Fitting a Morphable Model to Pose and Shape of a Point Cloud

David C. Schneider, Peter Eisert

Fraunhofer Heinrich Hertz Institute, Einsteinufer 37, 10587 Berlin, Germany
{david.schneider,eisert}@hhi.fraunhofer.de

Abstract

The paper addresses the task of fitting a morphable head model to a dense, unstructured and untextured point cloud. The problem is typically approached in a multi-step process, comprising of generic nonrigid registration, conversion to the model's topology and fitting of the model. Here, a direct approach is proposed where the morphable model is fitted to the point cloud itself in an optimization process following the Iterative Closest Points framework. Along with shape parameters, rigid pose and scale are estimated explicitly which leads to better fitting results than shape estimation alone. A compact formulation of a cost function is proposed, applicable also in the case of a model comprising of multiple, independent sub-models each of which describes a region of the face. The use of the algorithm for other tasks such as adding depth to mugshot-style face photos is discussed.

1 Introduction

"Morphable models" are adaptive models of 3D shape (and sometimes texture) of a certain object domain. They are based on orthogonal subspace models of 3D geometry which are obtained from a database of meshes by Principal Component Analysis (PCA). As a single PCA model cannot cover sufficient variability for complex shapes, the geometry can be split up into overlapping regions each described by its own model. Thereby, the morphable model becomes a multi-PCA or PCA mixture model. Once the model is computed, the shape of an object belonging to the model's domain can be described by a small number of parameters. By manipulation of the parameters and back-projection from parameter into geometric space, the statistically most significant variations of the shape can be generated. As the parameter space is continuous, shapes can be morphed smoothly. Morphable models have been used most successfully in the domain of the human head and face, where they are called *morphable head models* [9]. A wide number of applications in computer vision and graphics can be driven or supported by morphable models, including, for example, face recognition [3, 10], face tracking [17, 18] and several varieties of 3D reconstruction, e.g. from uncalibrated video [12] or from stereo images [2].

In order to fit the morphable model to the shape of a human head—i.e. to compute the model parameters that describe the head—the target head must be represented in a way that is *semantically consistent* with the model: The target must have the same number of vertices as the model and these vertices must be placed at semantically equivalent locations. If, for example, the vertex with index K describes the tip of the nose in the model, this must also be the case in the representation of the target head. Note that semantic consistency is not the same as (and does not require) topological equivalence as topology does not relate to semantics and matters of triangulation are not important for a morphable model.

In practice this condition is rarely satisfied. Typically, digitizations of real human heads are given as dense point clouds generated by laser-scanners, multi-view reconstruction algorithms or other 3D acquisition techniques. Therefore, correspondences to the model's vertices must be found in the target point cloud before the model parameters can be determined. This problem is typically addressed by means of *nonrigid registration*. A nonrigid registration algorithm aligns two meshes—called the *target* and the *template*—as closely as possible by manipulating the pose of the template as well as its shape. Shape manipulation has to be restricted in some way in order to preserve the templates overall appearance. Typically, nonrigid registration algorithms mimic the behavior of an elastic material that stretches or contracts within limits in order to

adapt to a target. In summary, the work-flow of fitting a morphable head model to a dense point cloud mesh of a head is the following: (1) Fit a generic template head mesh that has the vertex structure of the morphable model to the point cloud by means of a generic nonrigid registration algorithm. (2) Find correspondences for the vertices of the nonrigidly registered template in the point cloud (e.g. by taking the cloud point closest to the normal of each template vertex). (3) Fit the morphable model to the topology-converted mesh.

In this paper we describe a method for performing these three steps in a single iterative optimization process. The algorithm directly fits a morphable head model to a point cloud without a generic nonrigid registration step, which is often computationally expensive. It is capable of explicitly separating the rigid part of the registration (i.e. the head pose) from the non-rigid part (i.e. the shape parameters) and it works with mixture models where different regions of the head are described by independent subspace models. The optimization is performed in the Iterative Closest Points framework and is thereby closely related to the state of the art in rigid registration methods.

