

Algorithms For Automatic And Robust Registration Of 3D Head Scans

David C. Schneider and Peter Eisert

Fraunhofer Institute for Telecommunications—Heinrich-Hertz Institute
Einsteinufer 37, 10587 Berlin, Germany

email: {david.schneider | peter.eisert}@hhi.fraunhofer.de

Abstract

Two methods for registering laser-scans of human heads and transforming them to a new semantically consistent topology defined by a user-provided template mesh are described. Both algorithms are stated within the Iterative Closest Point framework. The first method is based on finding landmark correspondences by iteratively registering the vicinity of a landmark with a re-weighted error function. Thin-plate spline interpolation is then used to deform the template mesh and finally the scan is resampled in the topology of the deformed template. The second algorithm employs a morphable shape model, which can be computed from a database of laser-scans using the first algorithm. It directly optimizes pose and shape of the morphable model. The use of the algorithm with PCA mixture models, where the shape is split up into regions each described by an individual subspace, is addressed. Mixture models require either blending or regularization strategies, both of which are described in detail. For both algorithms, strategies for filling in missing geometry for incomplete laser-scans are described. While an interpolation-based approach can

be used to fill in small or smooth regions, the model-driven algorithm is capable of fitting a plausible complete head mesh to arbitrarily small geometry, which is known as “shape completion”. The importance of regularization in the case of extreme shape completion is shown.

Keywords nonrigid registration, ICP, geometry interpolation, morphable head models, shape completion, 3D face processing

1 Introduction

Laser scanners are a popular tool for acquiring detailed 3D models of human heads. However, the meshes generated by the scanners typically have a topology that reflects the operation of the scanner and is unsuitable for many applications. The data-set used in this work, for example, has a cylindrical grid topology with vertices of the form (ϕ_i, z_i, r_i) where ϕ_i are regularly spaced angles, z_i are regularly spaced vertical offsets and r_i are varying radii. In this paper, we describe two methods for semantically consistent registration and topology transfer of laser-scanned human heads that do not require manual intervention. The algorithms employ a user-provided template mesh which specifies the target topology. From arbitrary head scans they compute meshes with the following properties:

- All generated meshes have the same topology, i.e. that of the reference mesh. Different heads only vary in vertex locations.
- The mesh topology is semantically interpretable, i.e. topologically equivalent vertices in different heads have the same “meaning” such as tip of the nose, center of upper lip, etc.

Digital Peer Publishing Licence

Any party may pass on this Work by electronic means and make it available for download under the terms and conditions of the current version of the Digital Peer Publishing Licence (DPPL). The text of the licence may be accessed and retrieved via Internet at
<http://www.dipp.nrw.de/>.

First presented at the Conference for Visual Media Production 2008, extended and revised for JVRB

The first algorithm proposed is purely geometric and computes the registration without prior knowledge. The second algorithm involves a semantic shape model of the human head, which is a variant of the popular *morphable head model* of Blanz and Vetter [BV99]. While the model-driven algorithm is more robust than the geometric one, it presupposes a database of head meshes that are already registered in a common, semantically interpretable topology. Therefore, the geometric algorithm can be used to generate the database from which the model is computed. Both algorithms build on an error minimization approach computed in the well known Iterative Closest Points framework.

We also address the use of the registration algorithms with incomplete data, which is often an issue when laser-scanners or other surface reconstruction methods are used. For the geometric algorithm, we propose an interpolation method for filling in deficient areas that are either small or smooth. The model-driven algorithm can be used to fill in even huge missing parts with complicated geometry in a plausible fashion. In fact, with proper regularization, it is capable of performing “shape completion”, i.e. computing plausible geometry that matches arbitrarily small pieces of given data.

The paper is structured as follows. In section 2 we give an overview of related work in the areas of ICP, morphable shape models and registration of head meshes. In section 3, we briefly recapitulate the algebra of first order ICP and introduce some notation. The geometric registration approach is described in section 4 and the model-driven scheme is treated in section 5.

2 Related work

2.1 Iterative Closest Points

Both algorithms use the Iterative Closest Points optimization scheme which was first introduced by Besl and MacKay [BM92]. ICP is typically used to compute a rigid transform that aligns two or more partially overlapping meshes—e.g. laser-scans of a larger scene—such that the overlapping parts match as good as possible according to some error function. Common error functions are distances of closest points or point-to-plane distances (e.g. Chen and Medioni [CM92]). The optimal transform depends on matching the right points in both meshes; to match the right points, how-

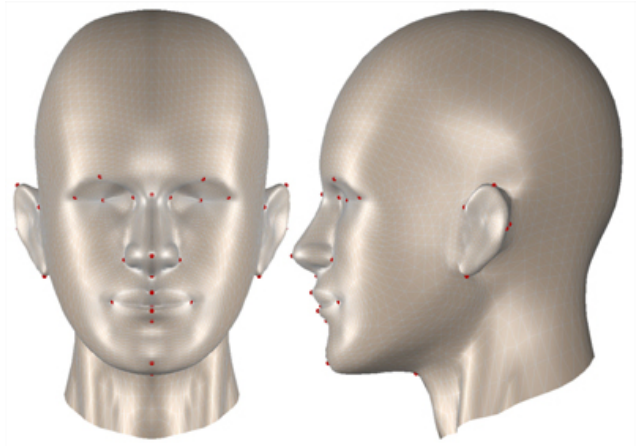


Figure 1: The reference mesh used for topology conversion with landmarks for the first algorithm stage.

ever, the optimal transform must be known. Therefore, ICP assumes that the closest points match; the transform induced by these correspondences is computed and applied and the procedure is iterated. For a rigid transformation, ICP is guaranteed to converge to a minimum of the error function [BM92]. However, this minimum may be local and thereby the meshes must be roughly aligned before ICP is invoked to find a good match.

There is a huge body of literature on ICP and numerous variants and optimizations have been proposed, aiming mostly at improving stability and speed of convergence. Rusinkiewicz and Levoy [RL01] give an overview and, more recently, Mitra et al. [MGPG04] introduced a general numeric optimization framework based on second order approximants which subsumes several previous approaches.

