3D Face Template Registration Using Normal Maps

Zhongjie Wang
MPI Informatik
Saarbrücken, Germany
zwang@mpi-inf.mpg.de

Martin Grochulla
MPI Informatik
Saarbrücken, Germany
mgrochul@mpi-inf.mpg.de

Thorsten Thormählen
Philipps-Universität Marburg
Marburg, Germany
thormae@informatik.uni-marburg.de

Hans-Peter Seidel
MPI Informatik
Saarbrücken, Germany
hpseidel@mpi-sb.mpg.de

***Abstract*—This paper presents a semi-automatic method to fit a template mesh to high-resolution normal data, which is generated using spherical gradient illuminations in a light stage. Template fitting is an important step to build a 3D morphable face model, which can be employed for image-based facial performance capturing. In contrast to existing 3D reconstruction approaches, we omit the structured light scanning step to obtain low-frequency 3D information and rely solely on normal data from multiple views. This reduces the acquisition time by over 50 percent. In our experiments the proposed algorithm is successfully applied to real faces of several subjects. Experiments with synthetic data show that the fitted face template can closely resemble the ground truth geometry.**

***Keywords*-normal maps; face registration;**

I. Introduction

3D face models are widely used for computer graphics and computer vision applications. Of particular interest are morphable 3D face models that are based on a single deforming 3D template mesh, which can represent different individuals or different facial expressions. The template deformation is typically controlled via a small set of parameters. Examples are hand-crafted blend shapes [1] or learned morphable face models [2], which are based on the analysis of a large database of 3D laser scans. In order to generate a morphable face model, a template mesh has to be registered to all 3D laser scans in the database. After registration a template vertex with a particular index in the template mesh is located at the same semantically corresponding point in all 3D scans (e.g., vertex no. 101 is always the tip of the nose, vertex no. 512 is the corner of the right eye, and so on). The registration is a crucial step in the generation of a morphable face model because once corresponding vertex positions are established in all the database exemplars, it is already possible to perform a linear blend between the exemplars to generate new individuals or interpolated facial expressions. To reduce the number of blending weights, typically a Principle Component Analysis (PCA) is performed, which generates a low dimensional parameter space that can still represent the observed differences in the vertex positions of the exemplars [2].

Using a so called *light stage* (a sphere with a large number of individually controllable light sources) a very detailed 3D scan of a human face can be captured [3]. Here, the low resolution 3D geometry is typically acquired using a structured light approach, and the fine details are captured via normal map generation. The normal maps are created by taking images under 4 to 7 different illumination conditions that are generated with the light stage. In addition, a projector is used to generate a series of stripe patterns (typically 5 to 15 patterns of increasing frequency) for the structured light reconstruction. Though projector and light stage patterns could be in theory displayed at fast succession, high frame rates are difficult to achieve in practice due to frame rate limitations of the camera as well as switching time limitations of the light stage and the projector. Consequently, the captured subject should not move during the acquisition, which is quite challenging, especially for less relaxed facial expressions.

In order to reduce the capturing time and effort, we propose in this paper a method to register a 3D face template only to the normal maps. The omission of structured light scanning reduces the capturing time by almost 50 percent. We claim that it is possible to skip structured light scanning because the low resolution 3D geometry is already approximately given by the initial 3D face template. However, in our experiments we found that a normal map from a single camera view can not resolve the depth ambiguities. Consequently, our approach uses multiple normal maps that are generated simultaneously by observing the face with multiple cameras, which does not increase the capturing time.

State-of-the-art approaches [4], [3], [2] use non-rigid ICP algorithms to fit a 3D template mesh to point cloud data (that is obtained via laser or structured light scanning). In our set-up this non-rigid ICP algorithm is replaced by an algorithm that registers a 3D face template to several normal maps. Thereby, the proposed registration approach performs three steps. First, some manually selected feature points and their projections on the normal maps are registered to roughly align the template. Second, a normal registration method is applied to align the template semantically to the normal maps. This step aims to find the correlation between the template geometry and the geometry information encoded in the normal maps. The result of this step is a deformed template mesh that better resembles the geometry of the real subject, but still maintains its basic structure. Third, to

further refine the shape of the template, a shape refinement is executed. In this step, we employ the constraint that for a given 3D position, its projections in neighbouring views should have the same normals.

The proposed algorithm resides on the following core contributions:

- A novel method to semi-automatically fit a 3D face template to normal maps. This includes three main steps: feature point registration, normal registration, and shape refinement.
- In normal registration, a novel optimization strategy to minimize a highly non-linear function is proposed. It splits the problem to several constrained optimization steps which can be linearised and solved efficiently.

As structured-light scanning is omitted, the acquisition time can be reduced by over 50%, while the fitting result is still accurate.

