Fast Nonrigid Mesh Registration with a Data-Driven Deformation Prior

David C. Schneider and Peter Eisert
Fraunhofer Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany
{david.schneider,eisert}@hhi.fraunhofer.de

Abstract

We propose an algorithm for non-rigidly registering a 3D template mesh with a dense point cloud, using a morphable shape model to control the deformation of the template mesh. A cost function involving nonrigid shape as well as rigid pose is proposed. Registration is performed by minimizing a first-order approximation of the cost function in the Iterative Closest Points framework. We show how a complex shape model, consisting of multiple PCA models for individual regions of the template, can be seamlessly integrated in the parameter estimation scheme. An appropriate Tikhonov regularization is introduced to guarantee the smoothness of the full mesh despite the splitting into local models. The proposed algorithm is compared to a recent generic nonrigid registration scheme. We show that the data-driven approach is faster, as the linear systems to be solved in the iterations are significantly smaller when a model is available. Also, we show that simultaneous optimization of pose and shape yields better registration results than shape alone.

1. Introduction

The aim of a 3D registration algorithm is to align two meshes or point clouds, called the template and the target, as closely as possible. To this end, the template undergoes certain transformations while the target remains unchanged. The registration is called non-rigid if the transformations include changes in shape. The way in which the shape is allowed to change is controlled by what is sometimes called the stiffness- or deformation prior, as it is defined independent of the data. In this paper we suggest an efficient nonrigid registration algorithm where deformation is controlled by a morphable shape model learned from a database of meshes. While the algorithm relies on the availability of a model, it is computationally more efficient than approaches with generic stiffness constraints based on vertex neighborhoods.

From a morphable model point of view, the proposed method can also be seen as a direct model fitting algorithm for point cloud targets. This is in contrast to the typical approach to fitting a morphable model to a point cloud, which is a three step process: First, a generic, nonrigid registration scheme is employed to register a reference mesh with the point cloud. Then the point cloud is converted to the topology of the reference mesh, which is the same as the topology of the morphable model. Finally, the model is fitted, which is trivial when the target shares its topology. The method to be proposed allows to fit the model directly to the point cloud without preceding registration and conversion steps. Throughout the paper, we assume the target to be a dense point cloud, gained, for example, from laser-scanning, 3D surface reconstruction, etc. While the proposed method is not specific to a certain domain, the morphable model used to generate examples and experimental data is one of human head shape, which seems to be the predominant domain of morphable models. Note, however, that in contrast to many works on morphable models of the head or face, we are concerned with shape alone; texture information is neither used nor required.

Our algorithm is based on minimizing a cost function that involves both nonrigid shape and rigid pose, which will be shown to give better registration results than shape alone. The optimization is computed using the Iterative Closest Points (ICP) scheme with a first order approximation of the cost function. Therewith our methodology is closely related to that of many rigid registration algorithms. Special attention is paid to the integration of a complex morphable model, consisting of multiple sub-models for individual regions, in the cost function.

The paper is structured as follows. After an overview of related work in section 2, we briefly recapitulate PCA-based morphable shape models in section 3. Next, we introduce the cost function and its first-order approximation (section 4). In section 5, we show how a morphable model that consists of multiple sub-models can be integrated in the cost function and address regularization issues. In section 6, we state the iterative algorithm used to estimate the parameters. Finally, we compare our method to a recent generic non-rigid registration algorithm and give examples of results (section 7).

2. Related work

As the literature on nonrigid 3D registration is extensive, our review of related work concentrates on methods developed in the context of morphable (head) models. For an entirely different class of approaches, based on surface geometry, see, for example, Bronstein et al. [12] and their excellent monograph [11] which provides an extensive bibliography. Also, the discussion of methods developed for 2D image registration is beyond scope here. PCA shape models have been successfully used in this context, e. g. by Albrecht et al. [1].

Morphable models of human heads or faces have been applied to a large number of problems in computer vision and graphics. These include, for example, recognition [10], tracking [16] and several varieties of 3D reconstruction, e. g. from uncalibrated video [13] or from stereo images [3]. The most prominent model is that of Vetter and Blanz [9], comprising both geometry and texture. Registration of novel laser-scans is performed with a modified optical flow algorithm that exploits texture and geometry information. An algorithm for fitting the 3D model to a 2D photograph of a face is proposed, using analysis by synthesis techniques and stochastic optimization in the model's parameter space. Romdhani and Vetter [17] extended this fitting algorithm, exploiting various image features such as edges or specular highlights. An analysis-by-synthesis approach was also recently used by Blanz et al. [8] for fitting the model directly to a textured 3D scan, paying special attention to compensating the effects of unfavorable lighting.