The paper is structured as follows. After an overview of related work, we briefly recapitulate PCA-based morphable models of shape in section 3. In section 4, we introduce the cost function we minimize to compute the fit. We show how a PCA mixture model can be integrated in the cost function in section 4.1 and address regularization issues in section 4.2. Optimization within the ICP framework is addressed in section 4.3. Finally, we discuss results and sketch out the use of our algorithm for adding depth to mugshot-style face photos (section 5).

2 Related work

The most prominent morphable model is that of Vetter and Blanz [9] who coined the term "morphable head model". In contrast to the model used in this work, it comprises both geometry and texture. Registration of novel 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 [19] extend the fitting algorithm, exploiting various image features including edges and specular highlights. Fitting is performed by maximizing the posterior of the parameters given the features. Analysis by synthesis 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, that is without using texture information. For example, Lee et al. [15] fit a morphable model to a number of binary silhouette images obtained from calibrated cameras. The downhill simplex algorithm is used to compute the fit.

The problem of fitting a model to an unstructured (and untextured) point cloud is approached by most authors in a three step process: First, a generic nonrigid registration scheme is employed, which is not based on the model. A reference mesh that shares the model's topology is registered nonrigidly with the point cloud. The point cloud is then converted into that topology and finally the model is fitted to the converted mesh.

Several registration schemes of this kind have been published. For example, Kähler et al. [14] iteratively subdivide a rough face mesh and align the new vertices to the reference model. The initial model for the subdivision is obtained from manually placed landmarks. Allen et al. [1] register full human body scans by using a generic nonlinear optimizer on a three-term error function. The function penalizes distances of closest points and dissimilarities between transforms of nearby points, thereby controlling the rigidity of the overall transformation. The registration is guided by manually placed landmarks. Anguelov et al. [6] 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 they do not address computational complexity, the size of the network and the complexity of the different measures suggest that the method is quite costly. The same authors fit a model of articulated pose and shape to laser-scans of the human figure in [5]. They use a nonlinear optimization scheme specifically designed for the task.

The Iterative Closest Points algorithm, first introduced by Besl and McKay [7], is typically used to

compute a rigid registration of two or more partially overlapping meshes or point clouds, e.g. laser-scans of a larger scene. Extensions of the algorithm use different distance measures, for example point-to-plane distances in Chen and Medioni [11], to improve convergence. An overview of different ICP optimizations is given by Rusinkiewicz and Levoy [20]. More recently, Mitra et al. [16] introduced a general numeric optimization framework based on second order approximants which subsume previous approaches. Several non-rigid variants of ICP have been proposed; ICP also is the optimization framework used in our work. Haehnel et al. [13], for example, combine ICP optimization with a coarse to fine approach in order to compute 3D models of non-stationary objects with a mobile robot. Amberg et al. [4] use the ICP scheme to minimize an error function similar to that of Allen et al. [1] in a stepwise optimal fashion. While they include a term for manually placed landmarks, these are not necessarily required.

3 Morphable shape model

In this section we briefly recapitulate the formalism behind morphable head models. To build the model a database of head 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 models are orthogonal subspace models obtained by PCA. As the shape of the human head is complex and displays a large variability between individuals, a single PCA will not yield a sufficiently adaptive model. Therefore, we split the head into different regions each of which is described by its own subspace model. In order to avoid discontinuities at the region borders, the regions are defined as overlapping and contributions of different regions to a single vertex location have to be blended. See figure 1 for the regions used in this work. The integration of these multi-PCA models in a single formal framework will be discussed in section 4.1. Here, we concentrate on a single region.

The complete geometry of a region in each mesh of the database can be represented as a single vector by concatenating vertex coordinates :

$$\mathbf{b}_i = (x_1\ y_1\ z_1 \dots x_N\ y_N\ z_N)^T \qquad (1)$$

Figure 1: Subdivision of the head into regions, each described by its own PCA model (overlap is not shown).