The rest of the paper is organized as follows. The next section reviews related work in the area of template fitting and 3D reconstruction from normal maps. In Section III, the employed hardware set-up for normal map generation is introduced. The proposed template registration algorithm is described in Section IV. In Section V, several experiments are presented to evaluate our algorithm. The paper ends with concluding remarks and future works in Section VI.

II. Related work

In the past few years, many algorithms for recovering 3D facial geometry have been proposed. This section gives an overview of the most related work to our method.

Marker-based Performance Capture: Marker-based performance capture systems are still the most widely adopted solutions for facial performance capture in the industry and have achieved great success in the commercial world. As markers at certain semantically significant locations are attached to the face, it is relatively easy to fit a template mesh to the marker data. The advantage of this technique is that it is fast, robust to noise, and easy to deploy. However, the captured detail is quite low, as measurements are only available at the marker positions.

Photometric Stereo: Photometric stereo [5] is an active illumination approach used for surface normals recovery. The normals provide derivative information of the 3D surface and can be used to generate accurate high-frequency local geometry [6]. Recent light stage developments [3], [7] adopt a spherical gradient illumination approach for generating a detailed normal map of the input face. Real-time computation [8] for this approach has been achieved using high-speed video cameras and a fast GPU implementation. Ma et al. [9] applied structured light scanning to capture low-frequency 3D structure, so their hardware system has to be well designed to allow capturing a large number of images in fast succession (13 images = 8 spherical illuminations + 5 structure light patterns per time instance). In contrast, our approach only requires 6 images per time instance for normal map calculation, thus can greatly increase the frame rate. Many extensions have appeared afterwards [10], [11], most related to our work is Wilson et al.'s approach [12], which made two improvements to Ma et al.'s work by firstly reducing the requirements of illumination condition, and secondly exploring dense temporal stereo correspondence for 3D structure reconstruction rather than structured light scanning. With the benefits of these improvements, their system can achieve higher frame rates and also more stable results. However, the aim of these approaches is to generate detailed 3D geometry for every captured frame rather than fitting a consistent template mesh.

3D morphable models: 3D morphable model based approaches [2], [13], [14] can provide useful prior knowledge for marker-based or image-based facial performance capturing. General facial models [15], [16], which are trained on a large database, may miss fine details unique to a specific person, hence, recent developments also focus on subject specific models [17], or single patch representation in region based variants [18], [19]. To build a 3D face model, these approaches require semi-automatic fitting of templates to 3D scanner data with manually selected markers, while our automatic approach relies solely on normal data.

III. Hardware set-up and normal map generation

The employed data capturing system is shown in Fig. 1. It comprises of a light stage consisting of 156 LEDs arranged on a spherical metal frame and six digital cameras. The light stage is used to produce six axis parallel spherical gradient illuminations. The set of images captured by the c-th camera is denoted as $\mathcal{L}_c = \left\{L_c^x, L_c^{-x}, L_c^y, L_c^{-y}, L_c^z, L_c^{-z}\right\}$. For the spherical gradient illumination the intensity values of the LEDs are translated and shifted to the range $[0, 1]$, since negative light cannot be emitted. The normal map of the c-th camera can be calculated using the spherical gradient illuminations in a pixel-wise manner (as proposed in [12]):

$$N_c = \frac{\left(L_c^x - L_c^{-x}, L_c^y - L_c^{-y}, L_c^z - L_c^{-z}\right)^\top}{\left|\left|\left(L_c^x - L_c^{-x}, L_c^y - L_c^{-y}, L_c^z - L_c^{-z}\right)^\top\right|\right|} \quad . \quad (1)$$

The digital cameras are calibrated with a calibration pattern. The calibration pattern is placed inside the light stage in an axis-aligned way such that its center coincides with the center of the light stage. This assures that the cameras are calibrated with respect to the light stage coordinate system. During camera calibration we employ the focal length given by the EXIF data provided by the camera and estimate the extrinsic camera parameters with Tsai's approach [20]. Then a bundle adjustment is performed to further refine the extrinsic camera parameters. The size of the normal maps used in our experiments is 2592 $\times$ 1728.