Fewer algorithms deal with fitting problems from a purely geometric point of view, i. e. without using texture information. As described in the introduction, the problem of fitting a shape model to an unstructured (and untextured) point cloud is often approached in a three step process of model-independent registration, topology conversion and model fitting. Generic registration schemes of this kind can be found, for example, in Kähler et al. [15]. Their method is based on iteratively subdividing a coarse mesh and aligning the new vertices to the reference model. The initial model for the subdivision is obtained from manually placed landmarks. Allen et al. [2] register full human body scans by using a generic nonlinear optimizer on a three-term cost function. The function penalizes distances to the target as well as dissimilarities between transforms of adjacent points, thereby controlling the rigidity of the overall transformation. The registration is guided by a set of manually annotated landmarks. Landmarks are not required by Anguelov et al. [6] who formulate the registration problem in the framework of a Markov random field. A variety of local and relational cost measures are incorporated in the field and the registration is computed by loopy belief propagation. While computational complexity is not addressed, the size of the network and the complexity of the different measures suggest that the method is costly. In [5], the same authors fit a combined model of articulated pose and PCA-based deformation to laser-scans of the whole human figure. A specifically designed nonlinear optimization scheme is used where iteratively subsets of the parameters are kept fixed while others are optimized; also the adaptation rate of different parameters is controlled in order to avoid local minima. The algorithm can be used for "shape completion" by fitting it to partial scans. Shape completion is also possible with our approach and briefly addressed in section 6.2.

The ICP algorithm, originally introduced by Besl and McKay [7], is regarded as state of the art for rigidly registering two or more partially overlapping meshes. Several authors have extended ICP to nonrigid registration problems. Haehnel et al. [14], for example, combine ICP optimization with a coarse to fine approach in order to learn 3D models of non-stationary objects with a mobile robot. Amberg et al. [4] use the ICP scheme to minimize a cost function similar to that of [2] in a stepwise optimal fashion. While they include a term for manually placed landmarks, these are not mandatory for their approach. We discuss their algorithm in more detail in section 7, where we compare it to our method.

3. Morphable shape models

In this section we briefly recapitulate morphable shape models. To build such a model for a certain domain—human heads in our case—a database of meshes with a common, semantically consistent topology is required. Semantic consistency means that topologically equivalent vertices in different meshes are required to represent the same point on the head such as the tip of the nose, the center of the upper lip, etc.

Morphable shape models are based on Principle Component Analysis (PCA) of the geometry. However, if the shapes to be described by the model are complex, a single PCA can often represent only an insufficient amount of shape variability. Therefore, it is common to split the domain into different regions each of which is described by its own PCA (e. g. [9]). To avoid discontinuities at the borders, the regions are defined as overlapping and contributions of different regions to a single vertex location have to be blended. The regions used in this paper are depicted in figure 1 (left). The integration of these multi-PCA models in a single formal framework is discussed in section 5.

For each region, the model is computed as follows. The geometry of the region in each mesh in the database can be represented as a single vector of concatenated vertex coordinates, where N is the number of vertices in the region:

$$\mathbf{b}_i = (x_1\, y_1\, z_1 \ldots x_N\, y_N\, z_N)^T \tag{1}$$

The whole database, with respect to the currently regarded region, is described by a matrix $\mathbf{B} = [\mathbf{b}_1 \ldots \mathbf{b}_M]$ where M is the number of meshes in the database. Denoting by $\boldsymbol{\mu} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{b}_i$ the region's mean shape, and by $\hat{\mathbf{B}} = [\mathbf{b}_1 - \boldsymbol{\mu} \ldots \mathbf{b}_M - \boldsymbol{\mu}]$ the database centered on the mean, the region's PCA model is obtained by computing the spectral decomposition

$$\mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T = \hat{\mathbf{B}}^T\hat{\mathbf{B}}. \tag{2}$$

$\mathbf{P}$ is the orthonormal matrix of eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. Typically, columns of $\mathbf{P}$ corresponding to eigenvectors with small eigenvalues are omitted to obtain a more compact model that captures the most dominant variations in the data. We will refer to $\mathbf{P}$ as the region's *PCA model matrix* in the following.

4. Cost function

We approach model-based non-rigid registration as a problem of estimating a set of parameters that minimize a cost or error function. In this section, we assume that correspondences between vertices of the template—i. e. the morphable model—and points of the target point cloud are known. Template vertices are denoted as vectors $\mathbf{p}_i$ and corresponding target points as $\mathbf{q}_i$. The correspondence assumption will, of course, be dropped in the final algorithm.

As a template vertex $\mathbf{p}_i$ belongs to the morphable shape model, it can be written as

$$\mathbf{p}_i = \mathbf{M}_i\mathbf{m} + \boldsymbol{\mu}_i. \tag{3}$$

We can adopt two points of view on this equation. From a morphable model perspective, $\mathbf{M}_i$ is a part of the PCA model matrix that describes the region $\mathbf{p}_i$ belongs to; $\boldsymbol{\mu}_i$ is a part of that model's mean vector and $\mathbf{m}$ is a parameter vector that determines the region's shape. From a non-rigid registration point of view, $\boldsymbol{\mu}_i$ is a vertex of the template mesh, $\mathbf{M}_i$ is a linear deformation model and $\mathbf{m}$ is a parameter vector that describes the deformation. We will refer to $\mathbf{M}_i$ as the *shape matrix* of the vertex $\mathbf{p}_i$ in the following and suspend the discussion of its structure until section 5.

To obtain a good registration, not only shape but also pose and scale of the target must be estimated. Otherwise, deviances in pose and scale will be compensated by the shape model, resulting in higher fitting errors and unnecessarily deformed meshes. This claim will be underpinned experimentally in section 7. We describe scale by a factor s and pose by a translation vector $\mathbf{t}$ and a vector $\boldsymbol{\theta} = [\theta_x\ \theta_y\ \theta_z]^T$ of Euler angles.

