-th principal component. The above equation tells us that the greater the value of σ^2, the more constraint will the model be, since the value of $df(\sigma^2)$ decreases.

Using the EM-algorithm we can have an estimate of the *full* matrix $\mathbf{C}$ even if some of the elements of $\mathbf{S}$ are missing (or unreliable). Note that in this case the inference equation (6) would have to be modified, since the vector $\mathbf{S}$ would have fewer dimensions. Denoting by $\mathbf{C}'$ the matrix obtained by removing from $\mathbf{C}$ the rows corresponding to the missing dimensions of $\mathbf{S}$, and decomposing it by SVD into $\mathbf{C}' = \mathbf{U}'\mathbf{W}'\mathbf{V}'^T$, we recover the shape coefficients as

$$\begin{aligned}\hat{\alpha} &= (\mathbf{C}'^T\mathbf{C}' + \sigma^2\mathbf{I})^{-1}\mathbf{C}'^T(\mathbf{S} - \bar{\mathbf{s}}) \\ &= \mathbf{V}'(\mathbf{W}'^2 + \sigma^2\mathbf{I})^{-1}\mathbf{W}'\mathbf{U}'^T(\mathbf{S} - \bar{\mathbf{s}}) \\ &= \sum_{i=1}^{m-1} \mathbf{v}'_i \frac{w'_i}{w'^2_i + \sigma^2}\mathbf{u}'_i \cdot (\mathbf{S} - \bar{\mathbf{s}}) \qquad (8)\end{aligned}$$

In the case of more dimensions than examples, and assuming we want to retain all the $(m-1)$ principal components, the estimate of σ^2 given by the EM-algorithm would be zero, because the reconstruction error would be zero. However, assuming we have a test set of examples not used in building the model and which we denote by the matrix $\mathbf{Z}$, one can apply the maximum-likelihood estimator of σ^2, defined as

$$\sigma^2 = tr\left(\mathbf{Z}\mathbf{Z}^T - \hat{\mathbf{Z}}\mathbf{Z}^T\right)/m^2 \qquad (9)$$

where the matrix $\hat{\mathbf{Z}}$ holds the reconstructions of the test examples, derived from the inference equations (6) - full information - or (8) - missing information. In case no test set is available, cross-validation can still be used.

Given the generative model of equation (5), we can rewrite the equation (1) for a polymorph as

$$\mathcal{M}(\alpha, \beta) = (K, \bar{\mathbf{s}} + \mathbf{C}_s\alpha, \bar{\mathbf{t}} + \mathbf{C}_t\beta) \qquad (10)$$

where α and β are $(m-1)$-dimensional vectors. Note that the simplicial complex K in this case has not to be restrict to a common domain for which we have correspondence to all the examples, because all the elements of the matrices $\mathbf{C}_s$ and $\mathbf{C}_t$ will be defined even if for some of the examples no corresponding point was found. The above equation is the definition of a 3D morphable model.

By following the same procedure described in sec. 2.1, and denoting by ρ the coefficients of a rigid transformation, we can project the morphable model described by equation (10) to an image representation $\mathbf{M}_{\alpha\beta\rho}(x, y)$. Note that in this case the image resulting from the projection depends on the model coefficients α and β and on the rigid transformation defined by ρ. In particular, the 2D projections $\mathbf{p}^i$ and the first dimension of $\mathbf{M}$ will depend on α and ρ, while the remaining dimensions of $\mathbf{M}$ will depend on the β.

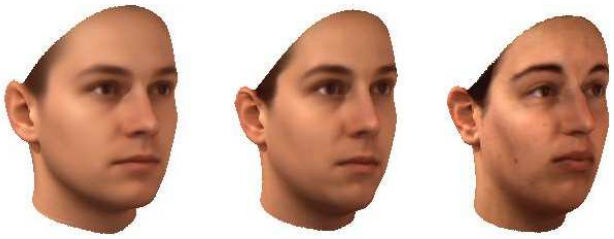

Figure 1. Rather than directly estimate the correspondence between the reference (left) and an example (right), an approximation of the latter is computed (middle) and used as reference in estimating the correspondence by optical flow.

