Self-Supervised Learning of Detailed 3D Face Reconstruction

Yajing Chen, Fanzi Wu, Zeyu Wang, Yibing Song, Yonggen Ling, and Linchao Bao

***Abstract*— In this article, we present an end-to-end learning framework for detailed 3D face reconstruction from a single image. Our approach uses a 3DMM-based coarse model and a displacement map in UV-space to represent a 3D face. Unlike previous work addressing the problem, our learning framework does not require supervision of surrogate ground-truth 3D models computed with traditional approaches. Instead, we utilize the input image itself as supervision during learning. In the first stage, we combine a photometric loss and a facial perceptual loss between the input face and the rendered face, to regress a 3DMM-based coarse model. In the second stage, both the input image and the regressed texture of the coarse model are unwrapped into UV-space, and then sent through an image-to-image translation network to predict a displacement map in UV-space. The displacement map and the coarse model are used to render a final detailed face, which again can be compared with the original input image to serve as a photometric loss for the second stage. The advantage of learning displacement map in UV-space is that face alignment can be explicitly done during the unwrapping, thus facial details are easier to learn from large amount of data. Extensive experiments demonstrate the superiority of our method over previous work.**

***Index Terms*— 3D face reconstruction, self-supervised learning, depth displacement, coarse-to-fine model.**

I. Introduction

RECOVERING the 3D human facial geometry from a single color image is an ill-posed problem. Existing methods typically employ a parametric face modeling framework named as 3D morphable model (3DMM) [1]. In a 3DMM there are a set of facial shapes and texture bases, which are built from real-world 3D face scans. A linear combination of these bases synthesizes a 3D face model. During the training process, a loss function is constructed to measure the difference between the input face image and the 3D face models. The linear coefficients (i.e., 3DMM parameters) can be generated by minimizing the computed loss. While conventional methods learn these coefficients via analysis-by-synthesis optimization [2], [3], recent studies have shown the effectiveness of regressing 3DMM parameters using CNN based approaches [4]–[8].

Learning to regress 3DMM parameters via CNN requires a large amount of data. For methods based on supervised learning, the ground-truth 3DMM parameters are generated by optimization-based fitting [4], [5] or synthetic data generation [11], [12]. The limitations appear that the generated ground-truth labels are not accurate and the synthetic data lacks realism. In comparison, the methods [6], [8] based on self-supervised learning[1] do not employ this process but learn directly from unlabeled face images. For example, MoFA [6] learns to regress 3DMM parameters by forcing the rendered images to have similar pixel colors as input images in facial regions. However, enforcing the pixel level similarity does not imply similar facial identities. Genova *et al.* [8] rendered images of a face from multiple views. They use a face recognition network to measure the perceptual similarity between the input faces and the rendered faces. Although the method is capable of producing 3D models resembling the faces in the input images, it ignores detailed facial characteristics and leads to unfaithful reconstructions.

In order to model facial details beyond 3DMM, a few deep learning methods have been proposed recently. For the methods [13], [14] represent 3D faces completely without using 3DMM, severely degraded results are usually obtained. More robust approaches typically represent 3D faces with detail modeling in addition to 3DMM [9], [10], [15]. For example, learned parametric correctives are employed in [9], and 3D detail maps are employed in [10], [15]. Since the learned parametric correctives [9] have very limited expressive capabilities (see Fig. 1), we advocate 3D detail maps for detail modeling. However, existing approaches employing detail maps [10], [15] rely on surrogate ground-truth detail maps computed from traditional approaches, which are error prone and limit the fidelity of the reconstruction.

In this article, we propose a two-stage framework to regress 3DMM parameters and reconstruct facial details via

Manuscript received October 25, 2019; revised March 17, 2020 and May 30, 2020; accepted July 31, 2020. Date of publication August 27, 2020; date of current version September 4, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shuicheng Yan. *(Corresponding author: Linchao Bao.)*

Yajing Chen, Yibing Song, Yonggen Ling, and Linchao Bao are with Tencent AI Lab, Shenzhen 518057, China (e-mail: jadechancyj907@gmail.com; yibingsong.cv@gmail.com; ylingaa@connect.ust.hk; linchaobao@gmail.com).

Fanzi Wu is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: fzwu@link.cuhk.edu.hk).

