Markerless Reconstruction of Dynamic Facial Expressions

Dominik Sibbing Martin Habbecke Leif Kobbelt
RWTH-Aachen University
{sibbing, habbecke, kobbelt}@cs.rwth-aachen.de

Abstract

In this paper we combine methods from the field of computer vision with surface editing techniques to generate animated faces, which are all in full correspondence to each other. The input for our system are synchronized video streams from multiple cameras. The system produces a sequence of triangle meshes with fixed connectivity, representing the dynamics of the captured face. By carfully taking all requirements and characteristics into account we decided for the proposed system design: We deform an initial face template using movements estimated from the video streams. To increase the robustness of the initial reconstruction, we use a morphable model as a shape prior. However using an efficient Surfel Fitting technique, we are still able to precisely capture face shapes not part of the PCA Model. In the deformation stage, we use a 2D mesh-based tracking approach to establish correspondences in time. We then reconstruct image-samples in 3D using the same Surfel Fitting technique, and finally use the reconstructed points to robustly deform the initially reconstructed face.

1. Introduction

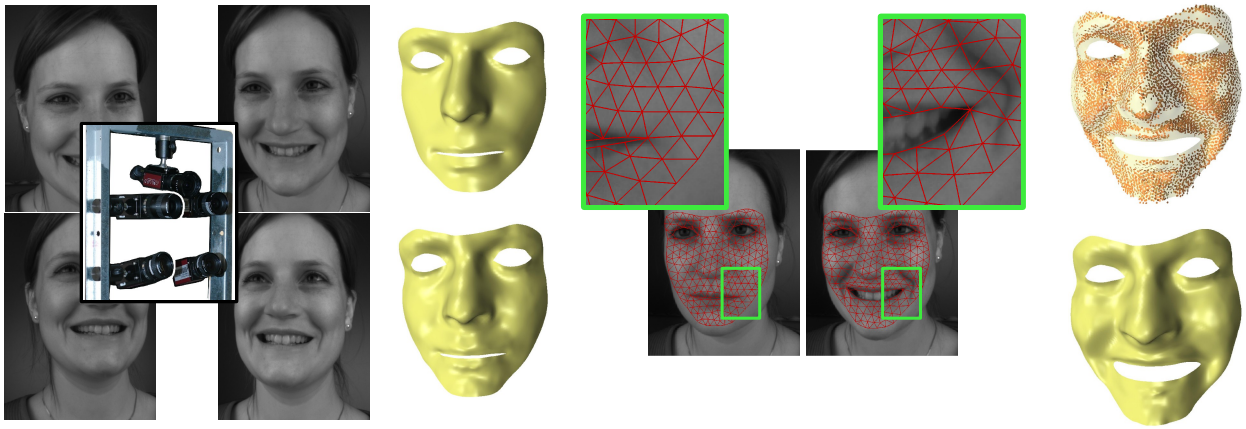
The dense motion capture of facial movements is an important part to generate data driven facial animations. The acquired motion data can be used to create animations for movies, computer games, or humanoid avatars which can be utilized in scientific as well as industrial applications. Standard tracker based motion capture systems often record only sparse temporal and spatial data. Fortunately, modern multi-camera and computer systems allow for the acquisition and analysis of a large amount of data, so the dense reconstruction of facial movements becomes possible.

In this paper, we exploit methods from computer vision and mesh editing to compute a dense motion field for facial animations from synchronized video streams. The motion field is represented by a predefined face template whose vertices move in time according to the underlying scene flow. In many applications such as the retargeting of facial movements, expression blending or statistical analysis of motion data, it is essential to establish correspondences between vertices not only from frame to frame but also between different data sets. Since we use a predefined face template which is fitted to an individual face, we immediately obtain the correspondences between all acquired reconstructions. Our predefined face template is a simple morphable model whose low-dimensional set of parameters is able to control the shape of neutral looking faces. In our work, this simple model additionally helps to stabilize the 3D multi-view stereo reconstruction of human faces, since it provides a good initial solution for a stereo reconstruction algorithm.

1.1. Related Work

A common technique to generate (caricaturated) facial movements for movies and computer games is Free Form Deformation (FFD). FFD provides a framework which allows artists to drag points on a cage or within 3D space to intuitively deform the space around that cage and thus the underlying geometry. The major difficulty for these methods is to find efficient mappings which do not induce large distortions (such as volumetric shrinking) to the model [13, 18, 1].