The whole database, with respect to the currently regarded region, can then be descried by a matrix $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_M]$ where M is the number of meshes in the database. Denoting by

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{b}_i \qquad (2)$$

the region's mean shape, the database is centered on its mean,

$$\hat{\mathbf{B}} = [\mathbf{b}_1 - \boldsymbol{\mu} \dots \mathbf{b}_M - \boldsymbol{\mu}], \qquad (3)$$

and the region's subspace model is obtained by computing the spectral decomposition

$$\mathbf{M}\boldsymbol{\Lambda}\mathbf{M}^T = \hat{\mathbf{B}}^T\hat{\mathbf{B}}. \qquad (4)$$

Here, $\mathbf{M}$ is the orthonormal matrix of eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. Those columns of $\mathbf{M}$ that correspond to eigenvectors with small eigenvalues are omitted to obtain a compact model that still captures the most significant variations in the data.

4 Parameter estimation

To avoid unnecessary clutter in notation, we first look at a single pair of points $\mathbf{p}, \mathbf{q} \in \mathbb{R}^3$, where $\mathbf{p}$ is a vertex of the morphable model and $\mathbf{q}$ belongs to the target point cloud. We assume for the moment that it is known that $\mathbf{p}$ and $\mathbf{q}$ correspond and concentrate on the parameters of the transformations

relating $\mathbf{p}$ and $\mathbf{q}$. The correspondence assumption will be dropped in the final algorithm.

The location of $\mathbf{p}$ is determined by the PCA shape model. Hence it can be written as

$$\mathbf{p} = \mathbf{M}\mathbf{m} + \boldsymbol{\mu} \tag{5}$$

where $\mathbf{M}$ is a part of a PCA eigenvector matrix, $\boldsymbol{\mu}$ is a part of the mean shape vector and $\mathbf{m}$ are the model parameters. We will refer to $\mathbf{M}$ as the *shape matrix* of the vertex in the following. As we look at a single point pair for now, we suspend the discussion of size and structure of the shape matrix until section 4.1 and simply assume it has three rows.

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, increasing fitting errors and creating unnecessarily deformed meshes; see section 5 for an empirical result on this claim. In the following we parametrize pose by a vector $\boldsymbol{\theta} = [\theta_x\ \theta_y\ \theta_z]^T$ of Euler angles and a translation vector $\mathbf{t}$. The 3×3 rotation matrix corresponding with the angles $\boldsymbol{\theta}$ will be denoted as $\mathrm{rot}(\boldsymbol{\theta})$ and scale is described by a factor s. Finding both pose and shape for a fixed correspondence can the be formulated as minimizing the error

$$e\left(s, \boldsymbol{\theta}, \mathbf{m}, \mathbf{t}\right) = \left\| s\,\mathrm{rot}(\boldsymbol{\theta})\left(\mathbf{M}\mathbf{m} + \boldsymbol{\mu}\right) + \mathbf{t} - \mathbf{q}\right\|^2. \tag{6}$$

To compute a solution iteratively, we formulate a first-order approximation of the cost function. This approach is also prevalent in the literature on rigid point cloud matching. To linearize rotation, we use the approximation for small angles via the cross product matrix

$$[\mathbf{p}]_\times := \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}. \tag{7}$$

A small rotatioin of a point $\mathbf{p} = [x\ y\ z]^T$ can thus be approximated by

$$\mathrm{rot}(\boldsymbol{\theta})\mathbf{p} \approx \mathbf{p} + [\mathbf{p}]_\times\,\boldsymbol{\theta} \text{ for } \boldsymbol{\theta} \approx \mathbf{0}. \tag{8}$$

See Wheeler and Ikeuchi [22] for a derivation of this result from the Jacobian of a quaternion rotation.