For given correspondences, the cost function to be minimized is

$$e\left(s, \boldsymbol{\theta}, \mathbf{m}, \mathbf{t}\right) = \sum_i \left\| s\,\mathrm{rot}(\boldsymbol{\theta})\left(\mathbf{M}_i\mathbf{m} + \boldsymbol{\mu}_i\right) + \mathbf{t} - \mathbf{q}_i \right\|^2, \tag{4}$$

where $\mathrm{rot}(\boldsymbol{\theta})$ is the 3×3 rotation matrix corresponding with the Euler angles in the argument. There are closed form solutions for the problem of estimating rigid the transformation parameters $\boldsymbol{\theta}$ and $\mathbf{t}$, which is sometimes referred to as the Procrustes problem. These methods are, however, difficult to combine with the estimation of other parameters such as the shape / deformation parameter $\mathbf{m}$ in our case. Therefore, our strategy is to formulate a first-order approximation of the cost function and compute the solution iteratively. This approach is also prevalent in the literature on ICP for rigid matching of point clouds.

To linearize rotation, we use the well known approximation of $\mathrm{rot}(\boldsymbol{\theta})$ for small angles in the argument, which can be derived from a Taylor series expansion of a rotation matrix:

$$\begin{aligned} \mathrm{rot}(\boldsymbol{\theta}) &\approx \mathbf{I}_3 + \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \\ &:= \mathbf{I}_3 + \Delta\boldsymbol{\Theta} \end{aligned} \tag{5}$$

$\mathbf{I}_3$ denotes the 3×3 identity matrix. To solve for the angles, they can be swapped with the coordinates of a rotated point $\mathbf{p} = [x\ y\ z]^T$:

$$\begin{aligned} \mathrm{rot}(\boldsymbol{\theta})\mathbf{p} &\approx \mathbf{p} + \Delta\boldsymbol{\Theta}\mathbf{p} \\ &= \mathbf{p} + \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} \qquad (6) \\ &:= \mathbf{p} + [\mathbf{p}]_\times \boldsymbol{\theta} \qquad (7) \end{aligned}$$

The notation $[\mathbf{p}]_\times$ is motivated by the fact that the matrix in equation (6) is the matrix form of the cross product of $\mathbf{p}$ with an arbitrary vector. Equation (7) can also be derived directly from the Jacobian of a quaternion rotation [18].

Now the first order approximation of the error function can be derived as follows. Assume a given pose and shape / deformation $(s,\ \boldsymbol{\theta},\ \mathbf{m},\ \mathbf{t})$ changes by an unknown amount $(\Delta s,\ \Delta\boldsymbol{\theta},\ \Delta\mathbf{m},\ \Delta\mathbf{t})$. The new error is

$$\begin{aligned} e = \sum_i \| &(1 + \Delta s)\, s \cdot \mathrm{rot}(\Delta\boldsymbol{\theta})\mathrm{rot}(\boldsymbol{\theta}) \cdot \\ &\left(\mathbf{M}_i\left(\mathbf{m} + \Delta\mathbf{m}\right) + \boldsymbol{\mu}_i\right) + \mathbf{t} + \Delta\mathbf{t} - \mathbf{q}_i \|^2 \end{aligned} \tag{8}$$

Assuming the change in rotation $\Delta\boldsymbol{\theta}$ is small, we can use equation (5) to substitute

$$\mathrm{rot}(\Delta\boldsymbol{\theta})\mathrm{rot}(\boldsymbol{\theta}) \approx \left(\mathbf{I}_3 + \Delta\boldsymbol{\Theta}\right)\mathrm{rot}(\boldsymbol{\theta}).$$

Assuming further the changes of the other parameters are small as well, computing a Taylor expansion of equation (8) and omitting higher order terms yields

$$e = \sum_i \|\tilde{\mathbf{p}}_i + \mathbf{t} + \tilde{\mathbf{p}}_i \, \Delta s + \Delta\boldsymbol{\Theta} \, \tilde{\mathbf{p}}_i \\ + s \, \mathrm{rot}(\boldsymbol{\theta}) \mathbf{M}_i \, \Delta\mathbf{m} + \Delta\mathbf{t} - \mathbf{q}_i\|^2 \quad (9)$$

with

$$\tilde{\mathbf{p}}_i = s \, \mathrm{rot}(\boldsymbol{\theta}) \left(\mathbf{M}_i \mathbf{m} + \boldsymbol{\mu}_i \right). \quad (10)$$

Note that $\tilde{\mathbf{p}}_i + \mathbf{t}$ is the registration prior to the change of parameters.

Applying equation (6) and rearranging, we get the following linear system which can be solved for the parameters minimizing the cost function:

$$\begin{bmatrix} \tilde{\mathbf{p}}_1 & [\tilde{\mathbf{p}}_1]_\times & s \, \mathrm{rot}(\boldsymbol{\theta}) \mathbf{M}_1 & \mathbf{I}_3 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{p}}_N & [\tilde{\mathbf{p}}_N]_\times & s \, \mathrm{rot}(\boldsymbol{\theta}) \mathbf{M}_N & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \boldsymbol{\theta} \\ \Delta \mathbf{m} \\ \Delta \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 - \tilde{\mathbf{p}}_1 - \mathbf{t} \\ \vdots \\ \mathbf{q}_N - \tilde{\mathbf{p}}_N - \mathbf{t} \end{bmatrix} \quad (11)$$

Here, N is total number of vertices in the template mesh / morphable model.

