Computer Aided Geometric Design 50 (2017) 1–13

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Robust 3D face modeling and reconstruction from frontal and side images ☆

Hai Jin [a], Xun Wang [b], Zichun Zhong [a], Jing Hua [a,*]

[a] Department of Computer Science, Wayne State University, Detroit, MI 48098, USA
[b] School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

ARTICLE INFO

Article history:
Received 21 April 2016
Received in revised form 6 November 2016
Accepted 17 November 2016
Available online 23 November 2016

Keywords:
3D face reconstruction
Non-negative Matrix Factorization
Face modeling

ABSTRACT

Robust and effective capture and reconstruction of 3D face models directly by smartphone users enables many applications. This paper presents a novel 3D face modeling and reconstruction solution that robustly and accurately acquire 3D face models from a couple of images captured by a single smartphone camera. Two selfie photos of a subject taken from the front and side are first used to guide our Non-Negative Matrix Factorization (NMF) induced part-based face model to iteratively reconstruct an initial 3D face of the subject. Then, an iterative detail updating method is applied to the initial generated 3D face to reconstruct facial details through optimizing lighting parameters and local depths. Our iterative 3D face reconstruction method permits fully automatic registration of a part-based face representation to the acquired face data and the detailed 2D/3D features to build a high-quality 3D face model. The NMF part-based face representation learned from a 3D face database facilitates effective global and adaptive local detail data fitting alternatively. Our system is flexible and it allows users to conduct the capture in any uncontrolled environment. We demonstrate the capability of our method by allowing users to capture and reconstruct their 3D faces by themselves.

1. Introduction

With the advancement in visual sensing and acquisition technology, the ability to accurately capture 3D human faces has been significantly improved in recent years. The popular methods include laser scanning (Axelsson, 1999), structured light scanning (Scharstein and Szeliski, 2003), RGBD camera (Khoshelham and Elberink, 2012; Zhang, 2012) or multiview stereo (Seitz et al., 2006). Capture and reconstruction of 3D face models enable many applications such as modeling (Wen and Huang, 2004), animation (Cao et al., 2013), gaming (Lim et al., 2006), security (Bowyer et al., 2006) and 3D printing (Campbell et al., 2011). The current solutions often require expensive equipments and a significant level of expertise to achieve high-quality captures and reconstructions. They are far beyond the capability of general end users and therefore limit the potential applications of the technologies. Ichim et al. (2015) presented a solution for creating 3D avatar using hand-held video input. However, the method mainly focuses on texture synthesis using the input video clip. The geometry of the reconstructed face mainly relies on a Structure-from-Motion (SFM) method to build a point set surface. It requires extensive smoothing and denoising with a morphable surface, which will generate a face model not very similar to the

☆ This paper has been recommended for acceptance by Konrad Polthier.

* Corresponding author.
E-mail address: jinghua@wayne.edu (J. Hua).

http://dx.doi.org/10.1016/j.cagd.2016.11.001
0167-8396/

Fig. 1. Reconstructed 3D face model from frontal and side images.

original human subject. Cao et al. (2013) proposed a system to animate an avatar face using a single camera. Their work focused on tracking a user's facial expressions and then synthesizing the corresponding expression geometry in an avatar rather than reconstructing high-fidelity 3D face models.

The goal of our work is to provide a viable solution for allowing a general end user to robustly and accurately model and reconstruct the user's 3D face using a single smartphone camera. With a single smartphone camera, the user can capture his/her face by himself or herself. Using the captured images as an input, the solution needs to robustly reconstruct a high-quality 3D face. The straightforward idea based on this input data would be relying on structure from motion or multiview reconstruction methods (Seitz et al., 2006). Unfortunately, these methods fail upon this low-resolution, blurred, noisy and often incomplete data. Robust surface reconstruction of a high-quality face model from the blurred, noisy and incomplete data is a very challenging task. Also, for the end users to record an entire head scan video of him/herself is a time-consuming and uneasy work.

To overcome these challenges, we develop a part-based 3D face representation, learned from a 3D face database using Non-negative Matrix Factorization (NMF) as prior knowledge, to facilitate robust global and adaptive detail data fitting alternatively to reconstruct an accurate and complete 3D face model. Only two selfie images from the front and side will be used as the input to a later iterative reconstruction process. Our iterative 3D face fitting method permits fully automatic alignment of the NMF part-based 3D face representation to the input facial images and the detailed 2D/3D features to reconstruct a high-quality 3D face model. Fig. 1 shows the reconstructed result using our proposed method. The system is flexible because it does not require any strong 3D facial priors and allows users to conduct the capture in any uncontrolled environment. The capability of our method is demonstrated by allowing users to capture a range of 3D faces using an iPhone 5 camera as shown in Section 3. We have also used two Internet-downloaded images to reconstruct a 3D face model.

The main contributions of this paper can be summarized as follows:

- It develops a deformable NMF part-based 3D face representation from a 3D face database which facilitates robust global and adaptive detail data fitting alternatively through the variations of weights. The bases have the property of local support.
- Our deformable part-based 3D face model serves as a better morphable model for data fitting and reconstruction under geometry and illumination constraints. It presents a fully automated iterative 3D face reconstruction method which automatically registers the deformable part-based 3D face representation to the acquired face images and the detailed geometric features as well as illumination constraints to reconstruct a high-fidelity 3D face model.
- It provides general end users with a novel 3D face capture and reconstruction solution that robustly and accurately acquires 3D face models from a single smartphone camera.

1.1. Related work

Capturing and reconstructing 3D surfaces of objects is one of major research topics in geometric modeling, computer graphics and computer vision. Human face reconstruction and modeling is one of the most active ones among general surface reconstructions. Various methods on face modeling (Kittler et al., 2005) have been extensively studied.

Thanks to the rapid development of capturing devices, the modeling of faces has become more accurate and automated. From scanning directly by professional laser scanners, to using multiple high-resolution cameras to capture 3D face based on multi-view geometry (Mohamed Daoudi, 2013), researchers have achieved significant successes recently. Commercial laser scanners, such as *Cyberware* and *NextEngine*, can now provide us high-quality face modeling. Also, stereo-based face

modeling techniques (Beeler et al., 2010, 2011; Amberg et al., 2007), relying on multiple high-resolution cameras, can also achieve high-quality face modeling. For example, Beeler et al. (2010) used five high resolution digital single-lens reflex (SLR) cameras to capture accurate 3D geometry of a face from a synchronized shot. However, the costs of these systems are still very high and they require a complicated calibration process before actual operation, which limits the uses of these systems in non-studio environments or by general end users.