4. Learning Algorithm

Algorithm: Iterative Learning

Input: $\{\mathcal{M}_i | i = 1 \dots m\}$

Output: $\mathcal{M}(\alpha, \beta)$,$\sigma^2$

```
initialize M(α,β) to M_k and A_1 to ∅;
while A_j ≠ {1,...m} do
    foreach M_i : i ∉ A_j do
        if i ≠ 1 estimate (α̂_i, β̂_i, ρ̂_i) and Î_i;
        estimate v̂_i;
        warp: M_i → M'_i;
    end
    select subset A_{j+1} ⊃ A_j;
    {M'_i | i ∈ A_{j+1}} ⇒ M(α,β);
    estimate σ̂²;
end
```

We can now reformulate the registration problem of equation (2) so that the reference mesh $\mathcal{M}_1$ is actually a 3D morphable model. Substituting $\mathbf{I}_1$ with $\mathbf{M}(\alpha, \beta, \rho)$ and using as metric the reweighted L_2-norm defined by equation (3), we have for each example $\mathcal{M}_i$:

$$E_i(\mathbf{v}, \alpha, \beta, \rho) = \sum_{u,v} \|\mathbf{I}_i \circ \mathbf{v} - \mathbf{M}(\alpha, \beta, \rho)\|^2 \qquad (11)$$

To minimize the error defined in (11), we take an iterative approach, alternating between the estimation of the parameters (α, β, ρ) and of the flow $\mathbf{v}$; the whole algorithm can be subdivided in six distinct blocks:

Initialization (step 1). We choose one of the examples, say $\mathcal{M}_k$, and set the morphable model to it:

$$\mathcal{M}(\alpha, \beta) \equiv \mathcal{M}_k = (K_k, \mathbf{S}_k, \mathbf{T}_k)$$

The choice of the initial reference $\mathcal{M}_k$ is highly influential on the final result: since it provides the common domain of parameterization K, it should ideally provides all the features present in the other samples.

Model Fitting (step 4). The morphable model is fitted to the example $\mathcal{M}_i$ by maximizing the posterior probability

$$p(\alpha, \beta, \rho | \mathbf{I}_i) \propto p(\mathbf{I}_i | \alpha, \beta, \rho) P(\alpha) P(\beta) P(\rho)$$

Assuming for the parameter ρ a Gaussian distribution with mean $\bar{\rho}$ and variances $\sigma^2_{R,i}$, let

$$\begin{aligned} E_p(\alpha, \beta, \rho) &= -2 \log \left[P(\alpha) P(\beta) P(\rho) \right] \\ &= \|\alpha\|^2 + \|\beta\|^2 + \sum_i \frac{(\rho_i - \bar{\rho}_i)^2}{\sigma^2_{R,j}} \end{aligned}$$

be the cost function derived from the prior probabilities of the model parameters. Then, for each input example $\mathcal{M}_i$, we minimize

$$\begin{aligned} E &= -2 \log p(\alpha, \beta, \rho | \mathbf{I}_i) \\ &= \frac{1}{\sigma^2_R} E_i(\mathbf{v}, \alpha, \beta, \rho)|_{\mathbf{v}=0} + E_p(\alpha, \beta, \rho) \end{aligned}$$

by using a stochastic Newton descent (SND,[7]); note that the variance σ^2_R of the residuals is, although related, different from the regularization term σ^2, and that its value is empirically chosen. The approximation of $\mathcal{M}_i$ is regularized by reconstructing it via equation (6), and we denote its cylindrical projection by $\hat{\mathbf{I}}_i$. Note that at the first iteration, when no morphable model has been estimated, this step is skipped and all the $\hat{\mathbf{I}}_i$ are set to the cylindrical projection of the reference $\mathbf{I}_k$.