5. Multi-PCA models

How does the linear system of equation (11) relate to a PCA-based morphable shape model with multiple overlapping regions, each described by its own PCA?

Clearly, by "stacking" the per-vertex mean vectors $\boldsymbol{\mu}_i$, we obtain $\left[\boldsymbol{\mu}_1^T \dots \boldsymbol{\mu}_N^T\right]^T$ which is simply the mean vector of the meshes. The relation of the per-vertex shape matrices $\mathbf{M}_i$ to the regions' PCA model matrices is more subtle and can be established as follows.

Assume that the model domain (here: the head) is split up into K different regions and denote the PCA model matrix of the j-th region by $\mathbf{P}^{(j)}$. Further denote the allocation of vertices to regions by index sets $\mathcal{R}_1 \dots \mathcal{R}_K$ with $i \in \mathcal{R}_j$ if a template vertex $\mathbf{p}_i$ belongs to region $\mathcal{R}_j$. Also note that $\mathcal{R}_j \cap \mathcal{R}_k \neq \emptyset$ if the two regions are adjacent due to overlap (see section 3).

The location of a vertex $\mathbf{p}_i$ belonging to region $\mathcal{R}_j$ is determined by $\boldsymbol{\mu}_i$ and three consecutive rows of the region's PCA model matrix $\mathbf{P}^{(j)}$. Denote these three matrix rows of $\mathbf{P}^{(j)}$ by $\mathbf{P}_i^{(j)}$ and let, by definition, be

$$\mathbf{P}_i^{(j)} = \mathbf{0} \qquad \text{if} \qquad i \notin \mathcal{R}_j. \quad (12)$$

Then the shape matrix $\mathbf{M}_i$ of each vertex in equation (11) can be defined as

$$\mathbf{M}_i = c_i^{-1} \left[\mathbf{P}_i^{(1)} \quad \dots \quad \mathbf{P}_i^{(K)} \right] \quad (13)$$

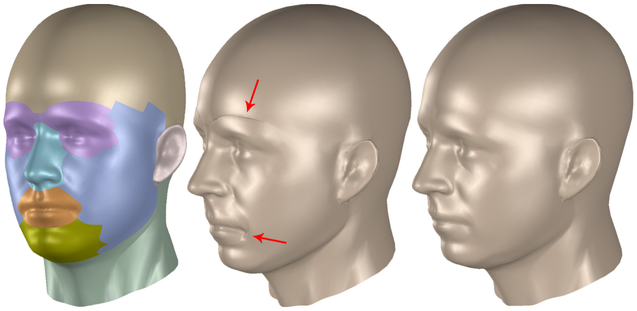

Figure 1. Left: Subdivision of the head into eight regions, each described by its own PCA model. Center: Discontinuity at the border between two regions. Right: The smoothness prior leads to an undistorted mesh ($\lambda_1 = 10$).

where c_i is the number of regions to which $\mathbf{p}_i$ belongs. The blending factor c_i^{-1} weights down the contribution of different region models if the vertex belongs to several regions. Otherwise, an area where n different regions overlap would be scaled by factor n. With the given definition of c_i, all regions a vertex belongs to contribute the same to its location. Other weighting schemes, preferring some regions over others for some vertices, are possible.

5.1. Regularization

In this section we address two problems—smoothness at region borders and total deformation strength—that can be met by adding regularization terms to the linear system of equation (11). Denoting the respective Tikhonov matrices by $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$, the matrix in the equation becomes:

$$\left[\begin{array}{ccc|c} \tilde{\mathbf{p}}_1 & [\tilde{\mathbf{p}}_1]_\times & \mathbf{I}_3 & s \, \mathrm{rot}(\boldsymbol{\theta}) \mathbf{M}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{p}}_N & [\tilde{\mathbf{p}}_N]_\times & \mathbf{I}_3 & s \, \mathrm{rot}(\boldsymbol{\theta}) \mathbf{M}_N \\ \hline & & & \lambda_1 \boldsymbol{\Gamma} \\ & \mathbf{0} & & \lambda_2 \boldsymbol{\Lambda} \end{array} \right] =: \mathbf{K} \quad (14)$$

Note that the translation and the shape term have been swapped to simplify notation. The factors λ_1, λ_2 control the influence of the regularizers.

Smoothness prior $\boldsymbol{\Gamma}$ Despite of the overlap at the borders of the regions, visible discontinuities can appear in the overall mesh: In the fitting process, the mean squared fitting error of each region is minimized. Therefore, for a vertex that belongs to several regions, each region model may yield a different location. The blending factor c_i^{-1} in equation (13) can hide this effect only if the deviation between different region models is small. Otherwise, discontinuities become visible, as shown in figure 1 (center).