One way to simulate more realistic movements of a human face is to use physically based methods, which usually rebuild some anatomical features of the human head with the aim of mimicking natural movements. In [24] Waters and Keith develop a parameterized facial muscle process by abstracting the facial action units originally introduced by Ekman and Friesen [7]. However, their work uses only a few muscles to reproduce basic human emotions. Similarly, Lee *et al*. [16] build an anatomically accurate physically based head model. They map the geometry and texture, obtained from laser scans, to a generic head model and augment this model with multilayer facial tissue, a skull and synthetic muscles, which are used to deform the tissue to produce facial expressions. Kähler *et al*. [14] use a similar model to perform real-time deformations based on antropometrically meaningful landmarks. Their method is also capable of simulating aging. A more recently introduced biomechanically based muscle model was suggested by Sifakis *et al*. [22]. This anatomically accurate model



(a) (b) (c) (d)

Figure 1. Workflow to reconstruct a dynamic face. (a) Views from different directions and camera rig. (b) Top: Morphable model after optimizing rigid transformation and shape parameters. Down: Reconstruction of the neutral face from the first image. (c) The *2D mesh tracking* establishes temporal correspondence between the frames. (d) Top: The *Surfel Fitting* produces a point could, which may contains some holes. Down: Result of the *3D mesh tracking*: Successfully reconstructed Surfels define constraints for a non-rigid deformation of the initial face template.

uses finite elements methods to deform the synthetic tissues around a skull model. It also uses a set of sparse surface landmarks to track facial movements with a motion capture system. The major problem with all biomechanical models is that they are difficult to build correctly, because our anatomical knowledge about human skin, muscles, and bone structures is still incomplete. Thus, models sometimes require extensive tuning to produce realistic output.

Data driven facial animation usually involves a motion tracking system that records the movement of markers placed in the face, see *e.g*. [5, 22, 2, 26]. The captured trajectories typically represent the movements of the face in a very sparse way. To improve geometric details, Bickel [2] added wrinkles to a facial base mesh, which is deformed by the motion capture data using a shell-based mesh deformation method. Active Sensing methods project special light patterns or colors onto an object to capture the three dimensional motion field without the use of markers. Hernández *et al*. [11] use multispectral photometric stereo to compute a dense normal field from untextured surfaces. Weise *et al*. [25] use active illumination based on phase-shift to reconstruct surfaces at high framerates. A drawback of this work is that they are unable to maintain correspondences between vertices in time. Additionally special hardware which is often not commercially available (like the scanner used by Weise *et al*.) is required. Blanz and Vetter [3] learn shape and texture parameters for a morphable model by performing a Principal Component Analysis (PCA) on a set of laser scans of human heads. In an image based approach, they optimize these parameters to extract the geometry and texture of a human head from a photo. In a related approach Dellepiane *et al*. [6] deform a dummy head to reconstruct the shapes of human heads from images and used them for binaural rendering. Active Appearance Models (AAMs), as in [20, 15], are used to track motion through (multiview) image sequences. As with morphable models, though, the reconstructions are always restricted to the low-dimensional space spanned by the parametric model. Vedula *et al*. introduced the term *dense scene flow* in [23], which was further improved by Li and Sclaroff [17]. They reformulate the optical flow problem, find corresponding pixels in time, and use disparity to find correspondences between different views. The extraction of geometric information which could be used for simple visibility tests was not considered. In [9] Furukawa extended their reconstruction approach to track vertices of a mesh reconstructed in the first image of a video stream. In the examples the captured faces had to be endued with additional paint to obtain highly textured surfaces. They also do not use a predefined template face, which keeps its topology over all frames.

1.2. Overview

To generate the input for our system, a camera rig with five synchronized cameras was constructed and calibrated to capture the dynamic facial expressions of different subjects. Each of the cameras record images at 30 FPS with a resolution of 640×480 (Figure 1a shows the rig, together with four of these images in the middle of a sequence). If necessary, the framerate could be increased to 60 FPS by triggering the cameras externally.

The first image of each sequence shows the face in its neutral pose. In order to be able to track facial movement we need to reconstruct the face seen in the first frame. Independently optimizing point depth values using *Surfel Fitting*

would produce a point cloud of arbitrary size. In addition to this, it can contain holes and outliers. We decided to use a simple morphable model to estimate the shape of the face seen in the first frame (see Section 3.1). This drastically increases to robustness of the *Surfel Fitting* because of the good initial Surfel parameters. One requirement of our system is that inter-subject correspondences have to be maintained. This becomes possible by using a face template, containing a fixed number of vertices, which is in a one-to-one correspondence with the vertices of the morphable model. Morphable models are restricted to a space spanned by their training examples. In order to represent more general shapes we additionally non-rigidly deform the resulting model to produce a smooth face template, which is used for all further steps of the pipeline (Figure 1b).

The initial face template is deformed during the whole sequence while correspondences between the vertices of the template and the captured face are maintained. To achieve this, we combine mesh modeling techniques with multi-view stereo reconstruction: 2D image-samples, placed in the first image of every view, are tracked over the entire sequence (Figure 1c). In order to establish temporal correspondences between successive frames, we use a 2D mesh based tracking approach. Simple feature tracker like the KLT tracker [21] often have the problem that features slide past each other, since their displacements ore optimized independently of their local neighborhood. In the proposed *2D mesh tracking* (Section 3.2) we can control for the global smoothness of the produced mapping, to prevent foldovers. Shooting rays through an image-sample of the first image hits the initial face template and thereby defines an anchor point in 3D. At each step, the tracked image-samples are reconstructed using the *Surfel Fitting* approach [10] (top of Figure 1d). Together with its anchor point lying on the surface of the face template, a successfully reconstructed image-sample will provide a constraint which is used in the modeling step to deform the face template (Figure 1d). See Section 3.3 for a more precise description. The proposed modelling step has two advantages: First, if the *Surfel Fitting* does not succeed the face template can still be deformed using surrounding successfully reconstructed Surfels. Second, since the tracked face template provides good initial solutions, *Surfel Fitting* becomes much more robust.