The ICP algorithm is neither limited to matching precisely overlapping meshes nor to the estimation of rigid transformations. In fact, any kind of transformation that is uniquely determined by a set of corresponding points can be estimated with ICP. However, as is the case for many optimization algorithms, the degrees of freedom of the transformation correlate with convergence: The more degrees of freedom there are, the more likely ICP is to converge to an undesired local minimum. Particularly relevant in the context of this work are non-rigid variants of ICP. These algorithms face the problem deforming the template in order to minimize the registration error while keeping its overall appearance. To this end, deformation is often controlled by a model that mimics the behavior of an elastic material. Examples of non-rigid ICP schemes

are Haehnel et al. [HTB03] who combine ICP optimization with a coarse to fine approach, or Amberg et al. [ARV07] who minimize a nonrigid matching error function similar to that of Allen et al. ([ACP03], see section 2.3) in a stepwise optimal fashion. Our model-driven algorithm (section 5) can be seen as a nonrigid ICP scheme with a data-driven deformation model; it was first published in [SE09] where we compare it to other approaches and address its computational complexity.

2.2 Morphable shape models

“Morphable models” are orthogonal subspace models of 3D shape (and sometimes texture) of a certain object domain. In order to compute a morphable shape model, a database of meshes sharing the same semantically consistent topology is required. This allows to treat the geometry of a mesh with N vertices as a single measurement in a $3N$ -dimensional data-space. The morphable model is obtained by performing Principal Component Analysis (PCA) on the database in this space. By omitting eigenvectors with low eigenvalues a low-dimensional subspace covering the most dominant variations in the dataset is obtained. For complex shapes, a single linear subspace typically cannot cover sufficient variability. Therefore, the geometry is often split up into regions and each region is described by its own subspace model. Typically, regions overlap at their borders in order to obtain a smooth overall mesh in the end. Thereby, the morphable model becomes a PCA mixture model.

Morphable models have been used most successfully in the domain of the human head and face, where they were introduced as “morphable head models” by Blanz and Vetter [BV99]. A wide number of applications in vision and graphics of faces can be driven or supported by morphable models, including, for example, face recognition [AKV08, BV03], face tracking [PF03, Rom05] and several varieties of 3D reconstruction, e.g. from uncalibrated video [Fua00] or from stereo images [ABF⁺07]. While examples besides faces are rare, a morphable model of teeth is described in Blanz et al. [BMVS04].

2.3 Head and face registration

Head and face registration algorithms can roughly be divided into methods that exploit texture information and purely geometric approaches. On the side of geometric methods, some specialize exclusively on heads

and faces while others try to solve the general problem of registering arbitrary surfaces by nonrigid transformations. The algorithms proposed in this paper belong to the class of head-specific purely geometric approaches. Several algorithms were introduced by the morphable model community since a database of registered meshes is required to compute a morphable model.

On the side of texture-based methods, Blanz and Vetter [BV99, VB98] use an algorithm based on optical flow. Traditionally used to estimate 2D image deformation, optical flow is extended to use 2D texture as well as local 3D surface properties of the face in order to compute a matching deformation. However, the authors themselves [BV99] as well as Paterson and Fitzgibbon [PF03] report the method to be unreliable on “exotic” faces. Thus Paterson and Fitzgibbon [PF03] manually annotate 30 landmark points in the texture and use radial basis function interpolation to compute a matching warp. 3D points are found by inverse-warping the texture coordinates associated with the 3D points.

Purely geometric head registration is used by Kähler et al. [KHYS02] to animate facial expressions of laser-scanned heads with an anatomical animation model. Their method is based on iteratively subdividing a rough face mesh and aligning the new vertices to a reference model. The initial model for the subdivision is obtained from manually placed landmarks. Allen et al. [ACP03] register full human body scans by using a generic nonlinear optimizer on an error function penalizing distances of closest points and dissimilarities between transforms of nearby points, thereby controlling the rigidity of the overall transformation. The approach requires some manually annotated landmarks to improve convergence. No landmarks are required by Anguelov et al. [ASP⁺05b], 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.

Geometric methods were also developed in the field of database retrieval and 3D face recognition. They typically aim at retrieving a stored prototype face from a database which is used to determine the identity of the scanned subject. To this end, Haar and Veltkamp [tHV08] use curves originating from the nose in all directions. Nair and Cavallaro [NC08] use ICP on incomplete faces in order to achieve invariance to facial expressions. These techniques, however, are

designed to retrieve a model of the same person rather than registering multiple faces in a semantically valid way.

Issues of filling in incomplete data, which are addressed for both algorithms described in this work, are treated in several papers. Knothe et al. [KRV06] use Local Feature Analysis, a localized variant of PCA, to fit a model to highly sparse data. Blanz et al. [BMVS04] have a probabilistic approach for dealing with sparse fitting targets, where we propose a geometric regularization (sections 5.2 and 5.3). Angelov et al. [ASP⁺05a] perform shape completion on full body scans in arbitrary pose. Note that in this work, the issue of filling in smaller holes is dealt with in the context of the geometric algorithm, while shape completion and fitting to extremely sparse data is treated in the context of the model-driven approach.

3 First order ICP

Both algorithms treated in this paper are best described on the basis of the linearized ICP algorithm. In the following, we briefly recapitulate the algebra of this approach and introduce some notation.

Let $\text{rot}(\boldsymbol{\theta})$ denote a rotation matrix for a vector $\boldsymbol{\theta}$ of three Euler angles. Further, be $\mathbf{t} \in \mathbb{R}^3$ a translation vector and s an anisotropic scale factor. Be $\mathbf{p}_i \in \mathbb{R}^3$ a vertex of the template mesh and $\mathbf{q}_i \in \mathbb{R}^3$ its corresponding point in the target point cloud. Putting point-to-plane distances and other optimizations aside, the ICP algorithm for estimating rigid transformation and anisotropic scale minimizes the following quadratic error function while simultaneously establishing the correspondences between template mesh and target point cloud:

$$e(s, \boldsymbol{\theta}, \mathbf{t}) = \sum_{i=1}^N \|s \text{rot}(\boldsymbol{\theta}) \mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2 \quad (1)$$

Here, N is the number of vertices in the template mesh.