Zeyu Wang is with the Department of Computer Science, Columbia University, New York, NY 10027 USA (e-mail: zw2723@columbia.edu).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIP.2020.3017347

[1]We in this article do not distinguish between the term "self-supervised" and "unsupervised", as both refer to learning without ground-truth annotations in our case. We prefer the term "self-supervised".

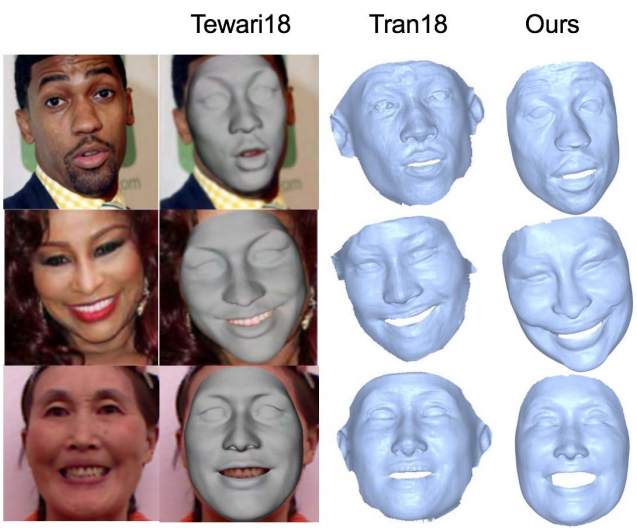


Fig. 1. Our method can produce more faithful 3D face models than state-of-the-art methods like Tewari18 [9] and Tran18 [10].

self-supervised learning. In the first stage, we use a combination of multi-level loss terms to train the 3DMM regression network. These loss terms consist of low-level photometric loss, mid-level facial landmark loss, and high-level facial perceptual loss, which enable the network to preserve both facial appearances and identities. In the second stage, we employ an image-to-image translation network to capture the missing details. We unwrap both the input image and the regressed 3DMM texture maps into UV-space. The corresponding UV maps are together sent into the translation network to obtain the detailed displacement map in UV-space. The displacement map and the 3DMM coarse model together are rendered to a final face image, which is enforced to be photometric consistent with the input face image during training. Finally, the whole network can be trained end-to-end without any annotated labels. The advantage of the detail modeling in UV-space is that all the training face images with different poses are aligned in UV-space, which facilitates the network to capture invariant details in spatial regions around facial components with large amount of data. The main contribution of our work is that we use a self-supervised approach to solve a challenging task of detailed 3D face reconstruction from a single RGB image and we achieve very high quality results. We conduct extensive experiments and analysis to show the effectiveness of our method. Compared with state-of-the-art approaches, the 3D face models produced by our method are generally more faithful to the input face images.

II. Related Work

In this section, we briefly perform a literature survey on single-view 3D face reconstruction methods. These methods can be categorized as the optimization based, the supervised learning and the self-supervised learning based methods. A more complete review can be found in [16].

A. 3DMM by Optimization

The 3D morphable model (3DMM) is proposed in [1] to reconstruct a 3D face by a linear combination of shape and texture blendshapes. These blendshapes (i.e., bases) are extracted by PCA on aligned 3D face scans. Later, Cao *et al.* [17] bring facial expressions into 3DMM and introduce a bilinear face model named FaceWarehouse. Since then, reconstructing a 3D face from an input image can be formulated as generating the optimal 3DMM parameters including shape, expression, and texture coefficients, such that the model-induced image is similar to the input image in the predefined feature spaces. Under this formulation, the analysis-by-synthesis optimization framework [2], [3], [18], [19] is commonly adopted. However, these optimization-based approaches are parameter sensitive. Non-realistic appearances exist on the generated 3D model.

B. 3DMM by Supervised Learning

Methods based on supervised learning requires ground-truth 3DMM labels by either optimization-based fitting or synthetic data rendering. Zhu *et al.* [4] and Tran *et al.* [5] use 3DMM parameters generated by optimization-based approaches as ground-truth to learn their CNN models. The performance of these methods are limited by unreliable labels. On the other hand, other approaches [7], [11], [12] try to utilize synthetic data rendered with random 3DMM parameters for supervised learning. Dou *et al.* [12] propose to use synthetic face images and corresponding 3D scans together for network learning. Richardson *et al.* [11] train a 3DMM regression network with only synthetic rendered face images. Kim *et al.* [7] show that training with synthetic data can be adapted to real data with the bootstrapping algorithm. However, the performances of these methods are limited by the unrealistic input and the 3D face models do not resemble the input images.