Since *Surfel Fitting* is used to reconstruct the initial face, as well as to track facial movements we will describe this approach in the next section. The advantage of this algorithm is its simplicity since each Surfel can be optimized independent of its local neighborhood.

2. Surfel Fitting

Our 3D multi-view stereo reconstruction method is based on the simple to use *Surfel Fitting* approach introduced by [10]. Assume we have an initial estimate of a surface element (Surfel) defined by a point and a normal. The Surfel's associated plane defines a homography which maps pixels from a calibrated reference image I_r to a calibrated comparison image I_c. *Surfel Fitting* optimizes the parameters of the plane by minimizing pixel intensity differences between the reference and comparison images.

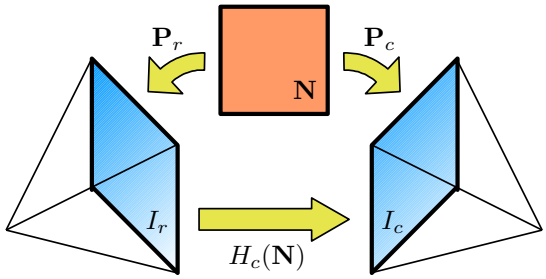


Figure 2. The 3D plane together with the image projection matrices define a homography H which maps image points from I_r to points in the image I_c.

Given the input plane defined by the initial position $\mathbf{p} \in \mathbb{R}^3$ and normal $\mathbf{n} \in \mathbb{R}^3$, a reference image, and a set of comparison images for the plane are defined by:

$$\begin{aligned} \mathbf{ref}(\mathbf{p}) &= I_r & r \in \{1,\dots,C\} \\ \mathbf{comp}(\mathbf{p}) &= \{I_{c_1},\dots,I_{c_k}\} & c_1,\dots c_k \in \{1,\dots,C\} \end{aligned}$$

where C is the number of views. The reference image can for example be chosen as the image where viewing direction and the vertex normal are closest to parallel. In all the presented steps of the tracking workflow, we have a good initial closed surface. Thus, the set of comparison images can be obtained by a simple visibility test using the OpenGL z-Buffer.

For simplicity we consider only one comparison image I_c in the following. Let the projection matrices for the reference image and the comparison image be

$$\mathbf{P}_r = [\mathbf{Q}_r|\mathbf{q}_r] \text{ and } \mathbf{P}_c = [\mathbf{Q}_c|\mathbf{q}_c]$$

Without loss of generality, we can transform the scene by a matrix $\mathbf{B}$ such that $\mathbf{P}'_r = \mathbf{P}_r\mathbf{B} = [\mathbf{Id}_3|0]$. Together with its normal $\mathbf{n}$ we define a plane at point $\mathbf{p}$ as $\mathbf{N}^T = \left[\mathbf{n}^T, \delta\right]$, with $\delta = -\mathbf{p}\cdot\mathbf{n}$. This determines a homography

$$\begin{aligned} H_c(\mathbf{N}) &= \left(\delta\mathbf{Q}_c - \mathbf{q}_c\mathbf{n}^T\right)\left(\delta\mathbf{Q}_r - \mathbf{q}_r\mathbf{n}^T\right)^{-1} \\ &= \left(\delta\mathbf{Q}'_c - \mathbf{q}'_c\mathbf{n}^T\right) \end{aligned}$$

which maps pixels $\hat{\mathbf{p}} \in \mathbb{R}^2$ from the reference image to the comparison image (Figure 2). The objective is to find new plane parameters which minimize the energy function

$$E_c(\mathbf{N}) = \sum_{\hat{\mathbf{p}}\in\Omega} \left(I_r(\hat{\mathbf{p}}) - I_c(H_c(\mathbf{N})\hat{\mathbf{p}})\right)^2$$

where Ω is a square region in the reference image around the projected vertex $\mathbf{P}_r\mathbf{p}$. Notice that the final energy takes all comparison images into account and can be expressed as $E(\mathbf{N}) = \sum_c E_c$. We set Ω to 15×15 pixels in all our experiments. Using a Taylor expansion of the intensity function $I_c(H_c(\mathbf{N})\hat{\mathbf{p}})$, we linearize the gradient of E and use Newton's method to solve for the optimal plane equation. For stability reasons we subtract the average pixel intensity from all pixels within the region. For more details on the Taylor expansion and how to set up the linear system for each step in Newton's method, please see [10].