Optical Flow (step 5). For each input example $\mathcal{M}_i$, we minimize

$$E_i(\mathbf{v}) = \sum_{u,v} \|\mathbf{I}_i \circ \mathbf{v} - \hat{\mathbf{I}}_i\|^2$$

via optical flow, as described in sec.(2.2). However, the squared distance is not summed over the whole images, but rather only in the regions where $\mathbf{I}_i$ and $\hat{\mathbf{I}}_i$ overlap. This condition is required because of the high sensitivity of the optical flow algorithm to the boundaries. If the 2D projection $\mathbf{I}_i$ is missing some part, it will present a boundary which in the approximation $\hat{\mathbf{I}}_i$ is placed somewhere else (since the morphable model has the missing part); the optical flow, trying to make the boundaries match, will end up with a wrong estimation of the correspondence.

At the first iteration, we also compute an estimate $\hat{\rho}$ of the rigid parameters by compensating for the rigid components in the flow $\mathbf{v}$; this is not needed at the subsequent steps since the estimate given by SND is much more precise.

Warping (step 6). Each input example $\mathcal{M}_i$ is warped to $\mathcal{M}'_i$ as described at the end of section 2.1. However, special attention must be paid in the implementation to the fact that the estimated correspondence incorporates a rigid transformation, either as a rigid component in $\mathbf{v}$ or explicitly through the parameter ρ of $\mathbf{M}(\alpha, \beta, \rho)$. In order to model with $\mathcal{M}(\alpha, \beta)$ only the non-rigid deformations of the object class, we have to get rid of these rigid components. In case some part of $\hat{\mathbf{I}}_i$ was not included in the minimization of step (5), the corresponding vertices of the warped mesh $\mathcal{M}'_i$ will be marked as being missing.

Subset selection (step 7). At each iteration, we select a training subset A_{j+1}, with

$$A_j \subset A_{j+1} \subseteq \{1, \ldots m\}$$

The warped meshes $\mathcal{M}'_i$ with $i \in A_{j+1}$ will be used for building the model. Given the difficulty in automatically assessing the correctness of correspondence, this step is carried out by a human supervisor.

The supervisor evaluates the correspondence of an example by checking its *caricature* and *anti-face*. Caricatures are 3D meshes computed by adding to the position (or texture colour) of each vertex of the example its difference (or a multiple of it) from a given reference, tipically the average. In the case of anti-faces the difference is subtracted to the reference. In terms of morphable model, this is equivalent to multiply the (shape) coefficients α by a factor of either 2 (caricature) or -1 (anti-face). Correspondence errors show up as visible artefacts in caricatures (of shape and of texture too) and anti-faces.

Statistical Analysis (steps 8-9). The warped examples $\{\mathcal{M}'_i | i \in A_{j+1}\}$ are processed as described in sec. 3, in order to estimate the matrices $\mathbf{C}_s$ and $\mathbf{C}_t$ which defines the morphable model (see equation (10)). A new estimate of the noise covariance is also computed as in equation (9); note that during most of the iterations, the set of warped meshes $\mathcal{M}'_i$ is partitioned into a training set A_{j+1} used for building the model, and the complement with regard to the entire set of examples; this latter subset can be used as test set in estimating the noise. A final estimate of the noise, that is when all the examples are used for training, is obtained via 10-fold cross-validation.

5. Results

Figure 2 shows three examples of wrong correspondences that the optical flow may produce, because of the forced matching between boundaries artificially set. The three rendered images on the top row are shape caricatures of samples for which this problem occurred. On the bottom row the results of the learning algorithm are shown: as one can see, the artefacts of the above row have disappeared.

Figure 2. Three examples where artefacts given by previous methods (top row) are removed (bottom row). The images are rendered from shape caricatures (see sec. 4) of warped 3D meshes.

One should be warned, however, of the intrinsic limitations of statistical reconstruction when dealing with high-dimensional data. In this case, an *edge effect* can occur, by which all new data tends to be at the edge of the distribution of training data, causing the reconstruction to be in effect an extrapolation. When this happens, the maximum-likely reconstruction of the data can often be sub-optimal for visualization tasks, especially because of the lack of an external constraint on the continuity of the surface. In figure 3 we show a 3D model of a face bounded to a small region (compared to the typical data we use) and with holes; in this case more than half (53.8%) of the vertices of the common domain is missing. As shown in the figure, the statistical reconstruction would not fit well to the original data, The reconstruction of the approximation obtained by SND can be used instead, since it minimizes the distance from the original surface. The remaining discontinuities are removed by blending. We prefer for this kind of processing the term *completion* since it is no real statistical reconstruction.