Now assume a given registration described by $(s,\ \mathrm{rot}(\boldsymbol{\theta}),\ \mathbf{m},\ \mathbf{t})$ changes by an unknown amount $(\Delta s,\ \mathrm{rot}(\Delta\boldsymbol{\theta}),\ \Delta\mathbf{m},\ \Delta\mathbf{t})$. The new error is

$$e = \left\|(1+\Delta s)\,s\cdot\mathrm{rot}(\Delta\boldsymbol{\theta})\mathrm{rot}(\boldsymbol{\theta})\cdot \left(\mathbf{M}\left(\mathbf{m}+\Delta\mathbf{m}\right)+\boldsymbol{\mu}\right)+\mathbf{t}+\Delta\mathbf{t}-\mathbf{q}\right\|^2 \tag{9}$$

If we assume that all parameter changes are small, we can derive an expression linear in the parameter changes by expanding the above equation, omitting higher order terms from a Taylor series expansion and linearizing the rotation change $\mathrm{rot}(\Delta\boldsymbol{\theta})$ as described in equation (8):

$$\tilde{\mathbf{p}}\Delta s + [\tilde{\mathbf{p}}]_\times\,\Delta\boldsymbol{\theta} + s\,\mathrm{rot}(\boldsymbol{\theta})\mathbf{M}\Delta\mathbf{m} + \Delta\mathbf{t} = \mathbf{q} - \tilde{\mathbf{p}} - \mathbf{t}, \tag{10}$$

where

$$\tilde{\mathbf{p}} = s\,\mathrm{rot}(\boldsymbol{\theta})\left(\mathbf{M}\mathbf{m} + \boldsymbol{\mu}\right). \tag{11}$$

Note that $\tilde{\mathbf{p}}$ describes the situation prior to the change of the parameters except for the translation $\mathbf{t}$. Also note that $\mathrm{rot}(\boldsymbol{\theta})$ is known in equations (10) and (11) as it is part of the given pose and therefore need not be linearized.

Extending the above to N points is straightforward. Using matrix notation we can write,

$$\begin{bmatrix} \tilde{\mathbf{p}}_1 & [\tilde{\mathbf{p}}_1]_\times & s\,\mathrm{rot}(\boldsymbol{\theta})\mathbf{M}_1 & \mathbf{I}_3 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{p}}_N & [\tilde{\mathbf{p}}_N]_\times & s\,\mathrm{rot}(\boldsymbol{\theta})\mathbf{M}_N & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta\boldsymbol{\theta} \\ \Delta\mathbf{m} \\ \Delta\mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 - \tilde{\mathbf{p}}_1 - \mathbf{t} \\ \vdots \\ \mathbf{q}_N - \tilde{\mathbf{p}}_N - \mathbf{t} \end{bmatrix}. \tag{12}$$

Given a sufficient number of corresponding points, a least squares solution can be found.

4.1 Multi-PCA models

If the shape model consists of a single subspace, the matrices $\mathbf{M}_i$ in equation (12) are simply three-row blocks of the PCA eigenvector matrix, corresponding with the respective vertices. If a PCA mixture model is used, the construction of the $\mathbf{M}_i$ is more complex. Assume the number of regions and hence the number of subspaces used is K and denote the principal component matrix of each region's eigenvector matrix by $\mathbf{M}^{(j)},\ j = 1 \ldots K$. Further be $\mathcal{R}_1 \ldots \mathcal{R}_K$ sets of vertex indices such that $i \in \mathcal{R}_j$ if vertex $\mathbf{p}_i$ belongs to region $\mathcal{R}_j$. Note that $\mathcal{R}_j \cap \mathcal{R}_k \neq \emptyset$ if two regions are adjacent due to overlap at region borders. If a vertex $\mathbf{p}_i$ belongs to region $\mathcal{R}_j$ its location is determined by three consecutive rows of the region's eigenvector matrix $\mathbf{M}^{(j)}$ (and three consecutive elements of the

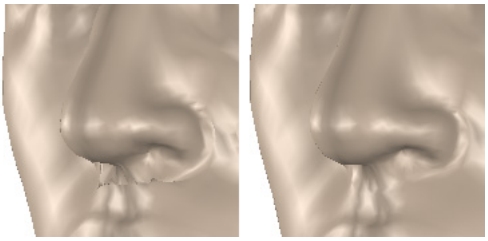

Figure 2: Left: Discontinuity at the border between two regions. Right: Regularization leads to an undistorted mesh.