After minimizing the energy function, the 3D position can be obtained by shooting a ray through the center of Ω and computing the intersection with the optimized plane. Occasionally, due to noise in the images or badly textured parts in human faces, this process does not succeed at every vertex. If the plane equation is numerically ill-conditioned or if the vertex was only visible in less than 2 cameras, we discard the result. We also use a histogram based discarding criterion: If the final error, is among the 20% largest errors we discard the Surfel.

3. Workflow to Reconstruct a Dynamic Face

In order to reconstruct the dynamics of a face, the first step is to fit the face template to the individual geometry of the first frame. Then, the *2D mesh tracking* establishes temporal correspondence between pixels of successive frames by tracking image-samples distributed over regions of the first frame. Finally, 3D reconstructions of these image-samples are used as handles to deform the face template and thereby capture the movement.

3.1. Initialization of the Face Template

To reconstruct a human face by just using *Surfel Fitting* would produce a point cloud of arbitrary size probably containing holes and outliers. We overcome this by using a morphable model as a shape prior. Our face template contains a fixed number of vertices ($\sim 8K$) and is in a one-to-one correspondence with the vertices of the morphable model. The basic appearance of a neutral face can be changed by adjusting some shape parameters. We initialize the morphable model by fitting it to a set of user defined points. The model is then automatically refined using the correspondences generated by *Surfel Fitting*.

Morphable model. The morphable model we use is similar to the one introduced by Blanz and Vetter [3]. To generate the model, we scanned about 50 faces in a neutral expression and established correspondences, similar to [3]. By adjusting shape parameters α_j we can approximate each face of the database as a weighted sum of eigenfaces $\mathbf{m}_j$ added to an average face $\bar{\mathbf{M}}$

$$\mathbf{M} = \bar{\mathbf{M}} + \sum_{j=1}^{k} \alpha_j \cdot \mathbf{m}_j$$

The small number of k eigenfaces are extracted by performing PCA on the laser scanned face data. If the database contains K faces, PCA extracts $K-1$ eigenfaces, describing the main deviations from the average face. By excluding eigenfaces with small eigenvalues, the dimensionality of the face space is reduced to a small number $k < K$, while keeping the important details. To neglect high frequencies and to obtain smooth surfaces, we set $k = 15$ in all our experiments.

Initial transformation and shape. For a rough estimate of the rigid translation and rotation w.r.t. the coordinate system of the cameras as well as the shape parameters, we use a few user defined points like the corners of eyes and lips. Defining these features in at least two views allows us to triangulate the 3D location of those features. We assume the user defined a very sparse set of L feature points denoted as $\mathbf{U} = [u_{i_1}, \ldots, u_{i_L}]^T$. Then the corresponding points of the morphable model are

$$\begin{aligned} \mathbf{M} &= [M_{i_1}, \ldots, M_{i_L}]^T \\ &= [\bar{M}_{i_1}, \ldots, \bar{M}_{i_L}]^T + \sum_{j=1}^{K} \alpha_j \cdot [m_{j_{i_1}}, \ldots, m_{j_{i_L}}]^T \end{aligned}$$

We alternately optimize the rigid transformation and shape parameters of the morphable model. To compute the model's translation and rotation, the method of Iterative Closest Points [12] is used which minimizes the squared distances between user defined points and corresponding model points. In what follows, $\mathbf{M}$ denotes the rigidly transformed morphable model.

After optimizing the rigid transformation, the new shape parameters can be obtained by minimizing the function

$$E = E_{Shape} + \lambda E_{Ave} = (\mathbf{M} - \mathbf{U})^2 + \lambda \sum_{j=1}^{K} \frac{\alpha_j^2}{\sigma_j^2}$$

where σ_j denotes the eigenvalue of the eigenface $\mathbf{m}_j$. The second term with weight λ has a smoothing effect, because faces near the average face have a smaller energy value. Deriving this function w.r.t. the shape parameters α yields a system linear in α. In each iteration λ is lowered proportional to the number of iterations, such that the morphable model slowly approaches the user defined points.

Improvement of transformation and shape. In the next step, we run the *Surfel Fitting* algorithm to calculate new depth values for the vertices of the face model. To do this for each vertex, a reference image is defined as that image with viewing direction most parallel to the vertex normal. The comparison images are obtained from a visibility

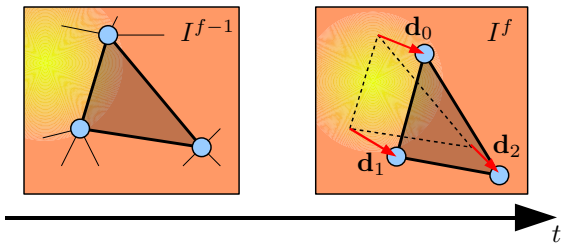


Figure 3. Displacing the triangle vertices by $\mathbf{d}_0, \mathbf{d}_1$ and $\mathbf{d}_2$ yield a similar intensity distribution within the triangle for both successive images I^{f-1} and I^f.

test. The resulting point cloud, possibly containing some holes, is used to augment the user defined feature points U. For each new point $u \in U$, a corresponding point is obtained as the point on the face model with minimal distance to u. This results in new pairs $(\mathbf{M}, \mathbf{U})$, which are used to compute new parameters for shape and rigid transformation, as described above.