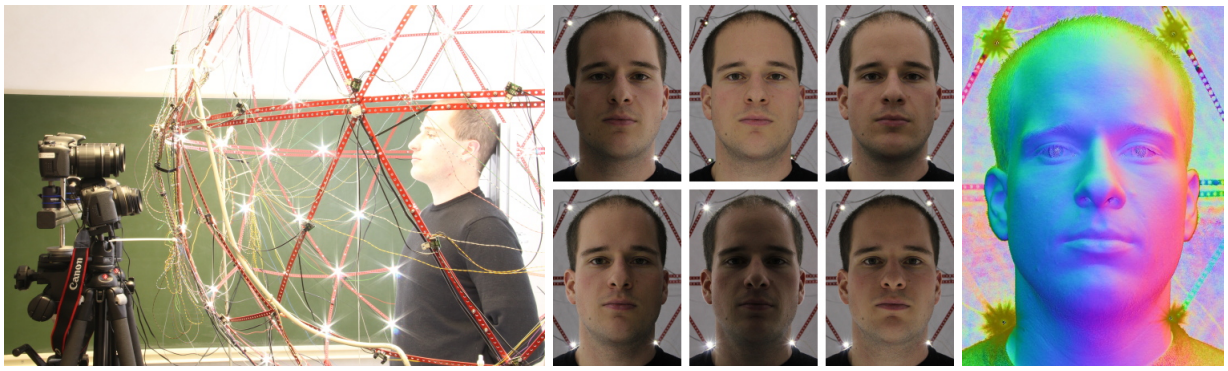

Figure 1. (from left to right) The employed data capturing system comprising a light stage, which can produce different illumination patterns, and several digital SLR cameras; six spherical gradient illumination patterns; normal map computed from the images of the six gradient illumination patterns.

IV. Algorithm

The proposed algorithm aims to register a mesh template to multi-view normal maps. The input consists of a face template, given as a polygonal mesh $\mathcal{S} = \{\mathcal{V}, \mathcal{E}\}$ which is defined by a set $\mathcal{V}$ of vertices $\mathbf{V}_i$ and a set of edges $\mathcal{E}$. Also, the set-up described in Section III provides the camera perspective projection matrices $\mathtt{P}_c$, and normal maps N_c for each camera (with index c) of the camera set $\mathcal{C}$. The output is a deformed template face which is fitted to the observed face.

The algorithm has three steps. Firstly, to roughly align the template and the normal maps, we manually select eight 3D feature points for the template and their projections for all the views. We register these eight points to the normal maps, and the rest of the template deforms smoothly. Secondly, ignoring the neighbour-view consistency, a normal registration method is executed to align the vertices of the template to their semantically correct positions in all the normal maps. Thirdly, through a multi-view refinement, the shape of the face is further improved by enforcing neighbouring-view consistency.

These three steps of the algorithm are described in the following three subsections in detail.

A. Feature-based registration

This step aims to match eight 3D feature points of the template to a set of user-defined 2D locations, while the rest of the vertices should deform smoothly. In our experiments we used the eight feature points visualized in Fig. 2. The problem is solved as a non-rigid registration problem.

We assign each vertex of the template $\mathbf{V}_i = (v_x, v_y, v_z)^\top$ a translation vector $\mathbf{T}_i = (t_x, t_y, t_z)^\top$. To enforce that the back-projection of a translated vertex is located at a user-defined 2D location $\mathbf{u} = (u_x, u_y)^\top$ in the normal maps, an energy term is defined by

$$E_{corner} = \sum_{c \in \mathcal{C}} \sum_{\mathbf{u}_{ic} \in \mathcal{U}} ||\mathbf{u}_{ic} - \mathtt{P}_c\left(\mathbf{V}_i + \mathbf{T}_i\right)||_2^2 \quad , \qquad (2)$$

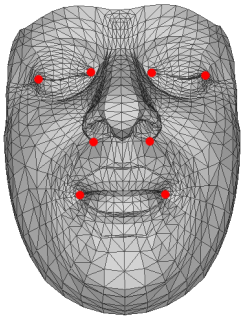

Figure 2. The template used in our experiments. It has 1250 vertices in total. The red points indicate the 3D feature points.

where $\mathtt{P}_c\{\cdot\}$ is a projection function which projects a 3D point to a 2D location in the image plane of the c-th camera; a set $\mathcal{U}$ includes all the user-defined 2D locations $\mathbf{u}$. To make the rest of the vertices deform smoothly, we constrain the translations of two connected vertices in the template mesh to be similar. The smoothness term is given by

$$E_{smooth} = \sum_{(i,j) \in \mathcal{E}} ||\mathbf{T}_i - \mathbf{T}_j||_2^2 \quad . \qquad (3)$$

Combining Eq. (2) with Eq. (3), the total cost can be written as

$$E = E_{corner} + \lambda E_{smooth} \qquad (4)$$

where λ is a weighting factor. The result is obtained by minimizing Eq. (4) using the non-linear optimizer.