region's mean vector). Denote these three matrix rows of $\mathbf{M}^{(j)}$ by $\mathbf{M}_i^{(j)}$ and define

$$\mathbf{M}_i^{(j)} = \mathbf{0} \qquad \text{if} \qquad V_i \notin \mathcal{R}_j. \tag{13}$$

Then the shape matrix $\mathbf{M}_i$ of a vertex in equation (12) is

$$\mathbf{M}_i = c_i^{-1} \begin{bmatrix} \mathbf{M}_i^{(1)} & \cdots & \mathbf{M}_i^{(K)} \end{bmatrix} \tag{14}$$

where c_i is the number of regions for which $i \in \mathcal{R}_j$. The role of the factor c_i^{-1} is to weigh down the contribution of different region models for vertices belonging to several regions. Without this downscaling, an area where different regions overlap would not match the size of the rest of the head.

4.2 Regularization for mesh smoothness

In the case of the PCA mixture model with overlapping regions, discontinuities can appear at region borders, the reason being the independent optimization of the parameters of each region subspace model. While this problem can be addressed by using large overlaps and heavy blending, a more principled solution is possible: Discontinuities can be eliminated by explicitly introducing smooth transitions between neighboring regions as an aim of the optimization. Consider a vertex $\mathbf{p}_i$ that belongs to two regions, $\mathcal{R}_j$ and $\mathcal{R}_k$. Clearly, the two region models yield the following locations $\mathbf{p}_i^{(j)}$ and $\mathbf{p}_i^{(k)}$ for the vertex:

$$\mathbf{p}_i^{(j)} = s\text{rot}(\boldsymbol{\theta}) \left(\mathbf{M}_i^{(j)} \mathbf{m}^{(j)} + \boldsymbol{\mu}_i \right) + \mathbf{t} \tag{15}$$

$$\mathbf{p}_i^{(k)} = s\text{rot}(\boldsymbol{\theta}) \left(\mathbf{M}_i^{(k)} \mathbf{m}^{(k)} + \boldsymbol{\mu}_i \right) + \mathbf{t} \tag{16}$$

The transition between the regions is smooth if $\mathbf{p}_i^{(j)} = \mathbf{p}_i^{(k)}$ which simplifies to

$$\mathbf{M}_i^{(j)} \mathbf{m}_j - \mathbf{M}_i^{(k)} \mathbf{m}_k = \mathbf{0}. \tag{17}$$

For each vertex belonging to two regions, the above equation can be derived. Together, the equations can be added to the linear system of equation (12) as a regularization term. For a more detailed account on our regularization see [21].

4.3 Iterative Closest Points solution

So far we assumed that correspondences between the template and target are known, which is not the case in practice. In order to establish correspondences and simultaneously estimate shape and pose parameters, we use the Iterative Closest Points optimization scheme: First, the parameters are initialized to $\mathbf{R} \leftarrow \mathbf{I}_{3\times3}$, $\mathbf{t} \leftarrow \mathbf{0}_{1\times3}$, $s \leftarrow 1$ and $\mathbf{m} \leftarrow \mathbf{0}$. Note that we maintain the overall rotation as a matrix $\mathbf{R}$ rather than a vector of angles in order to avoid the "gimbal lock" problem. During the iteration, the registration $\tilde{\mathbf{p}} = s\,\text{rot}(\boldsymbol{\theta})\,(\mathbf{M}\mathbf{m} + \boldsymbol{\mu})$ based on the current parameters computed. For each vertex, the closest point in the target point cloud is found using a KD-tree. Equation (12) is solved for the parameter changes by an SVD. The parameters are updated as follows:

$$\mathbf{R} \leftarrow \text{rot}(\boldsymbol{\theta}) \cdot \mathbf{R} \tag{18}$$

$$\mathbf{t} \leftarrow \Delta\mathbf{t} + \mathbf{t} \tag{19}$$

$$s \leftarrow (1 + \Delta s) \cdot s \tag{20}$$

$$\mathbf{m} \leftarrow \Delta\mathbf{m} + \mathbf{m} \tag{21}$$

Finally, the iteration starts over until the mean squared error $e = \frac{1}{N} \sum_{i=1}^{N} \|\tilde{\mathbf{p}}_i - \mathbf{q}_i\|^2$ does not change significantly any more.