C. 3DMM by Self-Supervised Learning

Self-supervised methods derive supervisions by using input images without labels. MoFA [6] uses a pixel-wise photometric loss to ensure the rendered image induced by the estimated 3DMM parameters to be similar to the input image. However, the photometric loss makes the network attend to pixel-wise similarity between rendered image and input image, while the identity of the input face is ignored. Recently, Genova *et al.* [8] propose to enforce the feature similarity between the rendered image and the input image via a fixed-weight face recognition network. Their model preserves facial identity, however the low-level features are not similar (e.g., illumination, skin color, and facial expressions). This is because it is designed to predict 3DMM parameters from illumination and expression invariant facial feature embeddings instead of original face images. In the case of multi-view 3DMM regression, MVF-Net [20] learns a deep network using multi-view face images with self-supervised view alignment loss.

D. Detail Modeling Beyond 3DMM

Due to the limited expressive power of 3DMM, some approaches try to model facial details with additional layers built upon 3DMM. Examples following such type of modeling include the depth maps [15] or bump maps [10], and trainable

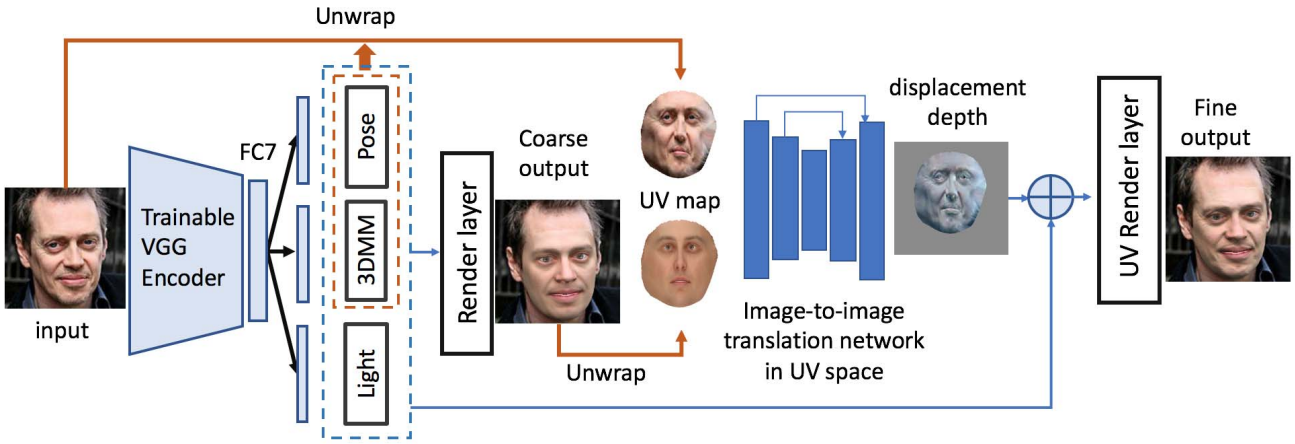


Fig. 2. Proposed pipeline. We use a 3DMM encoder to transform an input face image into a latent code vector to regress the 3DMM parameters. We unwrap both the input image and the reconstructed 3D model into UV space and estimate a displacement depth map. Then, the 3DMM-based coarse model and the displacement depth map are used to generate a 3D face model with fine details.

corrective models [9]. Besides, some other work employs non-parametric 3D representations [13], [14] to gain more degrees of freedoms, but usually are less robust than the methods built upon 3DMM.