3.2. Mesh Tracking in 2D

The objective of the *2D mesh tracking* is to establish temporal correspondences between successive frames. In our setup we use five video cameras to track the dynamic facial expression through time. Each view is tracked individually. To do this, we place a 2D mesh in the first image and calculate displacements for every frame such that the mesh tracks the 2D deformation. Since we can control for global smoothness, foldovers can be prevented.

Initialization. We project the face template, reconstructed as described in Section 3.1, into the first frame of the considered view. The projected mesh is then remeshed by the algorithm presented in [4], such that the new average edge length covers about 25 pixel. We denote the remeshed version of the mesh in the first frame as $\hat{\mathbf{S}}^1$.

Tracking. A view consists of a sequence of images

$$I^1, \ldots, I^{f-1}, I^f, I^{f+1}, \ldots$$

Given two successive images I^f and I^{f+1}, the aim is to find displacements $\mathbf{d}_i = [d_{i,x}, d_{i,y}]^T \in \mathbb{R}^2$ for every vertex of the given shape $\hat{\mathbf{S}}^f$, such that differences in the intensity distribution within each triangle of two successive images are small (see Figure 3). Consider one triangle T of $\hat{\mathbf{S}}^f$. A pixel $\hat{\mathbf{p}} = [x, y]^T \in \mathbb{R}^2$ within this triangle has barycentric coordinates $[\beta_0, \beta_1, \beta_2]$. The barycentric coordinates and the vertex displacements define a linear mapping $\pi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, which maps $\hat{\mathbf{p}}$ from the undeformed triangle of image I^f to a deformed triangle in the image I^{f+1}. If $I^f(\hat{\mathbf{p}})$ is the intensity function of an image I^f, the minimization function can be stated as

$$E_T = \sum_{\hat{\mathbf{p}} \in T} \left(I^f(\hat{\mathbf{p}}) - I^{f+1}(\pi(\hat{\mathbf{p}})) \right)^2$$

If the time between two successive frames is short, the input is already close to the optimal solution and a standard Levenberg-Marquard minimization procedure is suitable to solve for the displacements d_i. Summing these energy functions for all triangles yields the global energy function

$$E_{data} = \sum_{T \in \hat{\mathbf{S}}^f} E_T = \sum_{T \in \hat{\mathbf{S}}^f} \sum_{\hat{\mathbf{p}} \in T} \left(I^f(\hat{\mathbf{p}}) - I^{f+1}(\pi(\hat{\mathbf{p}})) \right)^2$$

To ensure a smooth distribution of the displacements we introduce the additional energy term

$$E_{smooth} = \sum_{i \in V(\hat{\mathbf{S}}^f)} \left(\frac{1}{\omega_i} \sum_{j \in \mathbf{Neigh}_i} \omega_{i,j} \parallel \mathbf{d}_i - \mathbf{d}_j \parallel \right)^2$$

where $V(\mathbf{S}^f)$ is the set of vertex indices of the mesh $\hat{\mathbf{S}}^f$ and $\mathbf{Neigh}_i$ denotes the 1-neighborhood of vertex $\hat{\mathbf{p}}_i$. The standard chordal weights $\omega_{i,j} = \parallel \hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j \parallel^2$, $w_i = \sum_{j \in N_i} \omega_{i,j}$ are used to set up the Laplace system. Putting both terms together, the final energy function is denoted as

$$E = E_{data} + \lambda E_{smooth}$$

where λ controls the smoothness term.

3.3. Mesh Tracking in 3D

As initialization for the *3D mesh tracking* we use again the surface we estimated for the first frame. The objective of the algorithm described in this section is to find a deformation of the face template for every frame, such that the highly detailed movements of the captured face are tracked by the template face. To achieve this, we generate image-samples in every view and track them in time. Using the *Surfel Fitting* of Section 2, these image-samples are reconstructed in 3D for every frame. Finally, these reconstructed 3D points are used to deform the template mesh. As stated above this increases the robustness of *Surfel Fitting* and decouples the reconstruction of the dynamics from the independent reconstruction of single image-samples.

Generating image-samples. Running the *2D mesh tracking* described in Section 3.2 on every view of the video sequence produces a 2D triangle mesh $\hat{\mathbf{S}}_c^f$ for each view c and frame f. Supersampling the triangles of the meshes in the first frame of the sequence generates 2D points that have barycentric coordinates w.r.t. the triangle they are placed in. For every view c this yields a set of points $\hat{\mathbf{p}}_{i,c}^1$ for the first frame, where a point can uniquely be identified by its index i and the view c it was put in. The number of image-samples is a user-defined parameter. We usually place 1600 samples in one view to obtain a dense reconstruction. The mapping π, which is defined by the deformation of a 2D mesh from one frame to the following, allows us to displace the image-samples and thereby track them through the whole sequence of a single view. This produces sequences of points $\hat{\mathbf{p}}_{i,c}^f$.