A straightforward optimization strategy for ICP is to linearize this error functional around the parameter estimate $(s, \boldsymbol{\theta}, \mathbf{t})$ of the last iteration and solve a linear system to find a small change in the parameters that decreases the error. Formally, this amounts to lineariz-

ing

$$e(s + \Delta s, \boldsymbol{\theta} + \Delta \boldsymbol{\theta}, \mathbf{t} + \Delta \mathbf{t}) = \sum_{i=1}^N \|(s + \Delta s) \text{rot}(\Delta \boldsymbol{\theta}) \text{rot}(\boldsymbol{\theta}) \cdot \mathbf{p}_i + \mathbf{t} + \Delta \mathbf{t} - \mathbf{q}_i\|^2 \quad (2)$$

in $(\Delta s, \Delta \boldsymbol{\theta}, \Delta \mathbf{t})$. To this end, rotation can be approximated as

$$\text{rot}(\boldsymbol{\theta}) \mathbf{p} \approx \mathbf{p} + [\mathbf{p}]_{\times} \boldsymbol{\theta} \text{ for } \boldsymbol{\theta} \approx \mathbf{0}. \quad (3)$$

where $[\mathbf{p}]_{\times}$ is the cross product matrix

$$[\mathbf{p}]_{\times} := \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix}. \quad (4)$$

For a formal derivation, see, for example, [WI95]. Expanding equation (2) and omitting higher order terms yields, after rearranging, the following linear system:

$$\begin{bmatrix} \tilde{\mathbf{p}}_1 & [\tilde{\mathbf{p}}_1]_{\times} & \mathbf{I}_3 \\ \vdots & \vdots & \vdots \\ \tilde{\mathbf{p}}_N & [\tilde{\mathbf{p}}_N]_{\times} & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \boldsymbol{\theta} \\ \Delta \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 - \tilde{\mathbf{p}}_1 - \mathbf{t} \\ \vdots \\ \mathbf{q}_N - \tilde{\mathbf{p}}_N - \mathbf{t} \end{bmatrix} \quad (5)$$

For meshes of practical interest, this system is overdetermined and can be solved in a least squares sense, for example by SVD. The parameters are updated as follows:

$$\mathbf{R} \leftarrow \text{rot}(\Delta \boldsymbol{\theta}) \cdot \mathbf{R} \quad (6)$$

$$\mathbf{t} \leftarrow \Delta \mathbf{t} + \mathbf{t} \quad (7)$$

$$s \leftarrow (1 + \Delta s) \cdot s \quad (8)$$

Note that rotation is maintained as a matrix and updated by multiplication rather than by adding up Euler angles in order to avoid gimbal lock. Finally, the updated transformation is applied and new point correspondences are established for the next iteration.

Since laser-scans are often incomplete, some vertices of the reference mesh may be assigned to overly far points in the cloud. These vertices have a large effect on the optimization due to the use of squared errors. Therefore, they should be treated as outliers by introducing a threshold on the distance between a template vertex and its target correspondence. Equations corresponding with outliers can be eliminated from the linear system in equation (5) by multiplying both sides with a diagonal binary matrix.

4 Geometric algorithm

Our geometric registration algorithm comprises three steps: First, landmark points are found on the laser-scan mesh using the ICP scheme. Second, a nonlinear transform is computed to match the reference with the scan, using the landmark correspondences from step one as anchor points. Finally, the scan is resampled in the topology of the reference mesh and holes can be filled by interpolation.

4.1 Automatic landmark placement

In order to localize a landmark, the template mesh is repeatedly registered with the target point cloud. In every iteration, the ICP error function is re-weighted in order to concentrate the registration process on an increasingly smaller area around the landmark. The template mesh and the landmarks we use are shown in figure 1. The proposed localization strategy requires landmarks to have a non-flat, geometrically distinctive vicinity. Currently, landmark locations in the template are determined manually. However, *slippage analysis* introduced by Gelfand and Guibas [GG04] could be an interesting approach to automatic search for suitable landmark locations.

For re-weighting, the ICP error of equation (1) is extended by a vertex specific factor w_i , $i = 1 \dots N$ which yields

$$e(s, \theta, \mathbf{t}) = \sum_{i=1}^N w_i \|s\mathbf{R}\mathbf{p} + \mathbf{t} - \mathbf{q}\|^2. \quad (9)$$

This propagates to the linear system of equation (5) as a multiplication of the matrix with a diagonal weight matrix

$$\mathbf{W} = \text{diag}(w_1, \dots, w_N). \quad (10)$$

The role of the weights is to reduce the influence of the vertices distant from the landmark in an iterative manner. Generally, the weighting scheme should be a function of the form

$$w_i = \psi(\mathbf{p}_i, L, \alpha) \quad (11)$$

where \mathbf{p}_i is the vertex the weight of which is to be determined, L is the landmark that is currently processed, and α is a parameter that decreases with the iterations of the algorithm, controlling the size of the neighborhood that is to be considered in the ICP error term. Also, the weighting scheme must satisfy the constraints

$$0 \leq \psi(\mathbf{p}_i, L, \alpha) \leq 1 \quad (12)$$

$$\psi(\mathbf{p}_i, L, \infty) = 1 \quad (13)$$

Hence, by setting $\alpha = \infty$ in the first iteration, un-weighted ICP is performed.

There are numerous possibilities to implement this strategy. The most simple weighting scheme probably is the binary function

$$\psi(\mathbf{p}_i, L, \alpha) = \begin{cases} 1 & \text{dist}(\mathbf{p}_i, L) \leq \alpha \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where $\text{dist}(\mathbf{p}_i, L)$ is a measure of distance from a vertex \mathbf{p}_i to L . More complex weighting schemes replace the step function by a smooth falloff in L . Regarding the distance measure, there are again multiple possibilities. In our implementation, we use approximate geodesic distances computed with the fast marching algorithm of [KS98]. Note that the distance measures can be precomputed and stored as long as the reference mesh stays constant.

In summary, the algorithm for finding all landmarks is the following:

1. **for each** landmark L
 - (a) Set $\alpha \leftarrow \infty$.
 - (b) **while** α is above a threshold
 - i. Compute vertex weights $\psi(\mathbf{p}_i, L, \alpha)$ for $i = 1 \dots N$.
 - ii. Register template with error-weighted ICP and apply the transformation.
 - iii. Decrease α .
 - (c) Return the point in the cloud closest to L in the current pose of the template as the correspondence of L .

An illustration of the weighting function in the inner loop is shown in figure 3 for the upper lip landmark and the function of equation (14). Examples of automatically placed landmarks in different head scans are depicted in figure 2.