In this article, we focus on 3D representations with a coarse 3DMM model and an additional detail-layer. In the coarse model, different from MoFA [6] and Genova *et al.* [8], we combine the use of low-level photometric loss and high-level perceptual loss to provide multi-level supervision for the coarse model. For the detail-layer representation, we prefer detail maps rather than trainable corrective models [9] due to more expressive power. Note that existing detail map based approaches [10], [15] are all based on supervised learning with surrogate ground-truth detail maps computed with traditional methods, while our approach is completely unsupervised. We use the unwrapped input image and 3D model in the UV space as inputs to the neural network, and make the network learn the detailed information from the difference between real and rendered images in an aligned space. Different from Guo *et al.* [21] that used RGBD data as model inputs and learns the per-vertex normal displacement with UV maps, we use RGB images as inputs and build a UV render layer that builds dense correspondence between UV space and the rendered image space. These differences are vital for better face reconstruction.

III. Proposed Method

In this section, we first give an overview of our method, and then explain the details of each module.

A. Framework Overview

Figure 2 illustrates the pipeline of the proposed framework. It consists of two modules. The first one is the 3DMM regression module and the second one is the detail modeling module. The 3DMM regression module learns to predict the 3DMM parameters from an input image with a trainable encoder network and a non-trainable differentiable renderer, which is similar to MoFA [6]. The detail modeling module employs an image-to-image translation network to predict a displacement depth map in UV space from the unwrapped input image and the regressed 3DMM texture UV map. The displacement depth map is then added back to the regressed 3DMM-based coarse model to get the final detailed 3D model. Finally, a differentiable UV renderer enables the whole learning process to be self-supervised by comparing the differences between the final rendered output and the input image.

The training process consists of two stages. In the first stage, we train the 3DMM regression module without the detail modeling module using two types of self-supervised losses including pixel-level photometric loss [6] and perceptual-level identity loss [8]. In the second stage, we fix the weights in the 3DMM regression module and train the detail modeling module using the pixel-level photometric loss, as well as additional smoothness losses and regularization terms. The whole training process does not rely on any 3D supervision and is fully self-supervised.

B. 3DMM Regression Module

We employ the 3DMM model including identity and expression parameters:

$$s = \overline{s} + B_{id}x_{id} + B_{exp}x_{exp}, \tag{1}$$

where $\overline{s}$ is the vector format of the mean 3D face model, B_{id} and B_{exp} are the identity bases and the expression bases from [1] and [17], respectively. The 3D reconstruction is formulated as regressing 3DMM parameters x_{id} and x_{exp} in Eq. (1).

Our 3DMM regression module employs a similar framework with existing methods [6], [8]. It takes a color face image as input and transforms it progressively to the latent code vector using multiple convolutional layers and nonlinear activations. Specifically, we adopt the VGG-Face [22] structure in the 3DMM encoder. As we notice the feature representation discrepancies between 2D face images and 3D face models, we randomly initialize the network parameters and train them

from scratch. During the training process, we project the output 3D face model into a 2D face image. The loss functions are mainly designed to measure the difference between the projected face and the input face. The total loss function for training the 3DMM-based coarse model is denoted as:

$$\mathcal{L}_{\text{coarse}} = w_1 \cdot \mathcal{L}_{\text{pixel}} + w_2 \cdot \mathcal{L}_{\text{lm}} + w_3 \cdot \mathcal{L}_{\text{id}} + w_4 \cdot \mathcal{R}_{\text{param}}, \tag{2}$$

where $\mathcal{L}_{\text{pixel}}$ is the photometric loss, $\mathcal{L}_{\text{lm}}$ is the landmark consistency loss, $\mathcal{L}_{\text{id}}$ is the perceptual identity loss, and $\mathcal{R}_{\text{param}}$ is the 3DMM parameter regularization term. The weights $\{w_1, w_2, w_3, w_4\}$ control the influence of each term and are set as constant values. The details are as follows.

1) Photometric Loss: The photometric loss is set to measure the pixel-wise difference between the input face image and the rendered face image. We denote the input face image as I and the rendered face image as I^R. The loss function is defined as:

$$\mathcal{L}_{pixel} = \frac{1}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} ||I_{i,j} - I^R_{i,j}||_2, \tag{3}$$

where $\mathcal{M}$ deontes the visible pixels on the I^R and (i, j) is the location of each visible pixel. We compute the photometric loss by averaging the $L_{2,1}$-distances for all visible pixels.