The problem can be met by introducing a "smoothness prior" $\boldsymbol{\Gamma}$ in the linear system of equation (11). This prior forces different region models to yield the same locations for shared vertices and is defined as follows. Be $\mathbf{p}_i$ a vertex that belongs to exactly two regions, $\mathcal{R}_j$ and $\mathcal{R}_k$. Then the shape matrix $\mathbf{M}_i$ of $\mathbf{p}_i$ has exactly two blocks that are nonzero, namely $\mathbf{P}_i^{(j)}$ and $\mathbf{P}_i^{(k)}$. Hence, the region models of $\mathcal{R}_j$ and $\mathcal{R}_k$ yield the same location for $\mathbf{p}_i$ if

$$\left[\dots \mathbf{P}_i^{(j)} \cdots - \mathbf{P}_i^{(k)} \dots\right] \mathbf{m} =: \mathbf{\Gamma}_i \mathbf{m} = \mathbf{0}. \tag{15}$$

For a vertex belonging to more than two regions, each unordered pair of regions the vertex belongs to yields a system like the one in equation (15). In practice, there are very few vertices of this kind and they can be ignored without harm. When added to the linear system of equation (11), the smoothness prior should be scaled by s. Rotation and translation can be omitted as they do not affect the Euclidean distances minimized by the prior. Finally, $\mathbf{\Gamma}$ is defined as

$$\mathbf{\Gamma} = s \begin{bmatrix} \mathbf{\Gamma}_1 \\ \vdots \\ \mathbf{\Gamma}_L \end{bmatrix} \tag{16}$$

where L is the number of vertices belonging to two regions. Figure 1 (right) shows the effect of the smoothness prior.

Deformation strength $\mathbf{\Lambda}$ So far, the shape model controls the "direction" of the deformation and the smoothness prior controls the interaction of the individual PCA models at the regions borders. However, the actual strength of the deformation is still unregulated. This can be met by adding a second regularization term, acting, like the smoothness prior, on the deformation parameters $\mathbf{m}$. Obviously, the strength of the deformation is directly correlated with the norm $\|\mathbf{m}\|$ as each element of $\mathbf{m}$ controls one "direction" of deformation. However, not all of them are equally important: The directions are the orthogonal dimensions of the PCA models, and dimensions associated with a larger eigenvalue represent a larger variance of shape in the mesh database than those with a smaller eigenvalue. To control deformation strength in accordance with the model, the elements of $\mathbf{m}$ must be weighted differently. Therefore, denote by $\mathbf{\Lambda}^{(j)}$ the eigenvalue matrix of the j-th region's PCA model (see section 3) and define the deformation strength prior as the diagonal matrix

$$\mathbf{\Lambda} = \operatorname{diag}\left(\sqrt{\mathbf{\Lambda}^{(1)}}, \dots, \sqrt{\mathbf{\Lambda}^{(K)}}\right)^{-1}. \tag{17}$$

6. Iterative estimation

In this section we state the iterative nonrigid registration algorithm that registers the template mesh—the mean mesh of the morphable shape model—with a given point cloud $\mathcal{C} \subset \mathbb{R}^3$. Or, equivalently, we state the algorithm that fits the morphable shape model to a dense point cloud. The algorithm establishes correspondences $\mathbf{q}_i \in \mathcal{C}$ for all vertices $\mathbf{p}_i$ of the template mesh / shape model and determines the scale, pose and shape / deformation parameters $s, \boldsymbol{\theta}, \mathbf{m}, \mathbf{t}$ that minimize the cost function defined in equation (4). Note that the reached cost minimum may be a local one, as is typically the case with ICP based registration schemes. Therefore, the point cloud must be roughly aligned with the template before optimization. In practice, an alignment of the principal axes is sufficient.

6.1. Algorithm

The following quantities are maintained by the algorithm and initialized as follows:

- The accumulated scale, rotation, shape and translation, $s \leftarrow 1$, $\operatorname{rot}(\boldsymbol{\theta}) \leftarrow \mathbf{I}_3$, $\mathbf{m} \leftarrow \mathbf{0}$, $\mathbf{t} \leftarrow \mathbf{0}$. Note that accumulated rotation is maintained as a matrix rather than a vector of Euler angles in order to avoid problems such as gimbal lock. We denote this matrix by $\operatorname{rot}(\boldsymbol{\theta})$ for reasons of consistency, although there is no vector $\boldsymbol{\theta}$ maintained by the algorithm.

- The current registration of the template is initialized by the undeformed mean of the database meshes: $\tilde{\mathbf{p}}_i \leftarrow \boldsymbol{\mu}_i$ for $i = 1 \dots N$. In order to avoid repeating equations, $\tilde{\mathbf{p}}_i$ is to be understood as the current registration "minus translation", as in equation (10).

Step 1 In the main loop, the first step is to establish preliminary point correspondences: For each point $\tilde{\mathbf{p}}_i + \mathbf{t}$ of the current registration, the point $\mathbf{q}_i$ of the target point cloud $\mathcal{C}$ which is closest to $\tilde{\mathbf{p}}_i + \mathbf{t}$ with respect to the Euclidean distance is searched. This can be computed efficiently with an appropriate data structure, e. g. a KD-tree.