The regularization term in the statistical model allows for a continuous modulation of the model complexity, rather than a discrete one based on discarding a certain number of principal components. Since the model complexity controls the generalization performance, adapting this complexity iteratively during the learning algorithm (where new examples are approximated), instead of selecting a subset of the principal components at the end of it, should result in a better model. The decrease in the model complexity can be assessed by comparing the effective degrees of freedom, as defined in equation 7, of a plain morphable model and of a regularized one obtained with the same data (200 examples of 3D face models). Estimating the model noise by 10-fold cross validation, the effective d.o.f. (in shape) of the regularized model has a value of 16.82, compared to a value of 21.10 for the non-regularized one (that is a decrease of about 20%); the smaller value is partially due to an increase of the noise estimate (from 628.517 a.u. to 685.211 a.u.), and partially to a greater compactness of the (warped) data distribution. The latter contribution can be seen by comparing the sum of the variances along the principal axes: for shape data the value of the sum is about 17.5% smaller for the regularized model.

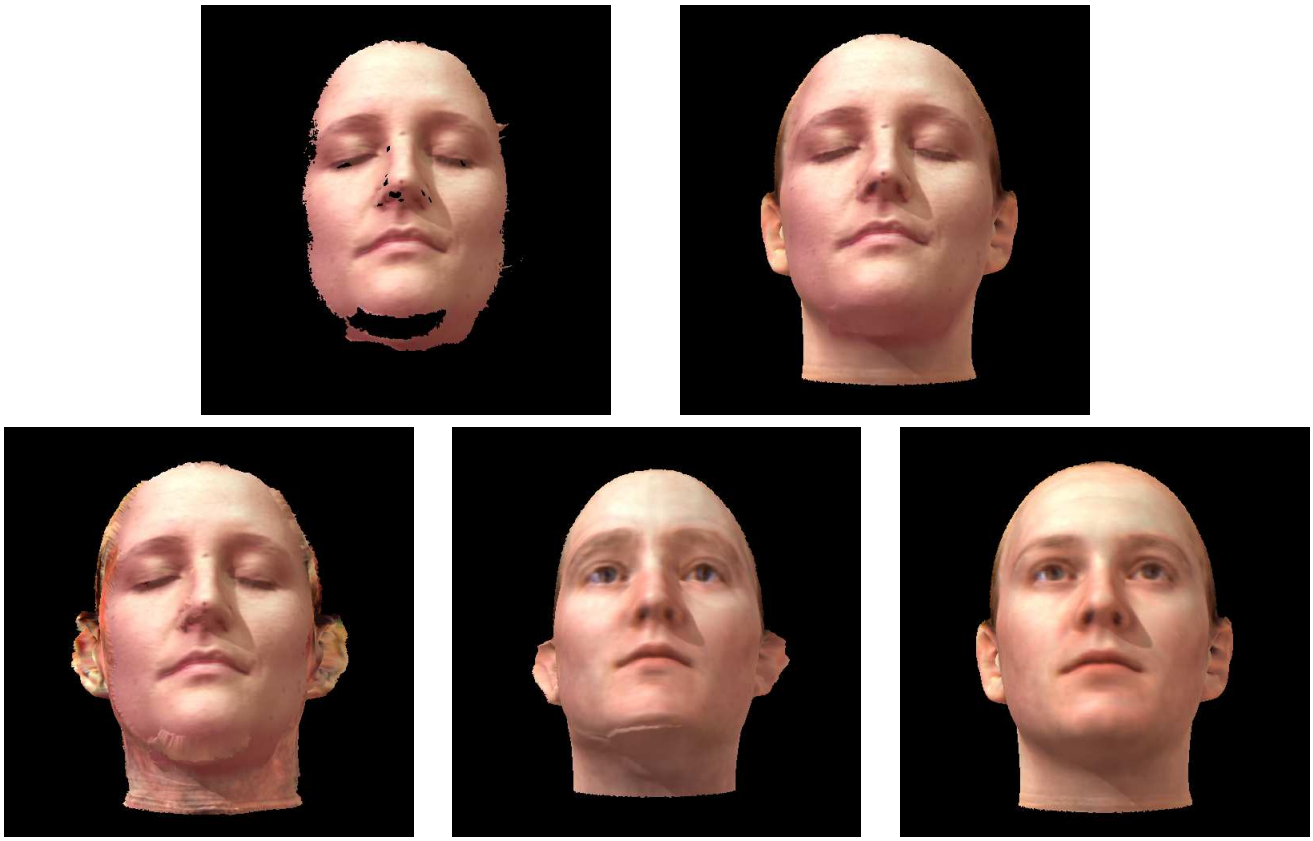

Figure 3. The missing data of a novel 3D mesh (top-left) can be statistically reconstructed (bottom-left). However, the results are influenced by the edge effect described in the text. By blending the approximation obtained via Stochastic Newton Descent (bottom-right) with the original data one can obtain a better-looking completion (top-right). The bottom-mid image shows the approximation obtained by SND without regularization.

6. Conclusions

In this paper we introduced the new concept of regularized 3D morphable models, by adding a regularization term to the statistical model previously used. The regularization term allows us to directly handle partial information during statistical analysis. Accordingly, we introduce a semi-automatic learning algorithm that estimates both 3D correspondence and the model concurrently. The non-automatic part of the algorithm consists in an evaluation of the correspondence results at that iteration.