On the contrary, to using expensive high performance scanners or stereo capturing systems, Blanz and Vetter (1999) presented a morphable face model for reconstructing 3D face from a single image and Lei et al. (2008) presented a face shape recovery method using a single reference 3D face model. In order to recover the missing depth information from 2D image, prior knowledge of face is needed. Learning through a face database is an effective approach for tackling this problem. Therefore, statistical face models based on Principle Component Analysis (PCA) are proposed and constructed (Blanz and Vetter, 1999), and then used as prior for estimating depth information. Similar face fitting methods, such as piecewise PCA sub-models (Tena et al., 2011), were proposed as well. Approaches based on 2D images achieved plausible 3D face reconstruction results (Blanz and Vetter, 1999), however, they need to carefully tune the parameters for pose and illumination, which requires a lot of empirical knowledge and makes it impractical for general end users. Also, the detailed geometry reconstruction is the main limitation of these methods due to the global property of PCA methods. As the extension of their work, Blanz et al. applied the morphable 3D face reconstruction method to facial recognition problem (Blanz and Vetter, 2003).

More recently, modeling based on RGBD camera such as *Kinect* (Zhang, 2012; Newcombe et al., 2015), has become another active research topic. Chen et al. (2013) proposed a system that captures a high-quality face model and its performances using a single *Kinect* device. They provided a markerless motion capture approach that increases the subjects' flexibilities and improves the resolution of facial geometry. Newcombe et al. (2015) presented a *Kinect* based 3D reconstruction method for non-rigid objects such as human face. Without using a RGBD device, Cao et al. (2013) proposed a system to animate an avatar face using a single camera. Other than capturing and reconstructing the entire face, they only detected a set of feature points for computing shape regression to animate the target avatar face. As the extension work, they proposed displaced dynamic expression regression method to further improve the performance of the system (Cao et al., 2014). Their work focused on tracking and synthesizing facial expression geometry rather than reconstructing high-fidelity 3D face models.

Another direction of work is 3D face reconstruction from photometric stereo-based method. Suwajanakorn et al. (2014) presented an impressive 3D face reconstruction technique from a collection of images or a clip of videos of the person. They first learned an average 3D face of the subject from the input images as the base shape, which was then used to fit the individual images with different expressions. A shape-from-shading method was used to optimize the fine details of the shape. However, since this method is reconstructing the shape by optimizing pixel depth of the image, the side of the 3D face such as cheeks and ears are not fully reconstructed. By this means, this method is a 2.5D reconstruction of the face.

SFM-based shape reconstruction has also been researched extensively. However, it is difficult to reconstruct a fine detailed 3D face model due to the high noise-to-signal ratio. Smoothing and denoising to the point cloud data will significantly reduce the high frequency details of the model. Ichim et al. (2015) presented a dynamic 3D avatar creation approach from mobile devices. The approach uses a noisy point cloud built from SFM as the constraint to deform the template 3D head. As the purpose is for entertainment applications on mobile devices, the details are added by refining the albedo texture and normal map instead of actually adjusting the local geometry details. In order to create a realistic detailed 3D avatar head, many off-line edits are still needed, which is a difficult task for the general end users who have little or no 3D modeling knowledge.

Our work mainly focuses on capturing and reconstructing face geometry robustly and automatically by general end users. Therefore, in this paper, we assume the input two images for our 3D face reconstruction are self-acquired from a single camera of a mobile device. Extending from learning-based approach, we instead establish a deformable part-based 3D face representation based on non-negative matrix factorization of a 3D scanned face database prior to the reconstruction stage. Compared to the previous methods, our deformable NMF-based 3D face model serves as a better and more robust morphable model for data fitting and reconstruction as the NMF bases are corresponding to localized features that correlate better with the parts of 3D faces under geometry and illumination constraints. During the reconstruction stage, our method can produce a high-quality 3D face model from the input images and recover details of the subject's face through surface reconstruction from shading constraints. Fig. 2 illustrates the entire process, which is fully automated without users' intervention.

2. Face reconstruction through deformable part-based 3D face model

In this section, we present in detail the 3D face reconstruction technique using our deformable NMF part-based 3D face model. In Section 2.1, we first explain Constrained Local Models (Cristinacce and Cootes, 2006) for the initial feature point detection and pose estimation based on the input images. Then, in Section 2.2, we explain the process to build our deformable part-based 3D face representation based on non-negative matrix factorization of a 3D face database. In Section 2.2.1, we show how the part-based 3D face model can be used as a deformable model to reconstruct a high-quality face from a frontal and a side facial images. In Section 2.2.2, we present the detail fitting process based on the illumination

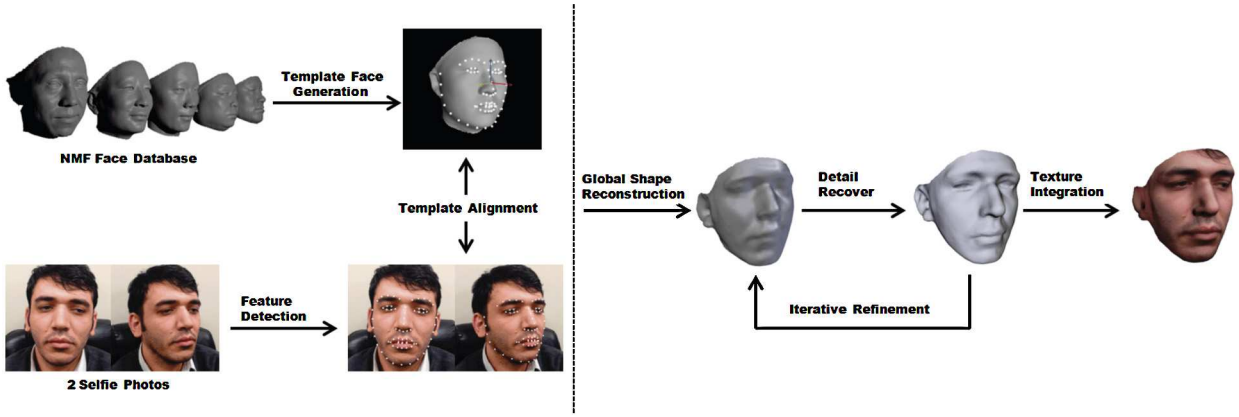
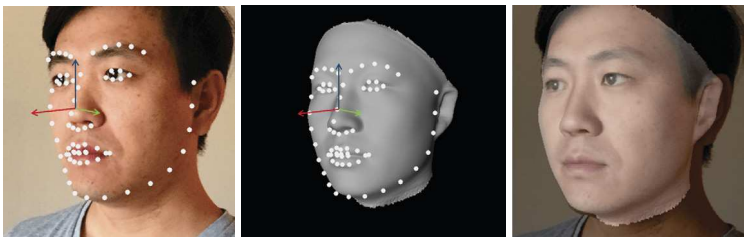


Fig. 2. The pipeline of our iterative 3D face reconstruction based on deformable NMF part-based 3D face model.