Limitations Depending on an individual head's geometry and on the quality of the laser-scan, the algorithm may converge to a bad localization of a landmark. Problem areas are the ears, where a good initial position of the template is required for good localization results, as well as the mouth where the geometry is not very pronounced for some individuals. A

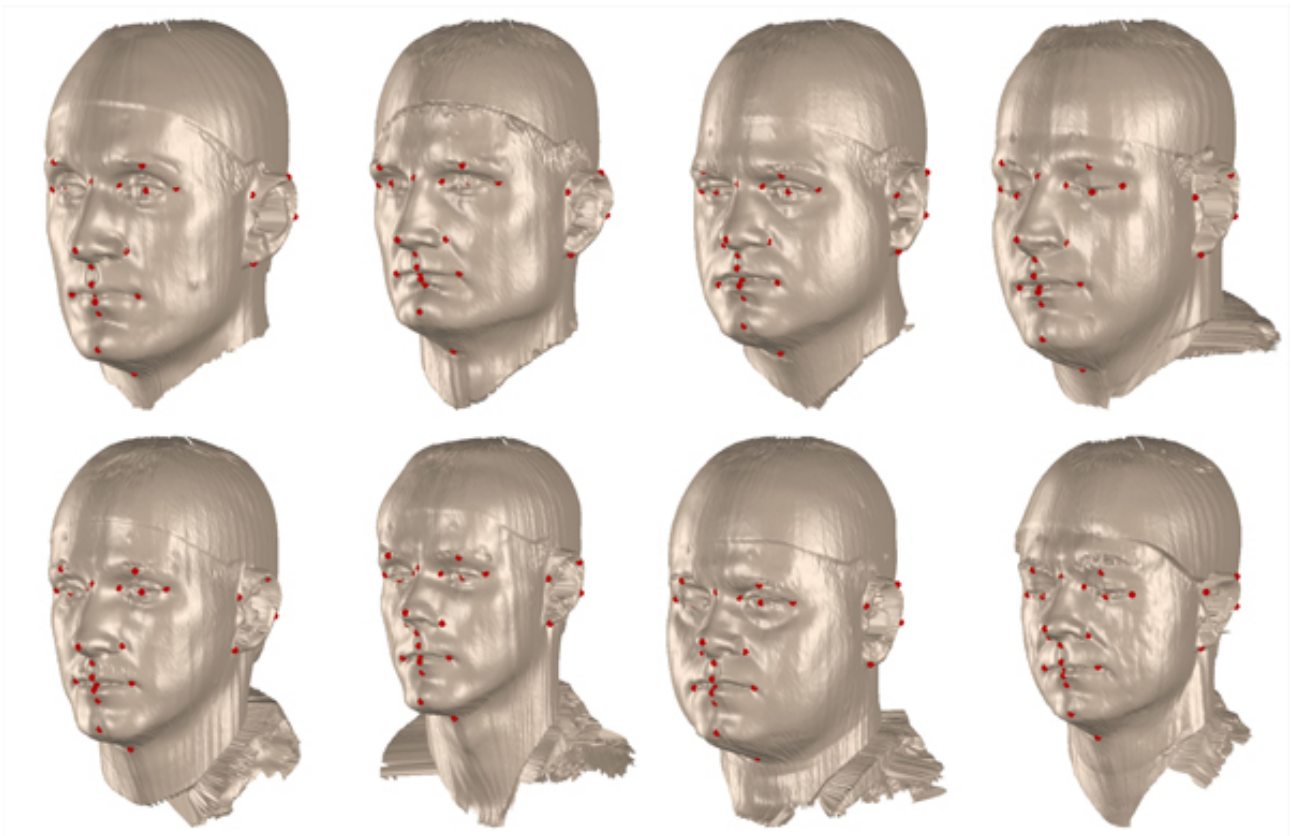


Figure 2: Examples of automatically placed landmark points in eight different raw laser-scans.

typical failure case in several landmarks is illustrated in figure 4. The algorithm also requires the scanned subjects to have the same facial expression as the template mesh. This is easiest to realize if both, subjects and template, display a neutral expression. Landmark localization quality is hard to evaluate numerically on our database since the low resolution of the laser-scans and lack of textures make a precise localization difficult for humans as well. However, the quality of the results is sufficient to build a morphable model that is capable of reproducing the facial characteristics of a wide range of individuals.

4.2 Nonlinear deformation

In the second step of the geometric algorithm, the template mesh is matched with the laser-scan point cloud in a global, nonlinear fashion using the established landmark correspondences. Denoting the landmarks in the template mesh by \mathbf{p}_i , $i = 1 \dots K$ and their point cloud correspondences by \mathbf{q}_i , we seek a transformation $\Phi(\cdot)$ such that $\Phi(\mathbf{p}_i) = \mathbf{q}_i$ for all i and such that the points between the landmarks are interpolated in a natural fashion. This is nicely realized by

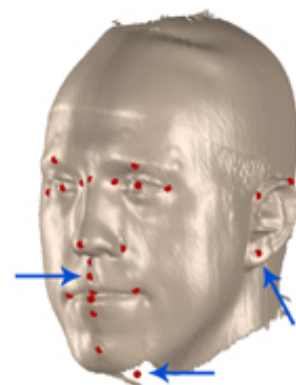


Figure 4: Failure case of landmark registration: Ear, mouth and missing parts in the scan may lead to mis-registration.



Figure 3: Landmark registration with an increasingly smaller area of influence for the error function: Example sequence of the influence area for the upper lip landmark with a binary weighting function.

the Thin Plate Spline formalism [Boo89], which yields a transformation minimizing a physically interpretable bending energy. Similar non-linear deformation models have been used successfully in face processing before, e.g. in [BKC⁺08, BBA⁺07].

>From a Thin Plate Spline perspective we solve a three dimensional scattered data interpolation problem from \mathbb{R}^3 to \mathbb{R}^3 . The known values are the displacement vectors from the laser-scan landmark points to the reference mesh landmarks,

$$\mathbf{d}_i = \mathbf{q}_i - \mathbf{p}_i. \quad (15)$$

The unknowns to be interpolated are the displacement vectors of all non-landmark points in the reference mesh. Dealing with a three dimensional problem, the radial basis function to use for the spline is $u(x) = |x|$ according to [Boo89] and thus, with $u_{i,j} = \|\mathbf{p}_i - \mathbf{p}_j\|$, we get a Thin Plate Spline matrix

$$\mathbf{S} = \begin{bmatrix} 0 & u_{1,2} & \dots & u_{1,K} & 1 & \mathbf{p}_1^T \\ u_{2,1} & 0 & & u_{2,K} & 1 & \mathbf{p}_2^T \\ \vdots & & \ddots & \vdots & \vdots & \vdots \\ u_{K,1} & u_{K,2} & \dots & 0 & 1 & \mathbf{p}_K^T \\ 1 & 1 & \dots & 1 & & \\ \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_K & & \mathbf{0}_{4 \times 4} \end{bmatrix}. \quad (16)$$

Hence the weight matrix for the mesh transform is

$$\mathbf{W} = \mathbf{S}^{-1} \begin{bmatrix} \mathbf{d}_1 & \dots & \mathbf{d}_K & \mathbf{0}_{3 \times 3} \end{bmatrix}^T \quad (17)$$

and an arbitrary point \mathbf{v} in the reference mesh transforms to

$$\mathbf{v}' = \mathbf{v} + \mathbf{W}^T \begin{bmatrix} u_1 & \dots & u_K & 1 & \mathbf{v}^T \end{bmatrix}^T \quad (18)$$

with $u_i = \|\mathbf{v} - \mathbf{p}_i\|$.