2) Landmark Consistency Loss: The landmark consistency loss measures the $L2$-distance between the 68 detected landmarks in the input face image and the rendered locations of the 68 key points in the 3D mesh. The detection of 2D landmarks is explained in [23]. The loss function is defined as:

$$\mathcal{L}_{lm} = \frac{1}{N} \sum_{i=1}^{N} ||p_i - p_i^R||_2^2, \tag{4}$$

where p_i is the i-th landmark position in the input face image, p_i^R is the corresponding i-th landmark position in the rendered face image, and $N = 68$ is the number of landmarks. The landmark consistency loss effectively controls the pose and expression of the 3D face model.

3) Perceptual Identity Loss: The perceptual identity loss reflects the perception similarity between two images. We send both the input face image and the rendered face image into the VGG-face recognition network [22] for feature extraction. We denote the extracted CNN features of the input face image as $\phi(I)$, the features of the rendered face image as $\phi(I^R)$. The perceptual consistency loss is defined as:

$$\mathcal{L}_{id} = ||\phi(I) - \phi(I^R)||_2^2, \tag{5}$$

where ϕ is the parameters of the VGG-face network [22] and is kept fixed during the training process.

4) Regularization Term: We propose a regularization term for the 3DMM parameters. Since the values of the 3DMM parameters are subject to normal distribution, we have to prevent their values from deviating from zeros too much. Otherwise, the 3D faces reconstructed from the parameters are distorted. The regularization term is:

$$\mathcal{R}_{param} = \omega_s ||x_{id}||^2 + \omega_e ||x_{exp}||^2, \tag{6}$$

where ω_s and ω_e are weighting parameters.

C. Detail Modeling Module

The detail modeling module is an image-to-image translation network in UV space. It consists of an encoder-decoder network with skip connections. The input image and the reconstructed coarse 3D face model are unwrapped into two UV texture maps of the same resolution. These two UV maps are concatenated and the invisible regions are masked out based on the estimated 3D poses. Then, the concatenated UV maps are fed into the encoder-decoder network. The network produces a displacement depth map. This map is added to the UV position map of the 3DMM-based coarse model to generate a refined UV map, which is wrapped back to a 2D face image by the UV render layer. We compare the output 2D image of the detail modeling network and the input image with a pixel-level photometric loss. During training, we use smoothness loss and regularization terms together with the photometric loss. The smoothness loss and regularization terms are set on the displacement depth maps to reduce both artifacts and distortions in the reconstruction process.

The total loss function of the detail modeling network is:

$$\mathcal{L}_{\text{fine}} = \omega_p \cdot \mathcal{L}_{\text{pixel}} + \omega_s \cdot \mathcal{L}_{\text{smooth}} + \omega_d \cdot \mathcal{L}_{\text{disp}}, \tag{7}$$

where $\mathcal{L}_{\text{pixel}}$ is the photometric loss to measure the pixel-wise difference between the input image and rendered 2D face image from UV renderer. The $\mathcal{L}_{\text{smooth}}$ term is the smoothness loss, and $\mathcal{L}_{\text{disp}}$ is the regularization term on displacement map. The weights $\{w_p, w_s, w_d\}$ are constant values to balance the influence of each loss term. We will introduce the smoothness loss and the regularization terms below:

1) Smoothness Loss: We propose the smoothness loss on both the UV displacement normal map and the displacement depth map to ensure similar representation of the neighboring pixels on these maps. Another advantage of the smoothness loss is that it ensures the robustness to mild occlusions. The smoothness loss can be written as:

$$\mathcal{L}_{\text{smooth}} = \sum_{i\in\mathcal{V}_{UV}} \sum_{j\in\mathcal{N}(i)} w_{sn}||\Delta n(i) - \Delta n(j)||^2 + w_{sz}||\Delta z(i) - \Delta z(j)||^2, \tag{8}$$

where $\Delta n(i)$ is the difference measurement on pixel i in the UV map. It computes the pixel distance between the original UV normal map (i.e., the vertex normal computed from coarse 3D model) and the UV normal map integrated with the displacement depth map. Similarly, $\Delta z(i)$ computes the pixel distance between the original displacement depth map and the updated displacement depth map. The $\mathcal{V}_{UV}$ are vertices in the UV space and $\mathcal{N}(i)$ is the neighborhood of vertex i with a radius of 1. The $\Delta n(i)$ measures the difference between the UV normal map before and after adding displacement map. The weights w_{sn} and w_{sz} are used to combine these two smoothing losses and they are set as 20 and 10.