(a) (b) (c)

Fig. 3. Feature point detection and initial alignment. (a) is detected features points and the estimated pose according to the input image; (b) is affine transformed template face based on the estimated parameters according to (a); (c) is aligned template face to the input face.

constraints. Briefly, our approach reconstructs a final 3D face iteratively by alternating two steps: global fitting and detail fitting.

2.1. Initial pose estimation and template face alignment

An accurate initial alignment of the template 3D face to the input images is important for the fitting process since a good initial state may significantly reduce the optimization iterations and improve the reconstruction quality. We employ the Constrained Local Models (CLM) (Cristinacce and Cootes, 2006) to estimate the feature points P_n (n is the number of points), which are used by a feature-based head pose estimator to obtain the initial template translation (T), rotation (R) and scale (S) automatically. Fig. 3 shows the detected feature points P_n on the input facial image and the posing direction of the face. P_n is a vector of 2D pixel coordinates of the feature points on the image.

Prior work (Cootes et al., 2002; Morency et al., 2003) has studied how to estimate head poses from monocular image. In this paper we propose a method similar to (Cootes et al., 2002). The ground-truth poses and their depth maps are acquired via a *Kinect* device along with the images to build a training database. Then, we train a view-based appearance model to estimate the head poses. For each key frame i, we obtain a training set $F_i = \{P_{ni}, \Omega_i\}$, where P_{nk} are the feature points detected by the CLM and $\Omega_i = T, R, S$ are the affine transformation parameters. Note that, the translation T and scaling S can be estimated straightforwardly using the shape of the feature points. Therefore, we focus on solving the rotation R with the training set P_{ni}, R. Particularly, the rotation term R consists of three angles $\theta_x, \theta_y, \theta_z$ around x, y, z axis, respectively. Therefore, the training set can be noted as $\{P_{ni}, \theta\}$, where θ represents θ_x, θ_y, or θ_z. The shape of the feature points can be represented as

$$P = \bar{P} + Q\vec{\beta}, \tag{1}$$

where $\bar{P}$ is the average shape of feature points among all the faces in the training database, Q is the variation matrix of the training data, which can be obtained using the PCA method, and $\vec{\beta}$ is the coefficient which controls the shape of feature points. The prediction model can be established via the following regression model:

$$\vec{\beta} = \vec{a} + \vec{b}cos(\theta) + \vec{c}sin(\theta), \tag{2}$$

where $\vec{a}, \vec{b}, \vec{c}$ are the parameters to be trained from the training set F. Eq. (2) can be solved by $(cos(\theta), sin(\theta))' = R^{-1}(\vec{\beta} - \vec{a})$, where R^{-1} is pseudo inverse of $(\vec{b}|\vec{c})$, i.e., $R^{-1}(\vec{b}|\vec{c}) = I_2$. Thus, given a detected feature point set, P, we first compute the representing coefficient $\vec{\beta}$ using Eq. (1), then, we can obtain θ based on the trained predictive model, i.e., Eq. (2).

Once the template face is transformed to the same pose as the input image, the same number of feature points are detected on the rendered template face (i.e., 2D projected image of the 3D template face), which are traced back to 3D space to obtain the nearest corresponding vertices V_n. V_n, corresponding with P_n, is a 3D vertex coordinate vector of the feature points on the 3D template face model. We denote this operation as $\mathbb{F}$ and the details will be described in Section 2.2.1.

2.2. Deformable part-based 3D face model for data fitting

PCA and vector quantization (VQ) are two common approaches to decompose a data and PCA is also widely used in 3D face data. PCA learns the face data globally and decompose it to 'eigenfaces', which are basis vectors in the face space. Different from PCA and VQ, NMF decomposes a shape in localized features (Lee and Seung, 1999). In this paper, we construct a deformable 3D face representation by parts based on non-negative matrix factorization, which will significantly improve the reconstruction of details of the 3D face. Any face can be represented as a linear combination of basis parts. The part-based 3D face model permits better local control, therefore leading to more accurate and robust morphable fitting to the target.

To generate better performing NMF bases, the scanned 3D face data samples need to be carefully aligned and registered first. There exist many methods facilitating this task (Li and Iyengar, 2015). Given a 3D face database with M examples, we first employ multi-scale expectation–maximization iterative closest point method to accurately register all the 3D face examples and then resample all the examples into the same number of vertices to establish a dense mapping and indexing (Granger and Pennec, 2002). The 3D face database can then be constructed as a $N \times M$ matrix S, where N is the number of vertices in a 3D face and M is the number of face examples in the database. Each column represents the geometry of the face with a data vector of 3D coordinates, $s_i = \{x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n\}^T \in \mathbf{R}^{3n}$. Note that, the quality of the dense correspondence will significantly affect the result of factorization.

Next, non-negative factorization of matrix S is constructed as $S \approx BW$, where B is the basis matrix and W is the weight, or it can be represented as

$$S_{ij} \approx (BW)_{ij} = \sum_{a=1}^{r} B_{ia} W_{aj}, \tag{3}$$

where r is the rank of factorized basis. Then each face in the database can be restored from

$$s_k = B\vec{\mathbf{w}}_k, \tag{4}$$

where $\vec{\mathbf{w}}_k = (w_1, w_2, \ldots w_a)^T$ is corresponding column vector in weight matrix W. New faces can be generated by manipulating the weight vector $\vec{\mathbf{w}}_k$ and compute the linear combination of the bases.

To find a factorization, we need to solve the following optimization problem,

$$(B, W) = \underset{B \geq 0, W \geq 0}{\operatorname{argmin}} \|S - BW\|^2. \tag{5}$$

According to the theorem in Lee and Seung (2001), the Euclidean distance $\|S - BW\|$ does not increase under the following update rules:

$$W_{aj} \leftarrow W_{aj} \frac{(B^T S)_{aj}}{(B^T BW)_{ia}}, \qquad B_{ia} \leftarrow B_{ia} \frac{(SW^T)_{ia}}{(BWW^T)_{ia}}. \tag{6}$$

In practice, B and W are initialized as random dense matrix (Berry et al., 2007) and a simple additive update rule for weight W is used as in Lee and Seung (2001),

$$W_{aj} \leftarrow W_{aj} + \eta_{aj}[(B^T S)_{aj} - (B^T BW)_{aj}], \tag{7}$$

where

$$\eta_{aj} = \frac{W_{aj}}{(B^T BW)_{aj}}. \tag{8}$$

Once NMF basis matrix B is computed, arbitrary new face can be decomposed based on the bases and represented by the corresponding weight vector. In other words, varying the weights over the NMF basis matrix B constitutes a deformable part-based 3D face model that can be used to fit in a nonrigid means to any given 2D/3D face data input. Therefore, we name $s = B\vec{w}$ as a deformable part-based 3D face representation which carries the prior knowledge of faces for nonrigid fitting.