4.3 Resampling and hole filling

The final step is to resample the laser-scan in the topology of the deformed reference mesh which now closely matches the scan. Therefore, for each vertex

in the template mesh, its normal is computed and all intersections of the laser-scan with a line through the vertex in normal direction are determined. If there are multiple intersections, the one closest to the vertex is used. Moreover, there is a threshold on the distance between the vertex and the scan in order to exclude deficient matches at the scan's border. The chosen intersection point is taken as the correspondence to the template mesh vertex.

To speed up the computation of the ray-mesh-intersection, the laser-scan mesh is represented as an axis aligned bounding box tree and the fast ray box intersection of Williams et al. [WBMS05] is used. Note that due to the size of a typical scan it is not feasible to build the tree down to the level of individual triangles. In our implementation with around 80 000 points in a scan, there are 100 triangles per leaf that have to be tested for each ray.

Due to scanning errors, the resampled scans have holes where no valid correspondence along the normal was found. The interpolation scheme described in section 4.2 can be used to fill these holes as long as the missing geometry is smooth and not too complex; a model-based approach capable of filling in more complex geometry is described in section 5.3. Again, we can formulate the task as a scattered data interpolation problem from \mathbb{R}^3 to \mathbb{R}^3 . Denote by \mathbf{h}_i the vertices in the template mesh that lack correspondence in the scan (i.e. the vertices that make up the "hole"). Denote by \mathbf{p}_j vertices "surrounding the hole" that do have correspondences \mathbf{q}_j in the scan. The known values are the displacements at the surrounding vectors $\mathbf{d}_j = \mathbf{p}_j - \mathbf{q}_j$. To be interpolated are the displacements $\hat{\mathbf{d}}_i$ at the vertices lacking a reference. These can be computed with the Thin Plate Spline formalism described above. Finally, the correspondence of each "hole vertex" \mathbf{h}_i is set to $\mathbf{h}_i + \hat{\mathbf{d}}_i$.

Final results of the registration algorithm before and after hole filling are shown in figure 5.



Figure 5: Results of the geometric registration algorithm. Top row: Original laser-scans. Middle row: Result after registration and topology transfer. Bottom row: Final result after hole-filling by offset interpolation.

5 Model-driven algorithm

The model-driven algorithm is based on a morphable model of head shape, which was computed as described in section 2.2. The database required to build the model was generated with the geometric registration algorithm reported above. For more details on the model-driven algorithm, including comparisons to other registration methods and issues of time complexity, see [SE09].

In the following, be \mathbf{M} the morphable shape model matrix and $\boldsymbol{\mu}$ the model's mean vector. Hence the geometry of every mesh that can be represented by the model can be written as

$$\mathbf{g} = \mathbf{M}\mathbf{m} + \boldsymbol{\mu} \quad (19)$$

where

$$\mathbf{g} = (x_1 \ y_1 \ z_1 \ \dots \ x_N \ y_N \ z_N)^T \quad (20)$$

is a single vector of all vertex coordinates and the parameter vector \mathbf{m} is the location of the mesh in the model's "shape space". Note that our morphable model is a PCA mixture model comprising of six subspace models for mutually overlapping regions. Therefore, the model matrix is not simply a matrix of principal components. Details on the construction of \mathbf{M} for the mixture model are given in section 5.1.

To use the morphable model for laser-scan registration, we again start from the ICP error functional of equation (1) and substitute equation (19) for the template mesh vertices \mathbf{p}_i :

$$e(s, \boldsymbol{\theta}, \mathbf{t}, \mathbf{m}) = \sum_{i=1}^N \|s\mathbf{R}(\mathbf{M}_i\mathbf{m} + \boldsymbol{\mu}_i) + \mathbf{t} - \mathbf{q}_i\|^2 \quad (21)$$

Note that in the above equation \mathbf{M}_i and $\boldsymbol{\mu}_i$ refer to three-row blocks of \mathbf{M} and $\boldsymbol{\mu}$ that correspond with individual mesh vertices. Additional to the scale and rigid transform parameters, the error now also depends on the shape parameter vector \mathbf{m} . The addition of the shape model propagates to the matrix of the linear system of equation (5) as follows:

$$\begin{bmatrix} \tilde{\mathbf{p}}_1 & [\tilde{\mathbf{p}}_1]_{\times} & \mathbf{I}_3 & s \text{rot}(\boldsymbol{\theta})\mathbf{M}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{p}}_N & [\tilde{\mathbf{p}}_N]_{\times} & \mathbf{I}_3 & s \text{rot}(\boldsymbol{\theta})\mathbf{M}_N \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta \boldsymbol{\theta} \\ \Delta \mathbf{t} \\ \Delta \mathbf{m} \end{bmatrix} \quad (22)$$

See [SE09] for a more detailed derivation. In the ICP loop, the system is now also solved for a shape change



Figure 6: Partition of the head into regions for the PCA mixture model. Each part is described by its own subspace model.

$\Delta\mathbf{m}$. The shape is initialized by $\mathbf{m} \leftarrow \mathbf{0}$ which boils down to the mean shape $\boldsymbol{\mu}$ of the model according to equation (19).

Once the model's shape and pose are fitted to the laser-scan, there are two ways to proceed: Either, the scan can be resampled by establishing correspondences along the normals of the template vertices, as in the geometric algorithm (section 4.3). Alternatively, the fitted morphable model may itself be used as the registered version of the laser scan. In this case, the results are, of course, confined to the space of shapes the model can represent. Which strategy is best depends on the application: Resampling the scan is computationally more expensive and gives noisier results but it preserves idiosyncrasies of the scanned individual that may not be captured by the model.

Note that due to the shape parameters, there are significantly more degrees of freedom in the error function minimized by the model-driven method than in that minimized by rigid ICP. Therefore, the result of the model-driven algorithm is more sensitive to the initial alignment of template and target. Best results are achieved when template and target are pre-aligned with respect to pose and anisotropic scale by regular ICP.

5.1 PCA mixtures

For a simple morphable model with a single subspace, the model matrix \mathbf{M} is simply the eigenvector matrix yielded by the PCA. If a PCA mixture model with multiple subspaces is used, \mathbf{M} has to be constructed from the region's individual eigenvector matrices. See figure 6 for an illustration of the regions used in this work.



Figure 7: Results of the model-driven registration algorithm. Original laser-scans in the bottom row, fitted morphable model in the top row. Six region models and 15 principal components per region were used.