Step 2 Next the parameter changes are estimated. The matrix $\mathbf{K}$ of equation (14) is set up and the parameter changes are computed, using the SVD of $\mathbf{K}$:

$$\begin{bmatrix} \Delta s \\ \Delta \boldsymbol{\theta} \\ \Delta \mathbf{m} \\ \Delta \mathbf{t} \end{bmatrix} = \mathbf{K}^{+} \left[\begin{array}{c} \mathbf{q}_1 - \tilde{\mathbf{p}}_1 - \mathbf{t} \\ \vdots \\ \mathbf{q}_N - \tilde{\mathbf{p}}_N - \mathbf{t} \\ \hline \mathbf{0} \end{array}\right] \tag{18}$$

Step 3 Next, the accumulated parameters are updated:

$$\begin{aligned} s &\leftarrow (1 + \Delta s) \cdot s & (19) \\ \operatorname{rot}(\boldsymbol{\theta}) &\leftarrow \operatorname{rot}(\Delta \boldsymbol{\theta}) \cdot \operatorname{rot}(\boldsymbol{\theta}) & (20) \\ \mathbf{t} &\leftarrow \Delta \mathbf{t} + \mathbf{t} & (21) \\ \mathbf{m} &\leftarrow \Delta \mathbf{m} + \mathbf{m} & (22) \end{aligned}$$

Step 4 Finally, a new registration is computed based on the current parameters:

$$\tilde{\mathbf{p}}_i := s \operatorname{rot}(\boldsymbol{\theta}) \left(\mathbf{M}_i \mathbf{m} + \boldsymbol{\mu}_i\right) \quad \text{for} \quad i = 1 \dots N \tag{23}$$

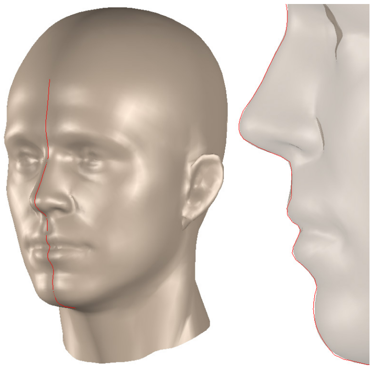

Figure 2. Shape completion: Here, the fitting target is only a profile curve (red line). A plausible head is found that closely approximates the target.

The process starts over with step one, until the change of the mean squared error (MSE), $e = \frac{1}{N}\sum_{i=1}^{N} \|\tilde{\mathbf{p}}_i + \mathbf{t} - \mathbf{q}_i\|^2$, computed with respect to the current correspondences, falls below a threshold.

6.2. Outlier rejection and shape completion

Often, the laser-scan point clouds have holes due to scanning errors or do not show the entire head. As can be seen in figure 3, our scans, for example, lack the top of the head as well as some parts of the lower throat. These areas of missing geometry have a bad effect on the optimization: Template vertices in those areas are matched with far away points of the cloud which leads to large squared errors with an overly strong influence on parameter estimation. To cope with this, a simple outlier rejection is used. A threshold is imposed on the distance of "closest points" in step 1 of the algorithm. The equations corresponding with points for which the threshold is exceeded can be omitted from the linear system when the parameters are estimated.

The proposed algorithm can also be used for shape completion—i. e. for estimating a plausible complete shape that fits an arbitrarily small piece of given geometry. Technically, shape completion is the same as outlier rejection in the proposed method: Equations referring to vertices that are missing in the target are omitted in step 2 of the algorithm. An extreme example of shape completion is shown in figure 2, where only the profile curve of the face is given as the target. Note that in such extreme cases the computation of the parameters relies mostly on the smoothness prior $\mathbf{\Gamma}$: If there are regions for which there is no given geometry, entire columns in the matrix of equation (11) contain only zero entries; this follows from the definition of the vertices' shape matrices in equation (13). If this is the case, the shape parameters of the affected regions are underdetermined in equation (11) and only the regularized system of equation (14) is solvable.

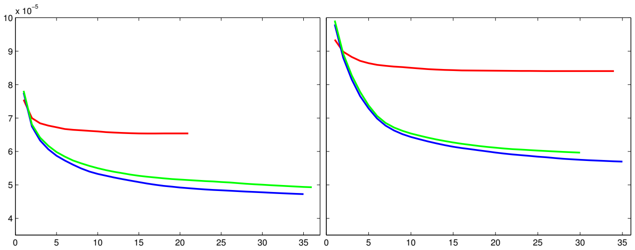

Figure 4. Typical progressions of MSE to model during the fitting process. Red: Generic nonrigid fitting algorithm. Green: Proposed method without rigid pose. Blue: Proposed method with rigid pose. For all trials, the template mesh was rigidly aligned with the point cloud prior to registration.

7. Evaluation

Figure 3 shows results of the fitting algorithm for several male heads with different facial characteristics. The shape model used is based on a database of 180 laser-scans of male adults. Meshes based on the target scans were omitted from the database for the evaluation experiments. Regarding scale, the template / mean mesh has a height of one. The point clouds were roughly pre-aligned by matching scale and principal axes. The template mesh has 6300 vertices. The eight regions depicted in figure 1 (left) were used and each region is described by 15 eigenvectors in the model. The regularization weights were established experimentally and are $\lambda_1 = 20$ and $\lambda_2 = 0$; i. e. a restriction of deformation strength was not necessary.

By omitting the parts of the fitting equation related to pose and scale, the proposed algorithm can be used to estimate shape parameters only. This leads to the question whether simultaneous optimization of pose and shape yields better results than a plain pose registration followed by a plain shape estimation. Hence we rigidly aligned point clouds with the template / mean mesh using regular ICP, and registered the model with the point clouds twice, once with and once without pose optimization. Example plots of MSE over the iteration for several trials are shown in figure 4. Despite of the preceding pose optimization, the final MSE of simultaneous pose and shape optimization was on average five percent lower than that of plain shape fitting (30 trials).