We used 120 scanned face data for training deformable part-based 3D face model. In this paper, the data was normalized and registered based on the multi-scale expectation–maximization iterative closest point method (Granger and Pennec,

Fig. 4. The sample 3D faces in our database.

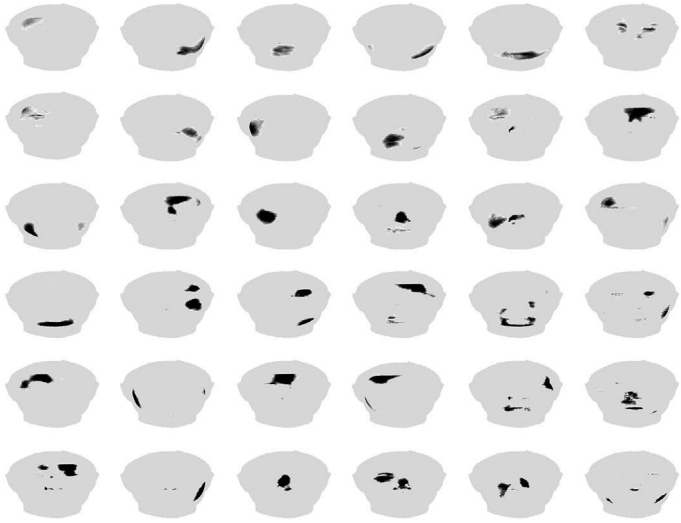


Fig. 5. The randomly selected 36 computed NMF bases projected onto an average face to display the local support of the bases.

2002). Every face data has 60000 vertices, which are represented as vector $\vec{x}_i = \{r_1, h_1, \theta_1, r_2, h_2, \theta_2, \ldots, r_n, h_n, \theta_n\} \in \mathbf{R}^{3n}$. Since NMF requires non-negative elements in the matrix, face data samples were transformed into cylindrical coordinate system so that all the values in the vector are positive. Data vectors of 120 subjects formed a $3n \times 120$ matrix for NMF decomposition. We chose 120 columns for basis matrix as the factorization basis, therefore, the weight vector $\vec{w}$ has 120 elements. Fig. 4 shows some samples of our 3D face database used in this paper and Fig. 5 shows the local support of the part bases on an average face model. Note that, all the faces used in training the deformable part-based 3D face model are not employed for testing the performance of our system.

2.2.1. Iterative global reconstruction with part-based 3D face representation

The reconstruction process is divided into two major parts: global fitting and detail fitting. In this section we explain the global shape reconstruction via feature point fitting by updating the deformable part-based 3D face model iteratively.

In a global fitting step, the deformable part-based 3D face model $B\vec{w}$ will optimize its weights, $\vec{w}$, based on the previously estimated rotation, scaling and translation factors (in Section 2.1) to fit the feature points P_n in input images I. In order to find the best fitted result, we minimize the Euclidean distance as follows,

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmin}} \sum_{k=1}^{m} \| P_n^k - \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))\|^2, \tag{9}$$

where m is the number of images used for global fitting, B is the basis of the part-based 3D face representation, R is a rotation and scaling matrix, T is a translation matrix, $\mathbb{F}$ is feature point extraction operation and $\mathbb{P}$ is the projection operation. We followed the approach in Zhang et al. (2000) to estimate the camera intrinsic parameters, which can be further used to compute the projection matrix. In practice, we use two images taken from the front and the side as the input, thus, the number of images $m = 2$. However, the number of input images can be unrestricted in our deformable fitting model. In general, using more images may produce a better quality reconstruction, but the computational cost will also increase and the user-experience can be adversely affected. To our experience, two images can already provide a rather high-quality reconstruction while keeping an excellent performance in terms of computational time. Fig. 2 shows the feature points detected in both the frontal and side images.

We solve the Eq. (9) via gradient descent method. We compute the partial derivative of the energy term in Eq. (9) with respect to the weight vector $\vec{w}$ as follows,

$$\nabla E(\vec{w}) = \frac{\partial E}{\partial \vec{w}} = -2 \sum_{k=1}^{m} (P_n^k - \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))) \frac{\partial \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))}{\partial \vec{w}}, \tag{10}$$

where

$$\frac{\partial \mathbb{P}^k(\mathbb{F}(R^k B\vec{w} + T^k))}{\partial \vec{w}} = \vec{c}_k \tag{11}$$

is a constant vector $\vec{c}_k$ for each image. Thus, the gradient in each iteration is only determined by the distance between the projected feature vertices in the current stage and the target feature points in the input image. In each iteration i, the weight vector $\vec{w}$ is updated by

$$\vec{w} \leftarrow \vec{w} - \nabla E(\vec{w}_i). \tag{12}$$

The Algorithm 1 iteratively updates the weight vector and obtains the optimized $\vec{w}$. This process continues until the weight vector converges, which usually takes around 5∼7 iterations. This step recovers the global features in the data, such as face size, and approximates the shape of different parts of the face. Since we can obtain the correspondence between the reconstructed model and the input image sequence, we can project the registered images to the result shape to obtain a texture, reconstructed 3D face model, $(B\vec{w}, \rho)$, where ρ is the texture over $B\vec{w}$. This will be used for the next NMF based detail fitting process.

Algorithm 1 Iterative Global Reconstruction.

```
procedure ITERATIVE 3D FACE RECONSTRUCTION
    w⃗, T, R, ℙ, 𝔽, threshold ← initialization
    Compute the gradient ∇E(w⃗_i) using Eq. (10)
    while ∇E(w⃗_i) > threshold do
        w⃗ ← w⃗ − ∇E(w⃗_i)
        Re-compute ∇E(w⃗_i) using Eq. (10)
    end while
    S = Bw⃗
end procedure
```

2.2.2. Shading based detail reconstruction

Although our part-based fitting can reconstruct a quite plausible 3D face with well fitted global features, the details (such as the major wrinkles and folds around mouth and nose) are still akin to the template shape. Thus, we perform a detailed refinement process based on the result of global fitting.