As target point clouds are often incomplete—see, for example, our laser scans in figure 3—some template vertices are assigned to far-away target vertices when correspondences are established. Since parameter estimation minimizes the least-squares error, these outliers have a distorting effect. Therefore, the maximum distance between a template vertex and its correspondence is thresholded. Equations relating to points that exceed the threshold are omitted in the linear system.

The proposed algorithm is no exception to the rule that ICP based approaches are sensitive to local

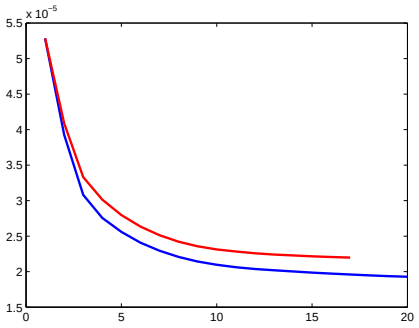

Figure 4: Blue curve: Mean squared errors during fitting with the proposed algorithm (average of 10 experiments). Red curve: Error curve for the shape fitting without pose correction. In both cases, the model's mean head was previously aligned with the point cloud by regular ICP. Simultaneous pose optimization leads to better fitting results. The model's mean head has a height of one.

minima. Therefore, the point cloud must be roughly aligned with the mean head of the model before optimization. While it often suffices to match scale and align principal axes, more difficult cases require a rigid alignment which we find with the "classic" ICP algorithm.

5 Evaluation and application

Figure 3 shows results of the fitting algorithm for a number of male heads with different facial characteristics. The model we use is based on a database of 180 laser-scans of male adults. The fitting targets were omitted from the model database. The regions used are depicted in figure 1. Each region was modeled with 15 eigenvectors. The average number of iterations required was 20. The blue curve in figure 4 shows an average progression of error during optimization.

By omitting the parts of the fitting equation related to pose and scale, the proposed algorithm can be used to estimate only shape parameters. So does simultaneous optimization of pose and shape yield better results than a plain shape estimation? We rigidly aligned point clouds with model's mean head—which is the starting point of the optimization—using regular ICP. Then, we fitted the model to the point clouds with and without pose optimization. The resulting mean error curves during optimization are shown in figure 4. Clearly, simultaneous pose optimization leads to better fitting results.

The proposed fitting algorithm cannot only be applied to point clouds of the complete head from laser-scanners or 3D reconstruction. Figure 5 shows a less obvious application example: From two mugshot type views of a head, plausible depth can be added to the face as follows. The profile view conveys the most specific information about the three-dimensional shape of a face. Therefore, we segment the face from the background and extract a profile curve from the border of the segmentation mask. Taking each pixel location on the profile curve and adding a z component of zero, we obtain a dense 3D point cloud of the face profile. The model is fit to this profile with the algorithm described above. Rotation is constrained to the axis perpendicular to the image plane of the profile curve and translation is only allowed in the image plane. Note that most vertices are outliers during this fitting process; however, the fitting is still stable due to the regularization described in section 4.2. Next, a number of landmarks are annotated in the frontal view to match the location of the eyes, the nostrils and the overall width of the face. The model is fit again, now, however, with fixed correspondences on the profile curve and the annotated landmarks. As the landmarks are only annotated in 2D, the missing depth component is added to the unknowns in equation (12). Note that this presupposes a orthographic projection model. Finally, the frontal view is projected on th resulting head mesh as texture.