Non-linear optimization: Since projection from 3D space to 2D space is non-linear, Eq. (4) becomes a non-linear optimization problem. We solve it as a non-linear least squares problem iteratively. The projection process is given as

$$m \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \bar{u} \\ \bar{v} \\ m \end{pmatrix} = \mathtt{P}_c\left(\mathbf{V}_i + \mathbf{T}_i\right) \quad , \qquad (5)$$

where u and v are the coordinate elements of a 2D location. To linearise this process, we can calculate the derivatives of

a 2D coordinate, i.e. $\frac{\partial u}{\partial \mathbf{T}_i}$ and $\frac{\partial v}{\partial \mathbf{T}_i}$, analytically. Then the Jacobian $\mathtt{J}$ of Eq. (2) is the concatenation of

$$\mathtt{J}_i = \begin{pmatrix} \cdots - \dfrac{\partial u}{\partial \mathbf{T}_i} & - \dfrac{\partial v}{\partial \mathbf{T}_i} \cdots \end{pmatrix} \tag{6}$$

for each vertex. The Jacobian of Eq. (3) can be written as

$$\mathtt{H} = \mathtt{D} \otimes \mathtt{I}_{3\times 3} \quad , \tag{7}$$

where $\mathtt{D}$ is the node-arc incidence matrix [21], and $\otimes$ is the Kronecker product operator. By combining Eq. (6) and Eq. (7), the final over-determined linearised equation system can be written as

$$\underbrace{\begin{pmatrix} \mathtt{J} \\ \lambda\, \mathtt{H} \end{pmatrix}}_{\mathtt{B}} \Delta\mathbf{T} = \underbrace{-\begin{pmatrix} \mathbf{r} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{b}} \quad , \tag{8}$$

where the residual vector $\mathbf{r}$ is the concatenation of $\mathbf{r}_i = \mathbf{u}_{ic} - \mathtt{P}_c\left(\mathbf{V}_i\right)$ for each vertex. The resulting normal equation is given as

$$\mathtt{B}^\top \mathtt{B}\, \Delta\mathbf{T} = \mathtt{B}^\top \mathbf{b} \quad . \tag{9}$$

Since the coefficient matrix in Eq. (9) is large and sparse, it can be solved efficiently using the iterative conjugate gradient method.

B. Normal registration

After feature-based registration, the template is roughly aligned. The purpose of normal registration is to semantically register the template to the normal maps. The constraint is that the projections of the normals of the template should be equal to the normals in the normal maps.

The main difference between color images and normal maps is that normal maps encode the geometric information, e.g. convex and concave positions, as a normal distribution. This means a surface can be recovered using normal integration from one normal map only, while for surface reconstruction multiple color images are needed (approximation techniques such as shape from shading can recover a surface from only one color image, but it also needs to approximate a normal map from the image). In order to overcome the influence of occlusions and also the errors in normal maps, an approach based on multi-view normal registration is proposed.

Since the three elements of a normal $\mathbf{n} = \left(n_x, n_y, n_z\right)^\top$ are correlated, e.g. $||\mathbf{n}||_2 = 1$, first we re-parametrize a normal to spherical coordinates, so that we can use two independent angular components θ and ϕ to represent a normal. Then the cost for the normal similarity is defined as:

$$E_{normal} = \sum_{c\in\mathcal{C}} \sum_{i\in\bar{\mathcal{V}}_c} \sum_{p\in\{\theta,\phi\}} \sum_{\mathbf{w}\in\mathcal{W}} \left(r_{cip\mathbf{w}}\right)^2 \quad , \tag{10}$$

where $\bar{\mathcal{V}}_c$ is the set of vertices which are not occluded in the view of camera c, p is the element index of a normal in spherical coordinates, $\mathcal{W}$ is a square window. Here, $r_{cip\mathbf{w}}$ is defined as

$$r_{cip\mathbf{w}} = \mathtt{G}\left(\mathbf{w}\right) \cdot N_c\left(\mathtt{P}_c\left(\mathbf{V}_i + \mathbf{T}_i\right) + \mathbf{w}\right)_p - \tilde{N}\left(\mathbf{V}_i + \mathbf{T}_i\right)_p \quad , \tag{11}$$

where $\mathtt{G}$ is a Gaussian kernel function defined on the window $\mathcal{W}$, and $N_c(\cdot)$ is a function which takes a pixel position as the input and returns the normal at the given pixel position of the c-th normal map.

Considering that the semantics of the template changes when some non-smooth deformations are applied, a smoothness term which not only allows a big range of deformation but also maintains the basic structure of the template is needed. In this paper, instead of using absolute translations, i.e. Eq. (3), to ensure the mesh rigidity, we employ mesh differentials as used in [22] to define the smoothness term. It can be written as

$$E_{smooth} = \sum_{\mathbf{V}_i\in\bar{\mathcal{V}}} ||\mathbf{V}_i + \mathbf{T}_i - \frac{1}{k} \sum_{\mathbf{V}_j\in\mathcal{N}\{\mathbf{V}_i\}} \left(\mathbf{V}_j + \mathbf{T}_j\right)||_2^2 \quad , \tag{12}$$

where $\mathcal{N}\{\cdot\}$ represents a set of neighbouring vertices.