Assume the number of regions and hence the number of subspaces used is K and denote each region's eigenvector matrix by $\mathbf{M}^{(j)}$, $j = 1 \dots K$. Further be $\mathcal{R}_1 \dots \mathcal{R}_K$ sets of vertex indices such that $i \in \mathcal{R}_j$ if vertex \mathbf{p}_i belongs to region \mathcal{R}_j .

If the regions do not overlap, i.e. $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for $i \neq j$, the matrix \mathbf{M} has a block-diagonal structure:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}^{(1)} & & \\ & \ddots & \\ & & \mathbf{M}^{(K)} \end{bmatrix} \quad (23)$$

The results from this type of model, however, show harsh discontinuities between the regions since each region is optimized independently from the others. Therefore, the region models have to be coupled by introducing overlap at the region borders, i.e. vertices that belong to several regions. In consequence, the matrix \mathbf{M} is no longer simply block diagonal.

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 $\mathbf{M}^{(j)}$, multiplied with the shape pa-

rameters for that region. Denote these three rows of $\mathbf{M}^{(j)}$ by $\mathbf{M}_i^{(j)}$ and define

$$\mathbf{M}_i^{(j)} := \mathbf{0} \quad \text{if} \quad i \notin \mathcal{R}_j \quad (24)$$

where $\mathbf{0}$ is an all-zero matrix of appropriate size. Then, \mathbf{M} can be defined block-wise as

$$\mathbf{M} = \begin{bmatrix} \alpha_1^{(1)} \mathbf{M}_1^{(1)} & \dots & \alpha_1^{(K)} \mathbf{M}_1^{(K)} \\ \vdots & \ddots & \vdots \\ \alpha_N^{(1)} \mathbf{M}_N^{(1)} & \dots & \alpha_N^{(K)} \mathbf{M}_N^{(K)} \end{bmatrix} \quad (25)$$

where $\alpha_i^{(j)}$ are *block-weights* satisfying

$$\sum_{i|i \in \mathcal{R}_j} \alpha_i^{(j)} = 1. \quad (26)$$

Note that for a vertex \mathbf{p}_i that belongs just to one region, only one block $\mathbf{M}_i^{(j)}$ is non-zero. As regions are typically defined to overlap only at their borders, the majority of vertices will be of this kind and \mathbf{M} is fairly sparse.

The role of the block weights $\alpha_i^{(j)}$ is to normalize the contributions of different region models for vertices that belong to more than one region. The most simple block weight is $\alpha_i^{(j)} = \frac{1}{R_i}$ where R_i is the number of regions the vertex belongs to. If the area of overlap is deeper than just one shared border vertex, more sophisticated weights may be considered, giving more influence to some regions than to others. In the following we give an example of a weight that depends only on the mesh topology.

Be a *border edge* an edge with only one incident triangle and a *border vertex* a vertex that is incident to at least one border edge. Denote by \mathcal{B}_j the set of border vertices of region \mathcal{R}_j and by \mathcal{N}_i the index set of neighbors of the vertex with index i . Now consider an arbitrary vertex and define its “centricity” $c_i^{(j)}$ with respect to region \mathcal{R}_j recursively as

$$c_i^{(j)} = \begin{cases} 0 & \text{if } i \notin \mathcal{R}_j \\ 1 & \text{if } i \in \mathcal{B}_k \\ \min\{c_k^{(j)} \mid k \in \mathcal{N}_i \cap \mathcal{R}_j\} + 1 & \text{otherwise} \end{cases} \quad (27)$$

Now the blending weight $\alpha_i^{(j)}$ can be defined as

$$\alpha_i^{(j)} = \frac{c_i^{(j)}}{\sum_{k=1}^K c_i^{(k)}}. \quad (28)$$

Equation (28) is equal to one for vertices that appear in only one part. For vertices appearing in multiple parts the weight is higher with respect to parts where the vertex is further away from the part border. The sum of blending weights of one vertex is always one, thus satisfying the constraint in equation (26).

5.2 Geometric regularization

It is generally desirable to introduce as little overlap between the regions as possible: Large overlap increases the size of the linear system that has to be solved in every ICP iteration. Also, from a modeling point of view, every sub-model should ideally describe it’s own part of the domain (here: the face) and interfere with other parts as little as possible. The downside to small overlap is the risk of discontinuities as shown in figure 8: The smaller the overlap, the more independent is the optimization of every region. As the optimization minimizes a sum of squared errors over all vertices in the region, two regions may yield significantly different locations for their shared vertices.

This problem can be addressed—even for the smallest possible overlap of one shared line of vertices along

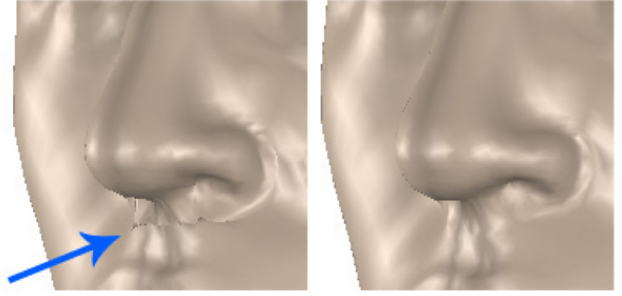


Figure 8: Effect of geometric regularization on a model with an overlap depth of one vertex. Left: A crease under the nose is clearly visible. Right: The same model fitted with geometric regularization.

the region border—by introducing a Tikhonov regularization term to the parameter estimation system. The aim of the regularization is to force shared vertices into the same position. Taking into account the block structure of \mathbf{M} as defined in equation (25), we can write the location \mathbf{p}_i of a vertex given shape and pose parameters $(s, \boldsymbol{\theta}, \mathbf{t}, \mathbf{m})$ as

$$\mathbf{p}_i = s \text{rot}(\boldsymbol{\theta}) \left(\sum_{j=1}^K \alpha_i^{(j)} \mathbf{M}_i^{(j)} \mathbf{m}^{(j)} + \boldsymbol{\mu}_i \right) + \mathbf{t}. \quad (29)$$

Note that $\mathbf{m}^{(j)}$ is the part of the shape vector that determines the shape of region \mathcal{R}_j . Now assume that \mathbf{p}_i is shared by exactly two regions \mathcal{R}_S and \mathcal{R}_T (i.e. it is a vertex at the border between two regions). Therefore, $\alpha_i^{(j)} = 0$ for all values of j except for $\{S, T\}$ and the summation in the above equation reduces to $\alpha_i^{(S)} \mathbf{M}_i^{(S)} \mathbf{m}^{(S)} + \alpha_i^{(T)} \mathbf{M}_i^{(T)} \mathbf{m}^{(T)}$. Clearly, the two regions yield the same location for the vertex, if

$$\alpha_i^{(S)} \mathbf{M}_i^{(S)} \mathbf{m}^{(S)} - \alpha_i^{(T)} \mathbf{M}_i^{(T)} \mathbf{m}^{(T)} = \mathbf{0}. \quad (30)$$

This term can be expressed in matrix form and added to the linear system for every vertex that is a border vertex between two regions. For vertices shared by more than two regions, each unordered pair of regions yields one regularization term. In practice, these vertices can be ignored without harm.