The evaluation task is crucial since it avoids the propagation of correspondence errors to the morphable model, and through it to the approximations computed at the next iteration. Because of this, we decided to let the training subset be selected by the user rather than automatically. However, in future we will test the feasibility of detecting automatically the artefacts showing up in the caricatures, and removing consequently any need for user interaction.

The results that we presented show how the use of the regularization term can improve results both avoiding errors in 3D correspondence and estimating a "better" statistical model, in the sense that we reduce the variance in the estimation without loosing too much on the bias side.

References

[1] M. Alexa and W. Müller. The morphing space. In *Proceedings of WSCG '99*, 1999.

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

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

[4] V. Blanz, C. Basso, T. Vetter, and T. Poggio. Reanimating faces in images and video. In *Proc. of Eurographics 2003*, 2003.

[5] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *ACM Computer Graphics (Proc. of SIGGRAPH '99)*, volume 33, pages 187–194, 1999.

[6] V. Blanz and T. Vetter. Reconstructing the complete 3d shape of faces from partial information. *it - Information Technology*, 44(6):295–302, January 2002.

[7] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(9), September 2003.

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

[9] A. Gregory, A. State, M. C. Lin, D. Manocha, and M. A. Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. In *Proc. of Computer Animation '98*, pages 64–71, 1998.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *ACM Computer Graphics (Proc. of SIGGRAPH '93)*, volume 27, pages 19–26, 1993.

[11] J. R. Kent, W. E. Carlson, and R. E. Parent. Shape transformation for polyhedral objects. In *ACM Computer Graphics (Proc. of SIGGRAPH '92)*, volume 26, pages 47–54, 1992.

[12] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *ACM Computer Graphics (Proc. of SIGGRAPH '01)*, volume 35, 2001.

[13] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622, 1999.

[14] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *Proc. of CVPR'97*, pages 40–46, 1997.

[15] T. Vetter and T. Poggio. Image synthesis from a single example image. In *Computer Vision - ECCV'96*, 1996.

[16] G. Wolberg. Image morphing: a survey. *The Visual Computer*, 14:360–372, 1998.

[17] J. y. Noh and U. Neumann. Expression cloning. In *ACM Computer Graphics (Proc. of SIGGRAPH '01)*, volume 35, pages 277–288, 2001.