Combining with the cost of the feature points in Eq. (2), the result can be obtained by minimizing the cost function

$$E = E_{normal} + \alpha E_{smooth} + \beta E_{corner} \quad , \tag{13}$$

where α and β are weighting factors.

Non-linear optimization: In Eq. (11), $N_c\left(\cdot\right)$ given a 2D image location returns the normal from the normal map of view c. Since the vertex' normal can be changed by deforming its neighbouring vertices, the representation of a vertex normal by using the positions of its neighbouring vertices, i.e. $\tilde{N}\left(\cdot\right)$, is a non-linear function. Concluded from that, Eq. (13) is a highly non-linear function. In particular, we find that the minimization of Eq. (13) cannot be solved by direct linearisation. In this paper, we address this problem by separating the whole optimization to several constrained optimization steps. These steps are:

1) Update the normals of the template using the current structure of the template, and fixate the normals.
2) Optimize Eq. (13) iteratively until the largest translation of all the vertices is smaller than a threshold.
3) If no further optimization of Eq. (13) is achieved, then finish, otherwise go back to step 1.

In step 2, when we fixate the normals of the template, Eq. (11) becomes

$$r_{cip\mathbf{w}} = \mathtt{G}\left(\mathbf{w}\right) \cdot N_c\left(\mathtt{P}_c\left(\mathbf{V}_i + \mathbf{T}_i\right) + \mathbf{w}\right)_p - \mathbf{m}_{ip} \tag{14}$$

where $\mathbf{m}_i$ represents the current fixated normal of the i-th vertex. We solve Eq. (14) by linearization. The Jacobian of Eq. (14) can be calculated as

$$\begin{aligned} \mathtt{R}_{cip\mathbf{w}} &= \mathtt{G}\left(\mathbf{w}\right) \frac{\partial N_c\left(\mathtt{P}_c\left(\mathbf{V}_i + \mathbf{T}_i\right) + \mathbf{w}\right)_p}{\partial \mathbf{T}_i} \\ &= \mathtt{G}\left(\mathbf{w}\right) \left(\frac{\partial N_{cp}}{\partial \mathbf{n}_x} \frac{\partial \mathbf{n}_x}{\partial \mathbf{T}_i} + \frac{\partial N_{cp}}{\partial \mathbf{n}_y} \frac{\partial \mathbf{n}_y}{\partial \mathbf{T}_i} \right) \end{aligned} \quad , \tag{15}$$

where N_{cp} is an abbreviation of $N_c\left(\mathtt{P}_c\left(\mathbf{V}_i+\mathbf{T}_i\right)+\mathbf{w}\right)_p$, and $\mathbf{n}=\left\{\mathbf{n}_x,\mathbf{n}_y\right\}^\top$ is a 2D image location.

In Eq. (15), $\frac{\partial \mathbf{n}_x}{\partial \mathbf{T}_i}$ and $\frac{\partial \mathbf{n}_y}{\partial \mathbf{T}_i}$ can be calculated analytically, and $\frac{\partial N_{cp}}{\partial \mathbf{n}_x}$ and $\frac{\partial N_{cp}}{\partial \mathbf{n}_y}$ are the normal gradients in image domain. Since we re-parametrize the normal representation to spherical coordinates, the two angles θ and ϕ are independent, so that we can interpolate a normal by performing a finite difference operation.

The Jacobian of Eq. (12) can be written as a constant matrix $\mathtt{Q}$ defined as follows:

$$\mathtt{Q}_{ij}=\begin{cases}1, & i=j\\ -\frac{1}{k_i}, & \mathbf{V}_j\in\mathcal{N}\left(\mathbf{V}_i\right)\\ 0, & \text{else}\end{cases}\quad, \tag{16}$$

where k_i is the number of the neighbouring vertices of $\mathbf{V}_i$.

The final linear equation system of Eq. (13) can be written as

$$\underbrace{\begin{pmatrix}\mathtt{R}\\ \alpha\,\mathtt{Q}\\ \beta\,\mathtt{J}\end{pmatrix}}_{\mathtt{B}}\Delta\mathbf{T}=-\underbrace{\begin{pmatrix}\mathbf{h}\\ \alpha\,\mathbf{s}\\ \beta\,\mathbf{r}\end{pmatrix}}_{\mathbf{b}}\quad, \tag{17}$$

where $\mathtt{R}$ is the concatenations of Eq. (15), $\mathtt{Q}$ is defined as in Eq. (16), $\mathbf{h}$ is the normal residual defined as the concatenation of

$$\mathbf{h}_{cip}=\sum_{\mathbf{w}\in\mathcal{W}}\mathtt{G}_{\mathbf{w}}\cdot N_c\left(\mathtt{P}_c\left(\mathbf{V}_i\right)+\mathbf{w}\right)_p-\mathbf{m}_{ip}\quad,$$

$\mathbf{s}$ is the smoothness term residual defined as the concatenation of

$$\mathbf{s}_i=\mathbf{V}_i-\frac{1}{k}\sum_{\mathbf{V}_j\in\mathcal{N}\{\mathbf{V}_i\}}\left(\mathbf{V}_j\right)\quad,$$

$\mathtt{J}$ and $\mathbf{r}$ are the correlated feature point terms defined in Eq. (8). As in Eq. (9), the resulting normal equation can be solved efficiently by the conjugate gradient method.