7.1. Comparison with generic fitting

Regarding comparison with other methods, we are primarily interested in the performance of our data-driven approach in comparison with recent generic non-rigid registration schemes. We chose the algorithm of Amberg et al. [4], which is an extension of Allen et al. [2], as a benchmark for two reasons. Firstly, like our algorithm, it does not require manually annotated landmarks (although it can incorporate them if available). Secondly, it is structurally similar to the proposed approach: It is based on minimizing a cost func-

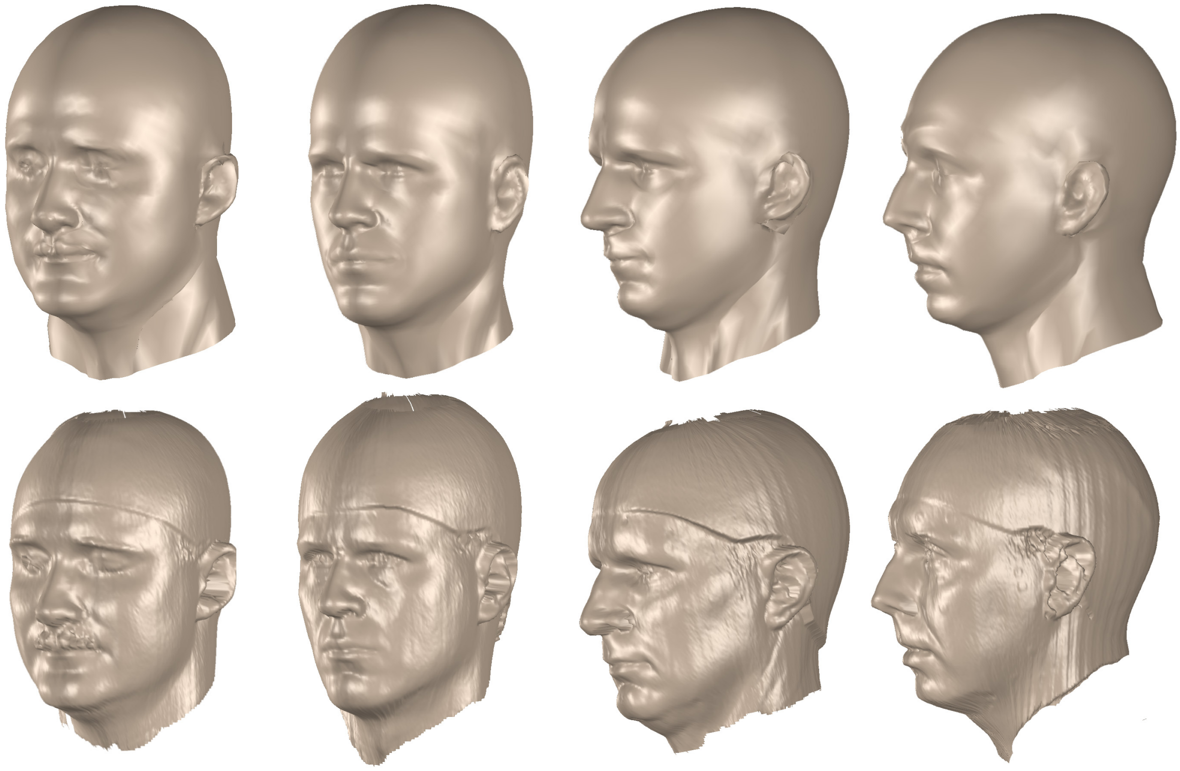

Figure 3. Results of the fitting algorithm. Laser scans in the bottom row, fitted model in the top row.

tion in the ICP framework with linear parameter updates during the iteration.

The primary difference between the investigated generic algorithm and the data-driven approach lies in the cost function. The generic algorithm estimates a full affine transform (i. e. 12 parameters) for every vertex of the template mesh. To prevent a free deformation of the template, a regularization term enforces similarity between transforms of vertices connected by an edge. This is achieved by minimizing the difference between transformation matrices of adjacent vertices with respect to the Frobenius norm. The strength of the regularization is determined by a stiffness factor and the entire ICP optimization is repeated several times with increasingly lower stiffness. Note that in contrast to the model-based approach, the generic algorithm does not allow to separate shape and pose parameters.

Regarding computation time, the data-driven algorithm outperforms the generic one by a factor of 17 on average (30 trials). This is not surprising given the linear systems the two algorithms solve in each iteration: Denoting by K the number of regions used and by P_i the number of principal components of the i-th region PCA model, the number of parameters to be estimated by the data-driven algorithm is $7 + \sum_{i=1}^{K} P_i$. Denoting by N the number of vertices in the template / model and by L the number of region-shared vertices used for regularization, the linear system comprises $3N + 3L + \sum_{i=1}^{K} P_i$ equations.

In contrast, the generic algorithm estimates $12N$ parameters. There are $4N$ equations governing the deformation due to the use of homogeneous coordinates. Computational cost, however, is dominated by the number of equations in the regularization term, which is four times the number of edges in the mesh. If the template is a triangle mesh with a Euler characteristic of two, the edge count is roughly three times the number of vertices which sums up to $16N$ equations in total.