In the NMF detail fitting step, we perform data fitting to the images in order to fine tune the model by minimizing the Euclidean distance from the rendered 3D face to input images. Since the global fitting recovered the rough shape of the face and the transformation matrix, the current face model can be automatically projected to the corresponding images using the information obtained from previous feature point alignment process. We denote the rendered image of the 3D face model as $\hat{I} = \mathbb{R}(B\vec{w}, \rho)$. That is to say, we project and render each vertex of the face model on the image plane to form $\hat{I}$, and

the comparison between the rendered image and the original image is based on the projected locations of the 3D vertices. Therefore, the main goal of the optimization problem is to minimize the sum of distances as follows,

$$\vec{w} = \underset{\vec{w}\in W}{\operatorname{argmin}}\left(\sum_{i=1}^{m} \|I_i(\mathbb{P}^i(V)) - \hat{I}_i\|^2 + \eta\|\vec{w} - \vec{w}_g\|^2\right), \tag{13}$$

where m is the number of images used for detail fitting, ρ is the albedo texture, $\mathbb{R}$ is the rendering operation, $I_i(\mathbb{P}^i(V))$ is the re-sample of the input image I based on the projected locations of the 3D vertices V on the 2D image domain, η is the regularization coefficient and $\vec{w}_g$ is the weight vector of the local parts derived from global fitting. The second term is the regularization term which constrains the final shape is close to the result of previous global fitting stage. In practice, we only use the frontal image for detail refinement, i.e., $m = 1$, since most of the details are captured in the frontal view of the image. Since the rendering operation $\mathbb{R}_i$ and the corresponding image I_i is known, the only variable that needs to be updated is weight vector $\vec{w}$. Based on the same idea to the global shape reconstruction step, we compute the derivative of the error function with respect to $\vec{w}$ to find the maximum decent direction, which is used for updating the weight vector.

In order to compute the partial derivative of the energy term with respect to $\vec{w}$, the shading model of the rendering operation needs to be defined. Inspired by Suwajanakorn et al. (2014), we transform the optimization problem in Eq. (13) to an optimization problem for the photometric normals as follows:

$$\boldsymbol{N} = \underset{\boldsymbol{N}}{\operatorname{argmin}} \|I(\mathbb{P}(V)) - \hat{I}\|^2. \tag{14}$$

For shading computation, we use the Phong reflectance model in our method. In this paper, the rendered image $\hat{I}$ of the 3D face is computed by

$$\hat{I} = \mathbb{R}(B\vec{w}, \vec{\rho}) = (k_a + k_d(\boldsymbol{N}\vec{l}) + k_s(\boldsymbol{V} * \vec{\boldsymbol{r}})) \circ \vec{\rho}, \tag{15}$$

where k_a, k_d and k_s are constant weights of ambient light, diffuse light and specular light, respectively. $\vec{l}$ is the light directions. $\boldsymbol{N}$ are the normals at the vertices and $\vec{\rho}$ are the albedo vector containing the texture information at each vertex. $\vec{\boldsymbol{r}}$ are the reflection vectors and $\boldsymbol{V}$ are vectors from each vertex to the view point. The $*$ represents a row-wise inner product and the $\circ$ represents the element-wise product. Since we assume a weak specular reflection in the model, we ignore the specular term in Eq. (15) when optimizing vertex normals. We compute the final vertex normals by minimizing the following equation:

$$\{\boldsymbol{N}, \vec{l}, k_a, k_d\} = \underset{\boldsymbol{N},\vec{l},k_a,k_d}{\operatorname{argmin}} \|I(\mathbb{P}(V)) - (k_a + k_d(\boldsymbol{N}\vec{l})) \circ \vec{\rho}\|^2. \tag{16}$$

Lighting Optimization: To optimize the shading through changing the vertex normals by Eq. (16), we first estimate the lighting parameters $\vec{l}, k_a, k_d$ by solving the optimization problem,

$$\{\vec{l}, k_a, k_d\} = \underset{\vec{l},k_a,k_d}{\operatorname{argmin}} \|I(\mathbb{P}(V)) - (k_a + k_d\boldsymbol{N}\vec{l}) \circ \vec{\rho}\|^2, \tag{17}$$

where the vertex normals $\boldsymbol{N}$ and the albedo ρ are considered as constants at this stage. To simplify the optimization problem, we let the albedo be equal to the input image I, so the problem becomes

$$\{\vec{l}, k_a, k_d\} = \underset{\vec{l},k_a,k_d}{\operatorname{argmin}} \sum_{i=1}^{e} \|1 - (k_a + k_d\vec{l} \cdot \vec{n}_i)\|^2, \tag{18}$$

where e is the number of vertices and $\vec{n}_i$ is the normal vector of each vertex. In practice, we can solve the linear equation in a linear time by randomly selecting 40 vertices on the shape. Since it is a over determined linear system, it can be easily solved by QR factorization. Fig. 6 shows the result of lighting optimization.

Normal Optimization: After we estimate the optimal lighting parameters, we compute the partial derivative of normal vector $\vec{N}$ with respect to the weight vector $\vec{w}$, and then the Eq. (13) can be solved by chain rule. We define the normal of each vertex as follows:

$$\vec{n}_i = \frac{\vec{u} \times \vec{v}}{\|\vec{u} \times \vec{v}\|}, \tag{19}$$

where $\vec{u}$ is the vector from V_i to its adjacent vertex in positive x direction and $\vec{v}$ is the vector from V_i to its adjacent vertex in positive y direction. Since the vertices are in 3D domain, we pre-compute the vertex location adjacency in cylinder coordinate system. Here, we only update the depth value of each vertex to modify the vertex normal. Therefore, we compute the partial derivative on z element of the normal, which is $\frac{\partial \vec{n}_{iz}}{\vec{w}}$. Therefore, the Jacobian of the normal vector $\vec{N}_z$ is

$$J = \frac{\partial \vec{N}_z}{\partial \vec{w}} = [\frac{\vec{n}_{1z}}{\vec{w}}, \frac{\vec{n}_{2z}}{\vec{w}}, ..., \frac{\vec{n}_{kz}}{\vec{w}}] = [\vec{\delta}_1, \vec{\delta}_2, ..., \vec{\delta}_k]. \tag{20}$$

(a) (b) (c)

Fig. 6. Lighting optimization result: (a) is the initial random lighting; (b) is optimized lighting; (c) is the original image.

Thus, the gradient of Eq. (13) can be computed as

$$\nabla E(\vec{w}) = \frac{\partial E}{\partial \vec{w}} = 2(\sum_{j=1}^{M} dI_j(k_d \vec{l}_z \vec{\delta}_j \rho_j) + \eta \vec{w}), \tag{21}$$

where M is the number of vertices, dI_j is the pixel difference in gray scale and $\vec{l}_z$ is the z element of the lighting direction.

We optimize Eq. (13) with respect to the weight vector $\vec{w}$ iteratively using the following algorithm.