C. Multi-view Refinement

After normal registration, the position of the vertices of the template are much closer to their semantically correct positions in the normal maps. However, a shape refinement is needed to further correct the resulting surface for two reasons: First, the smoothness term that was employed in normal registration maintains the basic structure of the template, and second, in order to make the optimization solvable, we use several constrained optimization steps to approximate the original problem formulation. In this refinement step, we make use of the neighboring view consistency information to enforce that the projections of a vertex to a pair of neighboring camera views should have the same normal in both normal maps. Since the shape refinement only refines the 3D shape but does not have any concept of semantics, a good initial result of normal registration from the previous step is required. A good result means that, after normal registration, vertices are moved very close to their semantically correct positions and the back-projections are consistent in all views.

We formulate the optimization problem as

$$\begin{aligned}\underset{\mathbf{t}_i}{\operatorname{argmin}}\quad & \sum_{(s,t)\in\mathcal{M}}\sum_{i\in\bar{\mathcal{V}}_s\cap\bar{\mathcal{V}}_t}\sum_{p\in\{\theta,\phi\}}\sum_{\mathbf{w}\in\mathcal{W}}\left(a_{stip\mathbf{w}}\right)^2\\ & +\sigma\sum_{(i,j)\in\mathcal{E}}\left\|\mathbf{T}_i-\mathbf{T}_j\right\|_2^2\\ & +\tau\sum_{c\in\mathcal{C}}\sum_{\mathbf{u}_{ic}\in\mathcal{U}}\left|\left|\mathbf{u}_{ic}-\mathtt{P}_c\left(\mathbf{V}_i+\mathbf{T}_i\right)\right|\right|_2^2\end{aligned} \tag{18}$$

where

$$\begin{aligned}a_{stip\mathbf{w}} &= N_s\left(\mathtt{P}_s\left(\mathbf{v}_i+\mathbf{t}_i\right)+\mathbf{w}\right)_p\\ &- N_t\left(\mathtt{P}_t\left(\mathbf{v}_i+\mathbf{t}_i\right)+\mathbf{w}\right)_p\end{aligned} \tag{19}$$

is the difference of the two normals in normal map N_s and N_t, $\mathcal{M}$ is a set which consists of all the available camera pairings, σ and τ are weighting factors. The smoothness term and feature data term in Eq. (18) are the same as the ones in the feature-based registration in Eq. (3) and Eq. (2). This optimization problem can also be solved by linearisation. The Jacobian of the two parts of Eq. (19) are calculated as in Eq. (15), and the final normal equations are similar to Eq. (9).

Multi-resolution: Both in the normal registration and multi-view refinement, a multi-resolution approach is employed to improve efficiency and accuracy. In the experiment, we use three layers with different resolutions. Since the three elements of a normal in Cartesian coordinates are correlated, we first convert the normal representation from Cartesian coordinates to spherical coordinates, and employ a Gaussian kernel in this domain to blur the images. While processing each layer, the weighting parameters of the cost function are fixated.

V. Results

Our template fitting approach is evaluated with synthetic data as well as real data. Both types of experiments are executed with the same camera setup. We use only 6 cameras to cover the frontal face, and the face template shown in Fig. 2 in all experiments.

In the synthetic data experiment, we use a 3D head model to represent the human head, and render it in 3D Max to generate normal maps. The size of the model is similar to the size of a real human head. Fig. 4 shows the result after each optimization step. We can see that after feature-based registration, the shape is still dissimilar to the input head model. After normal registration, the face is deformed closer to the head model but still keeping the basic shape of the face template. This step is performed to ensure that each semantic position, i.e. concave and convex positions, is moved closer to the corresponding position of the head model. The next

step refines the shape of the face. To compare the final result with the synthetic model, we show the comparison with several horizontal and vertical slices in Fig. 3. Since the mesh of the face template is sparser than the synthetic model, the two contours cannot be perfectly matched. We can see that larger errors appear only at the boundary of the face. That is because the camera matrix only covers the frontal area of the face. When computing the Hausdorff distance between the model and our result, the mean error is 1.65mm and the root mean squared error(RMSE) is 4.62mm.