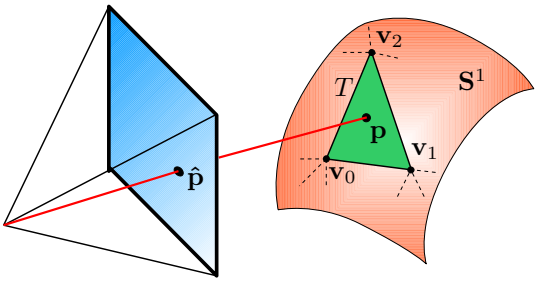


Figure 4. An anchor point of a image-sample $\hat{\mathbf{p}}$ is obtained by the intersection of a ray through $\hat{\mathbf{p}}$ with the initial face template $\mathbf{S}^1$.

From image-samples to 3D trajectories. Section 3.1 describes how to fit a face template to the first frame of a video stream. Assume the face template to be $\mathbf{S}^1$. For each image-sample $\hat{\mathbf{p}}^1_{i,c}$, we introduce an anchor point on the surface of $\mathbf{S}^1$ by shooting a ray through $\hat{\mathbf{p}}^1_{i,c}$ and determining the intersection with $\mathbf{S}_1$. This intersection is located within a triangle T and has barycentric coordinates $[\gamma_0, \gamma_1, \gamma_2]$ with respect to T (see Figure 4). An image-sample $\hat{\mathbf{p}}^f_{i,c}$ can be reconstructed in 3D by using the *Surfel Fitting* approach of Section 2. Given a face template $\mathbf{S}^{f-1}$ that was already fitted to frame $f-1$, a good initial solution for the Surfel position is obtained by evaluating the linear combination of the triangle vertices of $T \in \mathbf{S}^{f-1}$ weighted with $[\gamma_0, \gamma_1, \gamma_2]$. The normal of this triangle is also the initial plane normal for the Surfel. As a reference image, we select the view c that the image-sample was initially placed in. Since the fitted template of the frame $f-1$ is already a good approximation, this mesh is wellsuited for visibility tests to determine the set of (multiple) comparison images. If the *Surfel Fitting* succeeded, the reconstructed point $\mathbf{p}^f_{i,c} \in \mathbb{R}^3$ is stored in a list denoted by $\mathbf{Succ}(f)$.

Deformation of the template mesh. In order to deform the mesh we treat the $\mathbf{p}^f_{i,c}$ as handles which drag the surface $\mathbf{S}^1$. We define two objective functions. The first function measures the squared distance between the Laplace vectors of $\mathbf{S}^1$ and those of the deformed surface $\mathbf{S}^f$

$$E_L = \sum_{\mathbf{v} \in \mathbf{S}} \parallel \Delta \mathbf{v}^f - \Delta \mathbf{v}^1 \parallel^2$$

Here, Δ denotes the discrete Laplace operator using the cotangent weights evaluated on the surface $\mathbf{S}_1$. The second function penalizes large deviation of the anchor point from the reconstructed point and can be denoted as

$$E_C = \sum_{\mathbf{p} \in \mathbf{Succ(p)}} \parallel \mathbf{p} - \mathbf{anchor}(\mathbf{p}) \parallel^2$$

where the anchor point $\mathbf{anchor}(\mathbf{p}^f_{i,c})$ is calculated by interpolating the vertices of the triangle T associated with $\mathbf{p}^f_{i,c}$ using the precomputed barycentric coordinates:

$$\mathbf{anchor}(\mathbf{p}^f_{i,c}) = \sum_{\mathbf{v}_t \in T} \gamma_t \cdot \mathbf{v}^f_t$$

To obtain the new vertex positions of a mesh $\mathbf{S}^f$, we solve

$$E = E_L + \lambda E_C$$

in the least-squares sense and repeat the whole procedure for the next frame $f+1$.

It is worth mentioning that this procedure can also help improve the estimated surface $\mathbf{S}^1$ of the first frame. At the end of the process described in Section 3.1, a new point cloud can be extracted by *Surfel Fitting*. For each Surfel, we can compute an anchor point as the closest point on $\mathbf{S}^1$ w.r.t. the Surfel. These pairs can then be used to deform the face template $\mathbf{S}^1$, as described above. The deformed surface does not lie in the space spanned by the morphable model and is used as the input surface for all subsequent steps of the pipeline.

4. Results

We generated all our results using a 2.6Ghz Intel Core i7 CPU. During *2D mesh tracking*, the computation of the 2D displacements for five views of one frame took an average time of 52 seconds. The average time for the *Surfel Fitting* of one frame, where we optimized about 8K samples from all five views, was 50 seconds, leading to an overall computing time of less than 2 minutes per frame. We collected sequences of 5 different subjects each performing different facial expressions for a duration of approximately 2 to 4 seconds. In Figure 5 we present five examples. It shows one of the input images together with the reconstruction of the neutral face (left column). The middle column shows the result of the deformation step where the surfels act as handles to deform the neutral face (see the closeup images). Since we decided to use a predefined face template with a fixed mesh topology, further modelling steps, like *e.g.* the placement of eyes, can be performed automatically. The right column shows the same expression as in the middle with eyes and simple lids modeled as triangle meshes. In all these examples, we left the parameters at a fixed setting. We observed that the Surfel Fitting sometimes produces strong outliers if the Surfel is only visible by two cameras, which usually occurs at the face template's boundary. This problem could be solved by adding more cameras to the rig. In regions with large specular reflection, which normally lie on the nose or the cheek of a subject, *Surfel Fitting* does not produce reliable results. If such a Surfel was not discarded, it is possible that it has still a large influence in the modeling step, since we solve the system in the least squares sense. This might produce a wrongly reconstructed surface. In future work we would like to use RANSAC [8] methods to find outliers more robustly.