6 Conclusion

The contributions of this paper can be summarized as follows: We adapted the ICP algorithm to fit a morphable model to an unstructured point cloud without preceding nonrigid registration. Along with the shape parameters, the algorithm explicitly estimates pose and scale which yields better fitting results than fitting shape alone. We showed that a complex model consisting of multiple PCA models for individual regions can be integrated into the optimization scheme. We also sketched the use of the proposed algorithm for adding depth to mugshot-style photographs.

Figure 3: Results of the fitting algorithm. Laser scans in the bottom row, fitted model in the top row. The model heads match the facial characteristic of the different target heads.

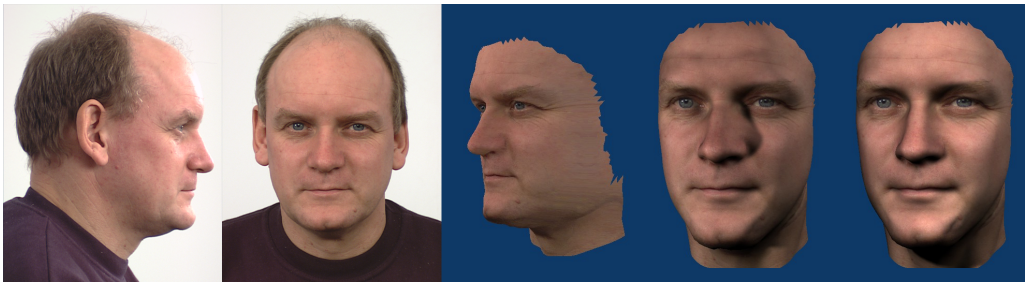

Figure 5: Model-driven depth, added to a photo by fitting the model to a mugshot view shown in the first and second image. Image 3 to 5 show changes in viewpoint and / or lighting. The model is first fit to a profile curve obtained from a segmentation of the first image. Manually annotated landmarks in the second image correct the width of facial features and the overall face. Finally the frontal image is projected on the mesh.

References

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

[2] Brian Amberg, Andrew Blake, Andrew Fitzgibbon, Sami Romdhani, and Thomas Vetter. Reconstructing High Quality Face-Surfaces using Model Based Stereo. In *Proceedings ICCV*, 2007.

[3] Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression Invariant 3D Face Recognition with a Morphable Model. In *8th International Conference Automatic Face and Gesture Recognition*, 2008.

[4] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal Step Nonrigid ICP Algorithms For Surface Registration. In *Proc. Conference on Computer Vision and Pattern Recognition CVPR'07*, 2007.

[5] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. SCAPE: Shape Completion and Animation of People. In *Proc. of ACM SIGGRAPH*, 2005.

[6] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. The Correlated Correspondence Algorithm for Unsupervised Registration of Nonrigid Surfaces. *NIPS*, 17:33–40, 2005.

[7] Paul J. Besl and Neil 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] Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Proc. of ACM SIGGRAPH*, 1999.

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

[11] Yang Chen and Gérard Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10:145 – 155, 1992.

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

[13] Dirk Haehnel, Sebastian Thruny, and Wolfram 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.

[14] Kolja Kähler, Jörg Haber, Hitoshi Yamauchi, and Hans-Peter Seidel. Head Shop: Generating Animated Head Models With Anatomical Structure. In *Proc. of ACM SIGGRAPH*, 2002.

[15] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju. Silhouette-Based 3D Face Shape Recovery. In *Graphics Interface GI'05, Halifax*, 2003.

[16] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. Registration of Point Cloud Data from a Geometric Optimization Perspective. In *Eurographics Symposium on Geometry Processing*, 2004.

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

[18] Sami Romdhani. *Face Image Analysis using a Multiple Features Fitting Strategy*. PhD thesis, Universität Basel, 2005.

[19] Sami Romdhani and Thomas 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.

[20] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Third International Conference on 3D Digital Imaging and Modeling*, 2001.

[21] David C. Schneider and Peter Eisert. Fast nonrigid mesh registration with a data-driven deformation prior. In *International Conference on Computer Vision, NORDIA workshop, Kyoto*, 2009.

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