Algorithm 2 Iterative Detail Refinement.

procedure ITERATIVE DETAIL REFINEMENT
  $\vec{w}, \alpha, I, threshold \leftarrow initialization$
  Compute normal vector $\vec{N}$ on mesh $S = B\vec{w}$
  $\hat{I} \leftarrow$ Render current mesh with Eq. (15)
  $d = \|I - \hat{I}\|^2$
  while $d > threshold$ **do**
    for each visible vertex v_i on the rendered image $\hat{I}$ **do**
      Compute $\vec{\delta}_i = \partial \vec{n}_{iz}$ with respect to $\vec{w}$
    end for
    Computer $\nabla E(\vec{w})$ using Eq. (21)
    Update $\vec{w} = \vec{w} - \alpha \nabla E(\vec{w})$;
    Update shape normals $\vec{N}$
    Update d
  end while
end procedure

3. Experimental results

In our experiments, the testing users' faces are new to our system. Their face models are not used in prior training process. In a data acquisition and initialization stage, a user takes two selfie photos from front and side using an iPhone 5. In order to align the template face to the input images, 68 feature points (Zhu and Ramanan, 2012) are first detected and the shape of them is decomposed to estimate the head pose using the method described in Section 2.1. The template 3D face is then transformed based on the detected head pose and aligned to the image. The same number of feature points are detected and back projected to the 3D space to obtain the feature vertices on the 3D template face. Then iterative reconstruction as illustrated in Algorithm 1 is conducted to find the optimal weight vector $\vec{w}$ for reconstructing the global shape of the 3D face model. Once the global fitting process converged, the system automatically continues the lighting parameter estimation process before the detail fitting process. Based on the estimated lighting parameters, the detail refinement process is done iteratively by Algorithm 2. Fig. 7 shows the number of iterations against the total fitting energy of 4 subjects. The energy is computed by normalizing the error between the rendered image and the input image in terms of the total number of pixels. Fig. 8 visualizes the intermediate result of detail fitting process after 10, 20, 30 and 40 iterations for subject 1.

The texture is finally mapped to the 3D face after the details are refined. We perform our experiment on a regular PC with 3.0 GHz Core2 CPU, 8 GB memory and GeForce 9800GT graphics card. The global fitting process takes around 5 iterations which takes about 3 seconds. Lighting parameters estimation takes approximately 300 ms and the detailed refinement process takes about 10 seconds.

We compared our NMF based method with PCA based method, and the result is shown in Fig. 9. Fig. 9 shows the reconstructed result by PCA and NMF respectively without detail fitting process. The more bases used in a reconstruction process, the better reconstruction quality can be obtained for both methods. In practice, both methods used the most significant 105 bases to fit the input image. We found the improvement of the reconstruction is very limited beyond these bases as

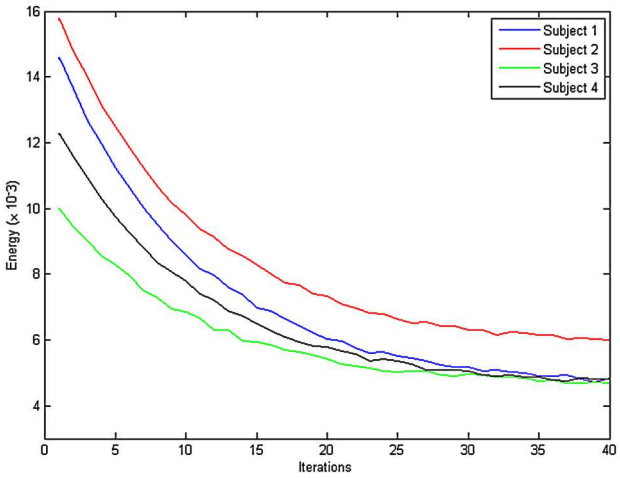


Fig. 7. Energy decrease of detail fitting process.

Fig. 8. Detail fitting results after 10, 20, 30, 40 iterations respectively.

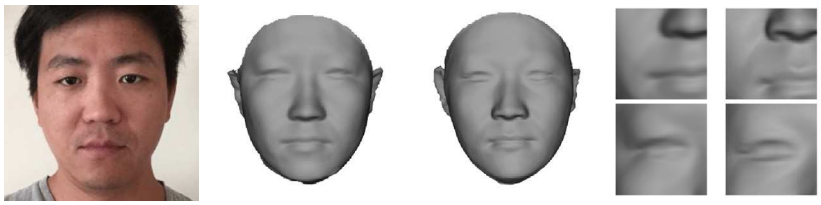


Fig. 9. Comparison of the model reconstructed by PCA (Blanz and Vetter, 1999) and by our method. (a) shows the input frontal image; (b) is the reconstructed result by PCA bases (without detail reconstruction); (c) is the reconstructed result by NMF bases (without detail reconstruction); (d) shows local details of the result from PCA method; (e) shows the local details of the result from our method.

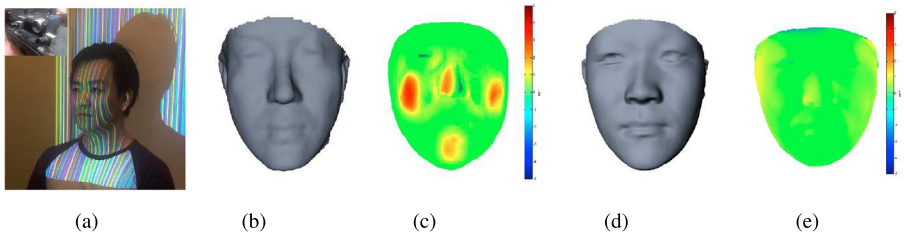


Fig. 10. Comparison of the fringe pattern scanned model with the results reconstructed using *Kinect* and our method. (a) shows a ground-truth scan using a fringe pattern scanner; (b) and (c) are the *Kinect* scanned surface generated by *Dynamic Fusion* (Newcombe et al., 2015) and its Hausdorff distance map (error map) to the ground truth scan; (d) is the reconstructed result using our method and (e) is the color map of Hausdorff distance to the same ground truth scan.

compared to the increase of computational cost. The result shows our NMF based method can effectively reconstruct major wrinkles on the face while PCA based method fails. NMF decomposes the database by parts whereas PCA decomposes globally, which means NMF is more effective on local detail reconstruction than PCA.