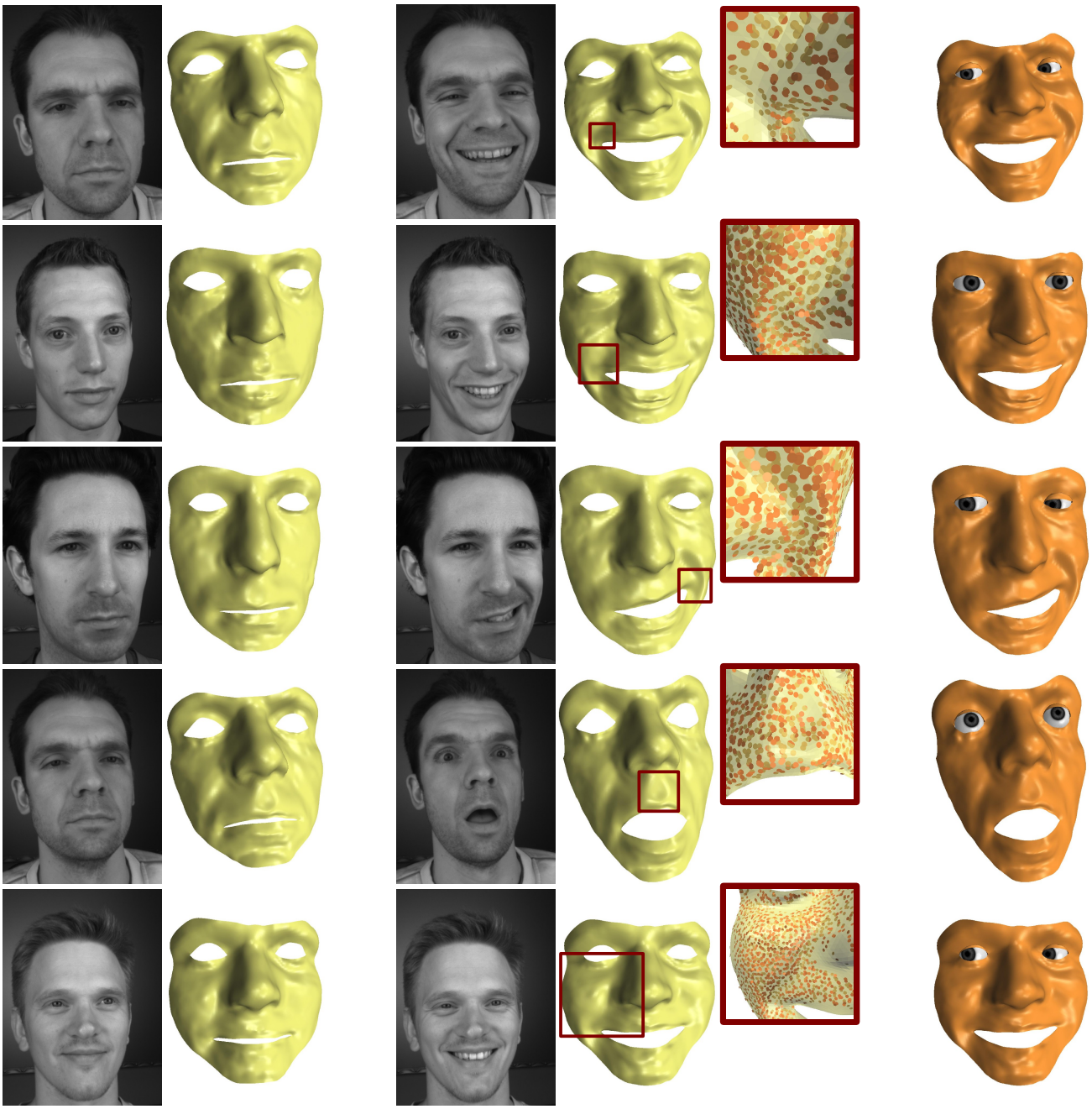


Figure 5. Reconstruction Results. The Left column shows input images of neutral faces and their reconstructions. The extracted Surfels and their anchor points define constraints in a modelling step to deform the surface. This is possible because we established temporal correspondences by using the *2D mesh tracking* algorithm. The deformation and the Surfels can be seen in the middle column. Since we use a predefined face template automatic placements of *e.g.* eyes is easy (right column).