In the real data experiment, we evaluate our algorithm with three subjects. The result is shown in Fig. 5. For each subject, we use a neutral face and a face with an expression. Since eyes and mouth have very complicated structures, in all experiments, we ask the subjects to close their eyes and mouth. We can see from the result that the normal maps have high-resolution, but our template is comparatively sparse. Consequently, our algorithm can only recover the most significant features of a face, some subtle features such as skin foldings are not captured. All computations are performed on a single consumer-level computer, and the running time of the complete algorithm is about 5 minutes. The most time-consuming operation is to solve the large sparse linear systems. In this paper, we solve these system directly using the conjugate gradient method. If a factorization step, such as Cholesky factorization, is applied, the execution time can be further reduced.

VI. Conclusion

We have presented a semi-automatic approach to fit a template mesh to multi-view normal data. This approach reduces the acquisition time compared to state-of-the-art approaches which employ structured light scanning to generate the low-resolution 3D reconstruction. The method consists of three steps: feature-based registration, normal registration, and multi-view shape refinement. In the feature-based registration, we match a few manually selected feature points. In the normal registration, we deform the template to align semantic positions to the normal data. Since the resulting cost function is highly non-linear, we propose a linearisation method for efficient optimization. In the multi-view refinement step, we further refine the shape by enforcing the consistency of normals in neighbouring views.

Future work: In the real data experiment, to prevent errors, we have asked our subjects to close their eyes and mouth. To relax this constraint, a reliable face contour tracker could help. By restricting the movements of eyes and mouth to tracked contours, many complicated expressions could be captured. Moreover, our camera set-up is currently only capable of covering the frontal face. However, it can be expected that adding more cameras allows fitting a complete 3D head template with the same approach. In future work, we would like to apply this technique to a large database of subjects to build a morphable face model that features very high resolution geometry.

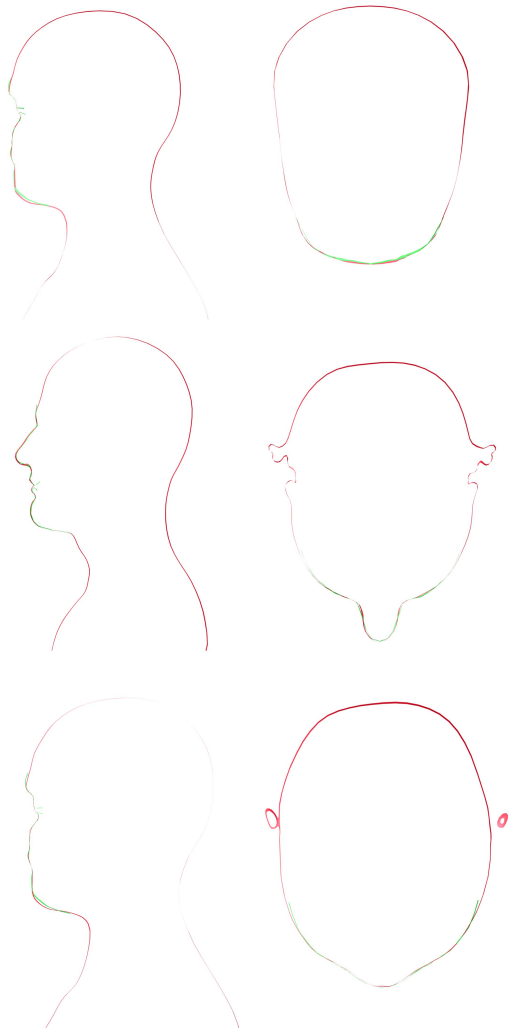

Figure 3. Some slices of the synthesis ground-truth model and our result. The red contour indicates the ground-truth model, and the green contour indicates our result. The left column includes three vertical slices, and the right column includes three horizontal slices. (To evaluate the overlap clearly, an interested reader can open a digital version of this paper and can zoom into the figure)

References

[1] Z. Deng, P.-Y. Chiang, P. Fox, and U. Neumann, "Animating blendshape faces by cross-mapping motion capture data," in *Proc. Interactive 3D Graphics and Games*, 2006, pp. 43–48.

[2] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proc. SIGGRAPH*, 1999, pp. 187–194.

[3] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, "Creating a photoreal digital actor: The digital emily project," in *Proc. IEEE Computer Vision and Pattern Recognition*, 2009, pp. 69–80.