5.3 From hole filling to shape completion

In section 4.3 we sketched an interpolation algorithm that can be used to fill in holes in the laser-scans after registration with the geometric algorithm. The data-driven methodology, too, lends itself to a hole-filling—in fact the use of a model allows much more

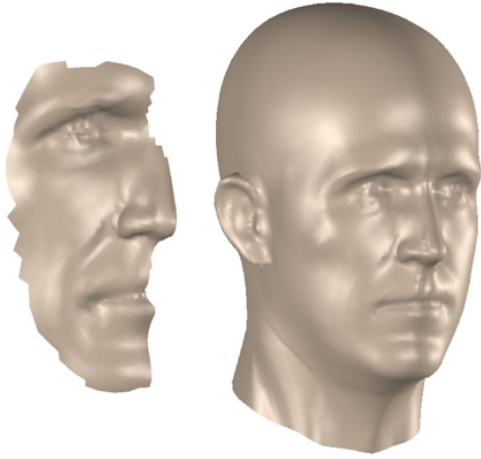


Figure 9: Shape completion with the model-driven fitting algorithm. The fitting target is only a small piece of geometry (left). A plausible complete head (right) is fitted to the given data. Note that there is no given data for the ear region.

radical “filling in”, up to the point of generating a plausible head that closely matches a very small piece of given geometry. This is sometimes referred to as *shape completion* in the literature.

In order to cope with holes in the scan, outlier handling should be added to the optimization as described in section 3. Once the model is fitted to the scan, no additional filling or interpolation step is necessary: The shapes generated by the model are—by construction—always complete. Algebraically, this follows from the fact that the system of equation (22) loses rows when outliers occur but not columns. Therefore and regardless of the outliers, a complete shape parameter vector \mathbf{m} is always computed. A full mesh is simply obtained by multiplying \mathbf{m} with the the original shape matrix that contains all rows.

There is a caveat, however, if huge parts of the target geometry are missing, which is typical for the shape completion scenario. This can be seen by looking more closely at the effects of outlier removal on the matrix \mathbf{M} . When a vertex with index i is classified as outlier due to the lack of a nearby target point, a three row block

$$\left[\alpha_i^{(1)} \mathbf{M}_i^{(1)} \quad \dots \quad \alpha_i^{(K)} \mathbf{M}_i^{(K)} \right]$$

corresponding to the outlier vertex is removed from \mathbf{M} . This is uncritical as long as there are sufficiently many vertices left in the regions to which the removed vertex belongs. What happens, however, if all information in a region \mathcal{R}_j is lost? It can be seen from the construction of \mathbf{M} in equation (25), that in this case all $\mathbf{M}_i^{(j)}$ for which $\alpha_i^{(j)} \neq 0$ are lost and \mathbf{M} contains a number of zero-columns. In consequence, the elements of \mathbf{m} that determine the shape of the “unknown” region are undetermined. If, however, geometric regularization is used, there is still sufficient information in the Tikhonov matrix to determine the shape parameters of region \mathcal{R}_j . Geometrically, this is the information that whatever the geometry of the region is, it must match the geometry of its neighboring region in the shared vertices. Note that this information also helps to prevent overfitting in the case of very few inlier points. Figures 9 and 10 show some extreme examples of shape completion. Note that in both examples, the ears are completely undetermined by the given data.

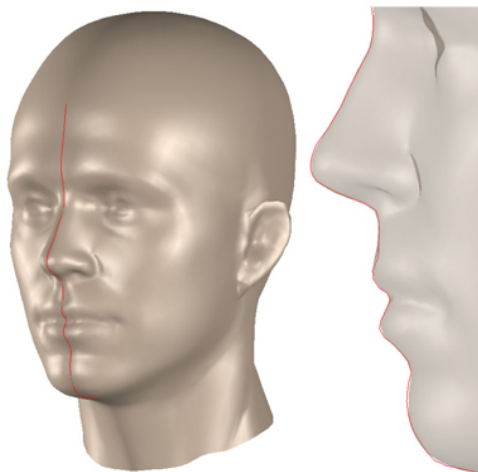


Figure 10: In this shape completion example, the target is only the red profile curve, which was extracted from a photo. The fitted head closely matches the target profile (right).

6 Conclusion and outlook

We described two robust methods for registering laser-scans of human heads and transforming them to a new semantically consistent topology defined by a user-provided template mesh. The first algorithm is based on finding landmark correspondences by iteratively registering the vicinity of a landmark with a re-weighted error function. Thin-plate spline interpolation is then used to deform the template mesh and finally the scan is resampled in the topology of the deformed template. The second algorithm directly optimizes pose and shape of a morphable shape model which can be computed using the results of the first algorithm. We outlined blending and regularization strategies for coping with PCA mixture models where the shape is split up into regions each described by an individual subspace. For both algorithms, we addressed strategies for filling in missing geometry for incomplete laser-scans. While the interpolation-based approach can fill in small or smooth regions, the model-driven strategy is capable of fitting a plausible complete head mesh to arbitrarily small geometry. We addressed the importance of regularization in the case of extreme shape completion.

Regarding the geometric algorithm, future research will concentrate on different re-weighting schemes in order to improve precision of automatic landmark placement, especially for difficult cases such as the ears and the corners of the mouth. For the model-driven method, an interesting open question is the choice of regions in the PCA mixture model. Currently, the regions are defined manually and based on an intuitive segmentation of the head into nose, mouth, eyes, etc. It would be interesting to investigate data-driven strategies that yield a segmentation as the result of an optimization process. The incorporation of classical ICP optimizations such as point-to-plane distances into both algorithms is another interesting direction of research.