In order to further show the quality of the reconstructed result, we used a high-resolution fringe pattern scanner to generate a faithful 3D model of the testing subject as shown in Fig. 10(a). The comparison among *DynamicFusion* (Newcombe

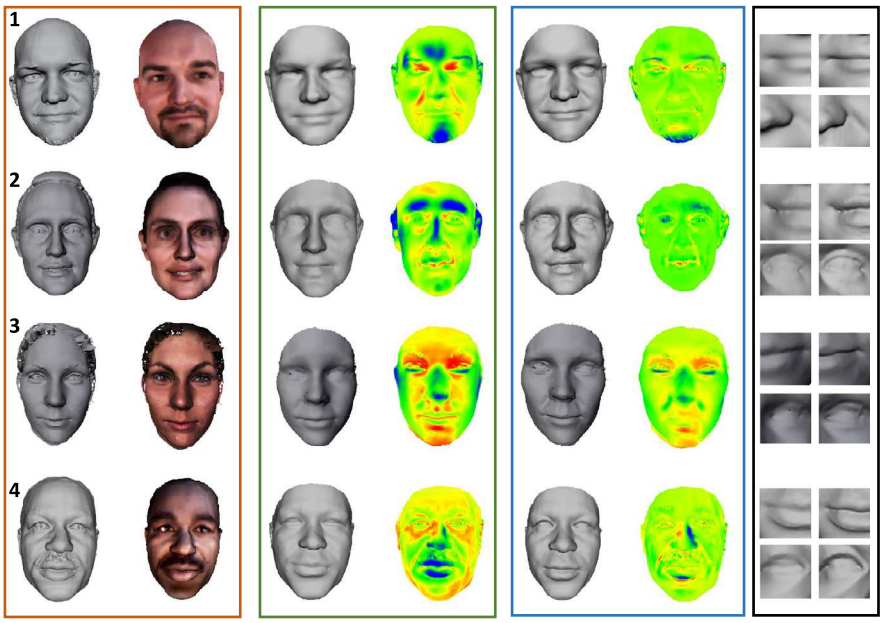


Fig. 11. Comparison of the model reconstructed by PCA and by our method. (a) shows the scanned 3D face models and the rendered image of them; (b) is the reconstructed result by PCA bases (without detail fitting) and its error map; (c) is the reconstructed result by NMF bases (without detail fitting) and its error map; (d) left column shows local details of the result from PCA method and (d) right column shows the local details of the result from our method.

Table 1
Quantitative comparison between PCA based method and our method of the results in Fig. 11. Table shows of the Mean Square Error (MSE), the Standard Deviation (σ_e) and runtime of the two methods.

Subject	PCA			NMF		
	MSE ($\times 10^{-1}$)	σ_e	runtime	MSE ($\times 10^{-1}$)	σ_e	runtime
1	4.89	0.72	3.2 s	2.15	0.41	3.3 s
2	4.45	0.68	3.5 s	1.68	0.38	3.4 s
3	5.95	0.85	3.0 s	3.14	0.51	2.9 s
4	5.21	0.76	3.3 s	1.89	0.45	3.5 s

et al., 2015) result and our result to the ground truth point clouds is shown in Fig. 10. Our reconstruction result recovers more details than *DynamicFusion* and reaches a higher accuracy in terms of 3D face modeling and reconstruction.

To further evaluate the effectiveness of our method, we reconstructed 3D face models from synthetic facial images using PCA based method (Blanz and Vetter, 1999) and our method. High resolution 3D models were rendered to generate the synthetic facial images (Fig. 11(a)), which were used as the input to the reconstruction algorithms. Fig. 11(b) and (c) shows the reconstructed results and the error maps using PCA based method and our method respectively. We computed the error maps for both results using the scanned 3D model as the ground truth, which show our method has smaller error compared to the PCA based method. From Fig. 11(d), we can confirm our method is more powerful in reconstructing local details. Table 1 shows the quantitative comparison between two methods. Our proposed method achieves smaller Mean Square Error and Standard Deviation than the PCA based method while keeping similar runtime performance.

Some more results of our method are shown in Fig. 12. Fig. 13 shows another example, where the two input images are downloaded from Internet. The reconstructed results are shown with and without texture map respectively.

4. Conclusion

In this paper, we present a novel 3D face capture and reconstruction solution that can robustly and accurately acquire 3D face models using a single smartphone camera. In this solution, a deformable NMF part-based 3D face model, learned from a 3D face database, is developed to facilitate robust global and adaptive detail data fitting alternatively through the variations of weights. The part-based 3D face representation serves as a better morphable model for data fitting and reconstruction under geometry and illumination constraints, as the NMF bases are corresponding to localized features that correlate better with the parts of 3D faces.

In our system, self-portrait frontal and side photographs are used as the input to the fully automated iterative reconstruction process. It permits fully automatic registration of the deformable part-based 3D face model to the input images

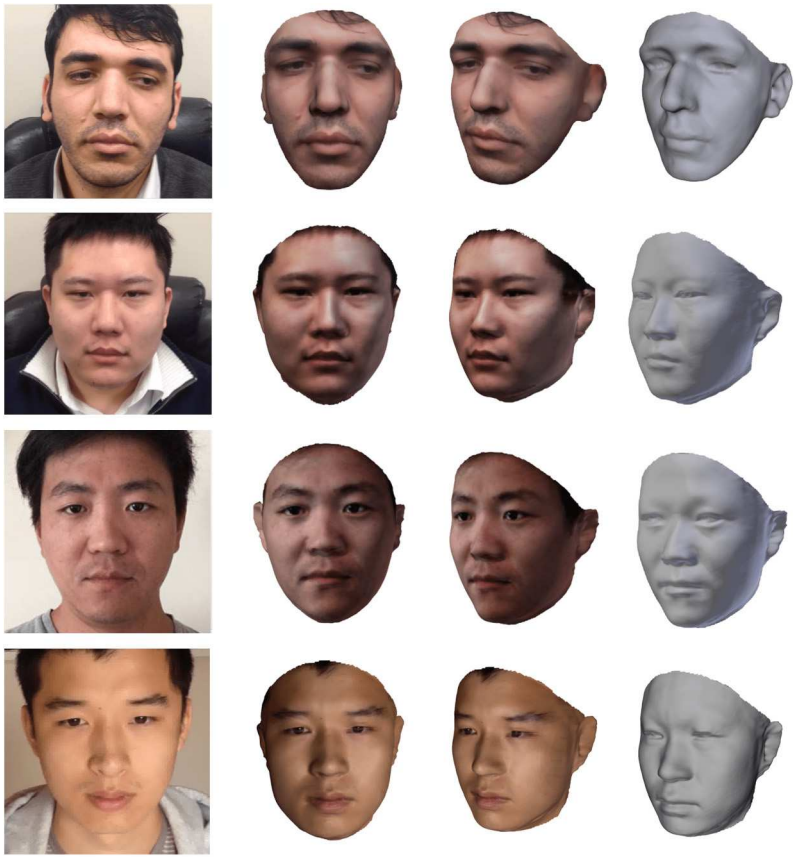


Fig. 12. Some reconstructed 3D faces from input frontal and side photos using our method.