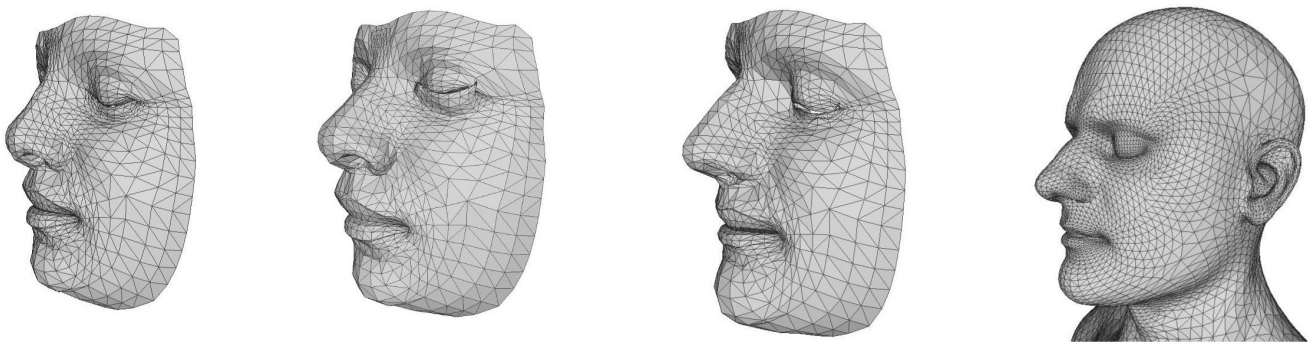

Figure 4. Template fitting progress of our algorithm. From left to right: feature-based registration result, normal registration result, final result, and the model used to generate the synthetic ground-truth input data.

Figure 5. Quantitative results. Each column represents a test. For any column, the first and the third figures are two of the normal maps used in the test; The second and the forth figures are two views of the result with normal map textured and template mesh overlapped.

[4] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid ICP algorithms for surface registration," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[5] R. J. Woodham, "Shape from shading," B. K. P. Horn, Ed. Cambridge, MA, USA: MIT Press, 1989, ch. Photometric method for determining surface orientation from multiple images, pp. 513–531.

[6] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3d geometry," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 24, pp. 536–543, Jul. 2005.

[7] B. De Decker, J. Kautz, T. Mertens, and P. Bekaert, "Capturing multiple illumination conditions using time and color multiplexing," in *Proc. IEEE Computer Vision and Pattern Recognition*, 2009, pp. 2536–2543.

[8] T. Malzbender, B. Wilburn, D. Gelb, and B. Ambrisco, "Surface enhancement using real-time photometric stereo and reflectance transformation," in *Proc. Eurographics Symposium on Rendering Techniques*, 2006, pp. 245–250.

[9] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec, "Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination," in *Proc. Eurographics Symposium on Rendering Techniques*, Jun. 2007, pp. 183–194.

[10] W.-C. Ma, A. Jones, J.-Y. Chiang, T. Hawkins, S. Frederiksen, P. Peers, M. Vukovic, M. Ouhyoung, and P. Debevec, "Facial performance synthesis using deformation-driven polynomial displacement maps," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 27, pp. 121:1–121:10, 2008.

[11] G. Fyffe, T. Hawkins, C. Watts, W.-C. Ma, and P. Debevec, "Comprehensive facial performance capture," *Computer Graphics Forum (Proc. Eurographics)*, vol. 30, pp. 425–434, Apr. 2011.

[12] C. A. Wilson, A. Ghosh, P. Peers, J.-Y. Chiang, J. Busch, and P. Debevec, "Temporal upsampling of performance geometry using photometric alignment," *ACM Transactions on Graphics*, vol. 29, pp. 17:1–17:11, Apr. 2010.

[13] F. Pighin, R. Szeliski, and D. H. Salesin, "Resynthesizing facial animation through 3d model-based tracking," in *Proc. IEEE International Conference on Computer Vision*, vol. 1, 1999, pp. 143–150.

[14] D. Decarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking," *International Journal of Computer Vision*, vol. 38, pp. 99–127, 2000.

[15] V. Blanz, C. Basso, T. Poggio, and T. Vetter, "Reanimating faces in images and video," *Comput. Graph. Forum (Proc. Eurographics)*, vol. 22, no. 3, pp. 641–650, 2003.

[16] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 24, pp. 426–433, 2005.

[17] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 30, pp. 77:1–77:10, Jul. 2011.

[18] H. Huang, J. Chai, X. Tong, and H.-T. Wu, "Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 30, pp. 74:1–74:10, 2011.

[19] J. R. Tena, F. D. la Torre, and I. Matthews, "Interactive region-based linear 3d face models," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 30, pp. 76:1–76:10, Jul. 2011.

[20] R. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," in *Proc. IEEE Computer Vision and Pattern Recognition*, 1986, pp. 364–374.

[21] M. Dekker, *Mathematical Programming*. CRC, 1986.

[22] O. Sorkine, D. C. Or, Y. Lipman, M. Alexa, C. Rossl, and H. P. Seidel, "Laplacian surface editing," in *Proc. 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 175–184.