References

- [ABF⁺07] Brian Amberg, Andrew Blake, Andrew Fitzgibbon, Sami Romdhani, and Thomas Vetter, *Reconstructing High Quality Face-Surfaces using Model Based Stereo*, Proc. International Conference on Computer Vision, 2007, pp. 1–8, ISBN 978-1-4244-1630-1.
- [ACP03] Brett Allen, Brian Curless, and Zoran Popovic, *The Space Of Human Body Shapes: Reconstruction And Parameterization From Range Scans*, Proc. of ACM SIGGRAPH, 2003, pp. 587 – 594, ISBN 1-58113-709-5.
- [AKV08] Brian Amberg, Reinhard Knothe, and Thomas Vetter, *Expression Invariant 3D Face Recognition with a Morphable Model*, 8th International Conference Automatic Face and Gesture Recognition, 2008, pp. 1–6, ISBN 978-1-4244-2153-4.
- [ARV07] Brian Amberg, Sami Romdhani, and Thomas Vetter, *Optimal Step Nonrigid ICP Algorithms for Surface Registration*, Proc. Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8, ISBN 1-4244-1180-7.
- [ASP⁺05a] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis, *SCAPE: Shape Completion and Animation of People*, Proc. of ACM SIGGRAPH, 2005, ISSN 0730-0301, pp. 408–416.
- [ASP⁺05b] Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis, *The Correlated Correspondence Algorithm for Unsupervised Registration of Nonrigid Surfaces*, Advances in neural information processing systems 17: proceedings of the 2004 conference, vol. 17, 2005, pp. 33–40, ISBN 9780262195348.
- [BBA⁺07] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross, *Multi-Scale Capture of Facial Geometry and Motion*, Proc. of ACM SIGGRAPH, 2007, Article no. 33, ISSN 0730-0301.
- [BKC⁺08] L. Benedikt, V. Kajić, D. Cosker, P. L. Rosin, and D. Marshall, *Facial Dynamics in Biometric Identification*, Proc. British Machine Vision Conference, 2008.
- [BM92] Paul J. Besl and Neil D. McKay, *A Method for Registration of 3-D Shapes*,

- IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992), no. 2, 239–256, ISSN 0162-8828.
- [BMVS04] Volker Blanz, Albert Mehl, Thomas Vetter, and Hans-Peter Seidel, *A Statistical Method for Robust 3D Surface Reconstruction from Sparse Data*, International Symposium on 3D Data Processing, Visualization and Transmission, 2004, pp. 293–300, ISBN 0-7695-2223-8.
- [Boo89] F. L. Bookstein, *Principal Warps: Thin-Plate Splines and the Decomposition of Deformations*, IEEE Transactions on Pattern Analysis and Machine Intelligence **11** (1989), no. 6, 567 – 585, ISSN 0162-8828.
- [BV99] Volker Blanz and Thomas Vetter, *A Morphable Model for the Synthesis of 3D Faces*, Proc. of ACM SIGGRAPH, 1999, pp. 1063–1074, ISBN 0-201-48560-5.
- [BV03] Volker Blanz and Thomas Vetter, *Face Recognition Based on Fitting a 3D Morphable Model*, IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003), no. 9, 1063 – 1074, ISSN 0162-8828.
- [CM92] Yang Chen and Gérard Medioni, *Image and Vision Computing* **10** (1992), no. 3, 145–155, ISSN 0262-8856.
- [Fua00] P. Fua, *Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data*, International Journal of Computer Vision **38** (2000), no. 2, 153–171, ISSN 0920-5691.
- [GG04] Natasha Gelfand and Leonidas Guibas, *Shape Segmentation Using Local Slippage Analysis*, Eurographics Symposium on Geometry Processing, 2004, pp. 214–223, ISBN 3-905673-13-4.
- [HTB03] Dirk Haehnel, Sebastian Thruny, and Wolfram Burgard, *An Extension of the ICP Algorithm for Modeling Nonrigid Objects with Mobile Robots*, Proc. of the International Joint Conference on Artificial Intelligence, 2003, pp. 915–920.
- [KHYS02] Kolja Kähler, Jörg Haber, Hitoshi Yamachi, and Hans-Peter Seidel, *Head Shop: Generating Animated Head Models With Anatomical Structure*, Proc. of ACM SIGGRAPH, 2002, pp. 55–63, ISBN 1-58113-573-4.
- [KRV06] Reinhard Knothe, Sami Romdhani, and Thomas Vetter, *Combining PCA and LFA for Surface Reconstruction from a Sparse Set of Control Points*, International Conference Automatic Face and Gesture Recognition, 2006, pp. 637–644, ISBN 0-7695-2503-2.
- [KS98] R. Kimmel and J. A. Sethian, *Computing Geodesic Paths on Manifolds*, Proc. of National Academy of Sciences **95** (1998), no. 15, 8431–8435, ISSN 0027-8424.
- [MGPG04] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas, *Registration of Point Cloud Data from a Geometric Optimization Perspective*, Eurographics Symposium on Geometry Processing, 2004, pp. 22–31, ISBN 3-905673-13-4.
- [NC08] Prathap Nair and Andrea Cavallaro, *Matching 3D Faces with Partial Data*, Proc. British Machine Vision Conference, 2008.
- [PF03] J. Paterson and A. Fitzgibbon, *3D Head Tracking Using Non-Linear Optimization*, British Machine Vision Conference 03, 2003, pp. 609–618.
- [RL01] Szymon Rusinkiewicz and Marc Levoy, *Efficient Variants of the ICP Algorithm*, Third International Conference on 3D Digital Imaging and Modeling, 2001, pp. 145–152, ISBN 0-7695-0984-3.
- [Rom05] Sami Romdhani, *Face Image Analysis using a Multiple Features Fitting Strategy*, Ph.D. thesis, Universität Basel, 2005.
- [SE09] David C. Schneider and Peter Eisert, *Fast Nonrigid Mesh Registration with a Data-Driven Deformation Prior*, ICCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, 2009.

- [tHV08] Frank B. ter Haar and Remco C. Veltkamp, *A 3D Face Matching Framework*, Proc. IEEE Shape Modeling International, 2008, pp. 103–110, ISBN 978-1-4244-2260-9.
- [VB98] Thomas Vetter and Volker Blanz, *Estimating coloured 3D face models from single images: An example based approach*, Proc. European Computer on Computer Vision, 1998, pp. 499–513, ISBN 3-540-64613-2.
- [WBMS05] Amy Williams, Steve Barrus, R. Keith Morley, and Peter Shirley, *An Efficient and Robust Ray-Box Intersection Algorithm*, International Conference on Computer Graphics and Interactive Techniques, 2005, Article no. 9.
- [WI95] M.D. Wheeler and Katsushi Ikeuchi, *Iterative Estimation of Rotation and Translation using the Quaternion*, Tech. Report CMU-CS-95-215, Computer Science Department, Carnegie Mellon University, 1995.

Citation

David C. Schneider and Peter Eisert, <i>Algorithms For Automatic And Robust Registration Of 3D Head Scans</i> , Journal of Virtual Reality and Broadcasting, 7(2010), no. 7, Oktober 2010, urn:nbn:de:0009-6-26626, ISSN 1860-2037.