5. Discussion and Conclusion

Standard 3D reconstruction approaches from sparse feature points are generally quite sensitive to noise. The reason for this is that in the energy function to be minimized only a small local image region is considered. We overcome this problem by using a simple morphable model of neutral faces to estimate the more global appearance of the face seen in the first image. This generates a surface similar to the one being reconstructed, which strongly increases the robustness of the *Surfel Fitting*.

In general a spatio-temporal correspondence can be obtained by tracking features between views and frames. We mainly had two problems using feature tracker like the KLT tracker introduced in [21] or deriving correspondences by using SIFT features proposed by Lowe [19]: First, the temporal tracking does not take the neighboring features into account. Because of that, it often produces trajectories which slide past each other. These foldovers inducing high distortions in the tracked face template. To correct this, we chose the proposed *2D mesh tracking* because we can control the global smoothness and prevent the features from sliding. Second, tracking features between views produces only a very sparse set of 3D features, which do not provide enough constraints for the modeling step to get reliable results. In our proposed method, we distribute a large set of (redundant) image-samples, so we are able to omit wrongly reconstructed image-samples but still end up with a large set of constraints for the modeling phase. Since each image-sample is considered independently, the 3D reconstruction is simple. Combining it with a simple modeling approach which fulfils each constraint in the least squares sense, a smooth surface can be produced and the robustness of the reconstruction method can be increased (visibility, good initial solutions), while simultaneously maintaining full correspondence between frames and subjects.

In this paper we introduced a system for markerless reconstruction of dynamic faces. Our system is able to establish inter-subject correspondence as well as temporal correspondence. We presented reasons why we combined different algorithms from the field of mesh modeling and computer vision and showed in numerous examples that our system performs well.

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, IRTG 1328).

References

[1] M. Ben-Chen, O. Weber, and C. Gotsman. Variational harmonic maps for space deformation. *ACM Trans. Graph.*, 2009.

[2] B. Bickel, M. Botsch, R. Angst, W. Matusik, M. Otaduy, H. Pfister, and M. Gross. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph.*, 26(3):33, 2007.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In A. Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 187–194, 1999.

[4] M. Botsch and L. Kobbelt. A remeshing approach to multiresolution modeling. In *Proc. SGP*, pages 185–192, 2004.

[5] C. Curio, M. Breidt, Q. C. V. M. Kleiner, M. A. Giese, and H. H. Bülthoff. Semantic 3d motion retargeting for facial animation. In *Applied perception in graphics and visualization*, pages 77–84, USA, 2006.

[6] M. Dellepiane, N. Pietroni, N. Tsingos, M. Asselot, and R. Scopigno. Reconstructing head models from photographs for individualized 3d-audio processing. *Proc. Pacific Graphics*, 27(7):1719–1727, 2008.

[7] P. Ekman and W. V. Friesen. Manual for the facial action coding system. *Consulting Psychology Press*, 1978.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 726–740, 1987.

[9] Y. Furukawa and J. Ponce. Dense 3d motion capture for human faces. In *Proc. CVPR*, 2009.

[10] M. Habbecke and L. Kobbelt. Iterative multi-view plane fitting. In *Vision, Modeling and Visualization*, 2006.

[11] C. Hernández, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Intl. Conf. on Comp. Vision*, 2007.

[12] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America. A*, 4:629–642, 1987.

[13] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26(3):71, 2007.

[14] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: generating animated head models with anatomical structure. In *Proc. SCA*, New York, USA, 2002.

[15] S. C. Koterba, S. Baker, I. Matthews, C. Hu, J. Xiao, J. Cohn, and T. Kanade. Multi-view aam fitting and camera calibration. In *Proc. ICCV*, volume 1, pages 511 – 518, 2005.

[16] Y. Lee, D. Terzopoulos, and K. Waters. Constructing physics-based facial models of individuals. In *In Proc. Graphics Interface*, pages 1–8, 1993.

[17] R. Li and S. Sclaroff. Multi-scale 3d scene flow from binocular stereo sequences. *Comput. Vis. Image Underst.*, 110(1):75–90, 2008.

[18] Y. Lipman, D. Levin, and D. Cohen-Or. Green coordinates. *ACM Trans. Graph.*, 27(3):1–10, 2008.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60, 2004.

[20] M. Odisio, M. Odisio, and G. Bailly. Shape and appearance models of talking faces for model-based tracking. In *Proc. AVSP*, pages 105–110, 2003.

[21] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[22] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.*, 24(3):417–425, 2005.

[23] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proc. ICCV*, volume 2, pages 722 – 729, September 1999.

[24] Waters and Keith. A muscle model for animation three-dimensional facial expression. In *Computer graphics and interactive techniques*, pages 17–24, USA, 1987.

[25] T. Weise, B. Leibe, and L. V. Gool. Fast 3d scanning with automatic motion compensation. In *Proc. CVPR*, June 2007.

[26] Williams and Lance. Performance-driven facial animation. In *Computer graphics and interactive techniques*, pages 235–242, USA, 1990.