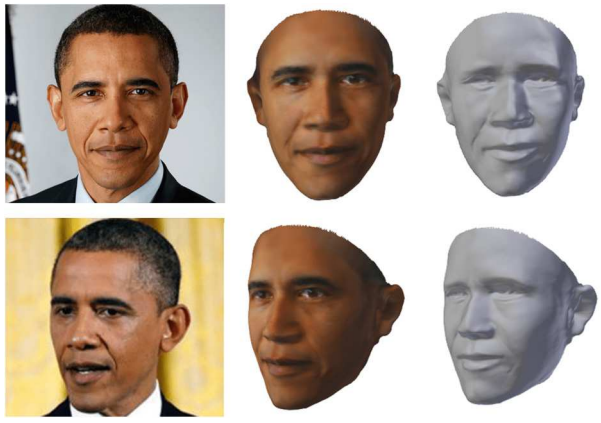


Fig. 13. A reconstructed 3D face from two photos downloaded from Internet.

and the detailed geometric features as well as illumination constraints to reconstruct a high-fidelity 3D face model. The system is flexible as it allows users themselves to conduct the captures in any uncontrolled environment. The capability of our method is demonstrated by several users to capture and reconstruct their 3D faces using an iPhone 5 camera.

Acknowledgements

We would like to thank the reviewers for their valuable suggestions which helped to improve this paper. This work is supported in part by the following grants: LZ16F020002 and NSF IIS-0915933, IIS-0937586, and CNS-1647200.

References

Amberg, B., Blake, A., Fitzgibbon, A., Romdhani, S., Vetter, T., 2007. Reconstructing high quality face-surfaces using model based stereo. In: ICCV, pp. 1–8.

Axelsson, P., 1999. Processing of laser scanner data—algorithms and applications. ISPRS J. Photogramm. Remote Sens. 54 (2), 138–147.

Beeler, T., Bickel, B., Beardsley, P., Sumner, B., Gross, M., 2010. High-quality single-shot capture of facial geometry. SIGGRAPH 29 (3), 40:1–40:9.

Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R.W., Gross, M., 2011. High-quality passive facial performance capture using anchor frames. SIGGRAPH, 75:1–75:10.

Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., Plemmons, R.J., 2007. Algorithms and applications for approximate nonnegative matrix factorization. Comput. Stat. Data Anal. 52 (1), 155–173.

Blanz, V., Vetter, T., 1999. A morphable model for the synthesis of 3D faces. SIGGRAPH, 187–194.

Blanz, V., Vetter, T., 2003. Face recognition based on fitting a 3D morphable model. IEEE Trans. Pattern Anal. Mach. Intell. 25 (9), 1063–1074.

Bowyer, K.W., Chang, K., Flynn, P., 2006. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. Comput. Vis. Image Underst. 101 (1), 1–15.

Campbell, T., Williams, C., Ivanova, O., Garrett, B., 2011. Could 3D printing change the world? Technologies, potential, and implications of additive manufacturing. Atlantic Council, 1–14.

Cao, C., Weng, Y., Lin, S., Zhou, K., 2013. 3D shape regression for real-time facial animation. SIGGRAPH 32 (4), 41:1–41:10.

Cao, C., Hou, Q., Zhou, K., 2014. Displaced dynamic expression regression for real-time facial tracking and animation. SIGGRAPH 33 (4), 43:1–43:10.

Chen, Y.-L., Wu, H.-T., Shi, F., Tong, X., Chai, J., 2013. Accurate and robust 3D facial capture using a single RGBD camera. In: ICCV, pp. 3615–3622.

Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J., 2002. View-based active appearance models. Image Vis. Comput. 20 (9), 657–664.

Cristinacce, D., Cootes, T.F., 2006. Feature detection and tracking with constrained local models. In: BMVC, vol. 1, p. 3.

Daoudi, Mohamed, Srivastava, Anuj, Veltkamp, Remco, 2013. 3D face modeling, analysis and recognition. Wiley.

Granger, S., Pennec, X., 2002. Multi-scale EM-ICP: a fast and robust approach for surface registration. In: ICCV. Springer, pp. 418–432.

Ichim, A.E., Bouaziz, S., Pauly, M., 2015. Dynamic 3D avatar creation from hand-held video input. SIGGRAPH 34 (4), 45.

Khoshelham, K., Elberink, S.O., 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors 12 (2), 1437–1454.

Kittler, J., Hilton, A., Hamouz, M., Illingworth, J., 2005. 3D assisted face recognition: a survey of 3D imaging, modelling and recognition approaches. In: Computer Vision and Pattern Recognition, p. 114.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788–791.

Lee, D.D., Seung, H.S., 2001. Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562.

Lei, Z., Bai, Q., He, R., Li, S., 2008. Face shape recovery from a single image using CCA mapping between tensor spaces. In: Computer Vision and Pattern Recognition, pp. 1–7.

Li, X., Iyengar, S., 2015. On computing mapping of 3D objects: a survey. ACM Comput. Surv. 47 (2), 34.

Lim, C.P., Nonis, D., Hedberg, J., 2006. Gaming in a 3D multiuser virtual environment: engaging students in science lessons. Br. J. Educ. Technol. 37 (2), 211–231.

Morency, L.-P., Rahimi, A., Darrell, T., 2003. Adaptive view-based appearance models. In: Computer Vision and Pattern Recognition, vol. 1. I–803.

Newcombe, R.A., Fox, D., Seitz, S.M., 2015. DynamicFusion: reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 343–352.

Scharstein, D., Szeliski, R., 2003. High-accuracy stereo depth maps using structured light. In: Computer Vision and Pattern Recognition, vol. 1, pp. I–195.

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR, vol. 1, pp. 519–528.

Suwajanakorn, S., Kemelmacher-Shlizerman, I., Seitz, S.M., 2014. Total moving face reconstruction. In: ECCV, pp. 796–812.

Tena, J.R., De la Torre, F., Matthews, I., 2011. Interactive region-based linear 3D face models. SIGGRAPH 30 (4), 76.

Wen, Z., Huang, T.S., 2004. 3D face modeling. In: 3D Face Processing: Modeling, Analysis and Synthesis, pp. 11–17.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22 (11), 1330–1334.

Zhang, Z., 2012. Microsoft kinect sensor and its effect. MultiMedia 19 (2), 4–10.

Zhu, X., Ramanan, D., 2012. Face detection, pose estimation, and landmark localization in the wild. In: CVPR, pp. 2879–2886.