This is reflected in the average computation times for solving a single linear system in the parameter estimation step (Matlab algorithms, 4 core Xeon, 3 GHz, 4GB RAM):

	sparse solver	non-sparse solver
model-based	2.7 s	1.0 s
generic	47.8 s	not enough memory

In summary, the model-based approach profits from fixed relations between vertices that are encoded in the model: The ways in which a vertex may move is determined by the data used to build the model and not by its neighbors. Therefore, the vertices can be treated as independent during optimization. In contrast, vertex relations are dynamic in the generic approach and hence must be included in the parameter estimation.

Finding an informative measure for comparing the algorithms with respect to registration quality is difficult. An obvious choice would be the remaining MSE after registration. This makes sense for the data-driven algorithm, as the space of possible deformations is limited by the region PCA models. However, for the generic algorithm, the MSE can trivially be brought to zero by setting the stiffness factor to zero at some point of the optimization; this allows the template's vertices to move freely to the locations of their current correspondences. The MSE does not convey any information about the "quality" of a registration, which may involve factors such as semantic consistency with the template mesh, distortion of triangles, etc.

Therefore, we suggest another error measure for the generic algorithm: We use the morphable model's mean mesh as a template and register it with the point cloud with the generic method. Then the model is fitted to the registered template, which is now trivial, as correspondences have been established by the registration algorithm. The remaining MSE of this fit is taken as the error measure. Thereby, the morphable model becomes the judge for the registration quality—registration counts as good if it is well reproduced by the morphable model. Note that from a morphable model point of view, this amounts to a comparison between direct fitting as proposed in this paper and indirect as described in the introduction. By this measure, the final MSE of the model-based method is, on average, 30 percent lower than the final MSE of the generic algorithm.

8. Conclusion

The contributions of this paper can be summarized as follows: We used the ICP scheme to non-rigidly fit a template mesh to an unstructured point cloud, guiding the deformation by a morphable shape model. The fitting was described as a minimization of a cost function by iterative estimation of nonrigid shape and rigid pose. Including rigid pose proved to yield better registration results than shape alone. We showed how a complex shape model, consisting of multiple PCA models for individual regions, can be integrated in the linear parameter estimation scheme. To avoid discontinuities at the region borders, we introduced a smoothness prior which is integrated in the fitting equations as a Tikhonov regularization. The proposed algorithm yields better results than a recent generic nonrigid registration scheme when the model is used to judge registration quality, which is difficult to measure. Solving smaller linear systems in each iteration, our method was shown to be significantly faster.

References

[1] T. Albrecht, M. Lüthi, and T. Vetter. A Statistical Deformation Prior for Non-Rigid Image and Shape Registration. In *Proc. Conference on Computer Vision and Pattern Recognition CVPR'08*, 2008.

[2] B. Allen, B. Curless, and Z. Popovic. The Space Of Human Body Shapes: Reconstruction And Parameterization From Range Scans. In *Proc. of ACM SIGGRAPH*, 2003.

[3] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter. Reconstructing High Quality Face-Surfaces using Model Based Stereo. In *Proceedings ICCV*, 2007.

[4] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Nonrigid ICP Algorithms For Surface Registration. In *Proc. Conference on Computer Vision and Pattern Recognition CVPR'07*, 2007.

[5] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis. SCAPE: Shape Completion and Animation of People. In *Proc. of ACM SIGGRAPH*, 2005.

[6] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis. The Correlated Correspondence Algorithm for Unsupervised Registration of Nonrigid Surfaces. *NIPS*, 17:33–40, 2005.

[7] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[8] V. Blanz, K. Scherbaum, and H.-P. Seidel. Fitting a Morphable Model to 3D Scans of Faces. In *Proceedings of ICCV*, 2007.

[9] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Proc. of ACM SIGGRAPH*, 1999.

[10] V. Blanz and T. Vetter. Face Recognition Based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25/9, 2003.

[11] A. M. Bronstein and M. M. B. an Ron Kimmel. *Numerical Geometry of Nonrigid Shapes*. Springer, 2008.

[12] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Topology-invariant similarity of nonrigid shapes. *International Journal of Computer Vision*, 81(3):281–301, 2009.

[13] P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38:153 – 171, 2000.

[14] D. Haehnel, S. Thruny, and W. Burgard. An Extension of the ICP Algorithm for Modeling Nonrigid Objects with Mobile Robots. In *Proc. of the International Joint Conference on Artificial Intelligence*, 2003.

[15] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head Shop: Generating Animated Head Models With Anatomical Structure. In *Proc. of ACM SIGGRAPH*, 2002.

[16] J. Paterson and A. Fitzgibbon. 3D Head Tracking Using Non-Linear Optimization. In *British Machine Vision Conference 03*, 2003.

[17] S. Romdhani and T. Vetter. Estimating 3D Shape and Texture Using Pixel Intensity, Edges, Specular Highlights, Texture Constraints and a Prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[18] M. Wheeler and K. Ikeuchi. Iterative estimation of rotation and translation using the quaternion. Technical Report CMU-CS-95-215, Computer Science Department, Carnegie Mellon University, 1995.