2) Regularization Term: We propose the regularization terms on both the displacement depth map and the displacement normal map to reduce severe depth changes, which may introduce distortion in face on the 3D mesh. The regularization

term can be written as:

$$\mathcal{L}_{\text{disp}} = \sum_i w_{dn}||\Delta n(i)||^2 + w_{dz}||\Delta z(i)||^2, \quad (9)$$

where w_{dn} and w_{dz} are set to 0.5 and 0.01, respectively.

D. Camera View

The pose parameter in the proposed model is 7D, including scale f, rotation angles(in rads) r_x, r_y, r_z, and translation t_x, t_y, t_z. We apply orthogonal projection to project the 3D vertices into 2D. We denote the vertex in 3D as $\mathbf{v}$, the projection operation as Π, the projected 2D points as $\mathbf{p}$, respectively. Then we have:

$$\mathbf{p} = \Pi(f\mathbf{R}\mathbf{v} + \mathbf{t}), \quad (10)$$

where $\mathbf{R}$ is the rotation matrix computed by r_x, r_y, r_z, and $\mathbf{t} = (t_x, t_y, t_z)^T$ is the translation vector.

E. Rendering Layer

The rendering layer is a modification to [8]. We use spherical harmonics as our lighting model instead of the Phong reflection model. And orthogonal projection is applied here.

F. UV Render Layer

The UV render layer takes two inputs. One is a coarse UV position map that is built by unwrapping the coarse face mesh. The other is a predicted displacement depth map in UV space from the detailed modeling module. Using these two inputs, the UV render layer first computes the detail UV position map by adding the displacement to the coarse UV position map. Then a final output triangle mesh can be generated by connecting neighboring pixels in the detail UV position map. With the pose, lighting and texture parameters estimated in 3DMM regression module, a final image can be rendered. The whole rendering process is differentiable.

IV. Experiments

In this section, we provide more implementation details and conduct extensive experiments to demonstrate the effectiveness of our method. For implementation details, please refer to our code at: https://github.com/cyj907/unsupervised-detail-layer.

A. Implementation Details

We train our model on the CelebA dataset [24]. Before training, we use a landmark detector [23] to exclude failure samples that are not faces. Then, we separate the remaining images into two parts. The first part is the training dataset which contains 162,129 images, out of which we keep 1,000 images for validation. The second part is the testing set which contains 19,899 images. The architecture of the 3DMM regression module is VGG-Face [22]. For the detail modeling module, we employ an image-to-image translation architecture similar to pix2pix [26]. We randomly initialize all the weights in our model and train them from scratch.

TABLE I

Point-to-Plane Error on the MICC Florence Dataset [27] for 3DMM-Based Self-Supervised Learning Approaches

Method	Condition		
	Indoor Cooperative	PTZ Indoor	PTZ Outdoor
MoFA [6]	1.38 ± 0.35	1.27 ± 0.29	1.28 ± 0.27
Genova18 [8]	1.41 ± 0.37	1.34 ± 0.37	1.26 ± 0.31
Ours-3DMM	$\mathbf{1.35 \pm 0.31}$	$\mathbf{1.27 \pm 0.24}$	$\mathbf{1.25 \pm 0.21}$

The weighting parameters $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ controlling the total loss for the coarse model in Eq. (2) are set as $\{1.3, 1.0, 1.5, 20.0\}$. We set the weights $\{\omega_s, \omega_e, \omega_t\}$ in the regularization terms in Eq. (6) as $\{1.3, 1.0, 1.3\}$. The weights $\{\omega_p, \omega_s, \omega_d\}$ of the total loss for the detail modeling module in Eq. (7) are set as $\{1.0, 10.0, 10.0\}$. The training in the first stage for the coarse model uses an initial learning rate as 0.0001, decaying every 5000 steps at rate 0.9. The learning rate for training the detail modeling module in the second stage is set to 0.002, decaying every 5000 steps at rate 0.98. The batch size is set to be 10. We adopt Adam optimizer to train the network on NVIDIA Tesla M40 for over $200,000$ steps for the coarse model and $20,000$ steps for the detail model.

B. Evaluation on 3DMM Regression

1) Shape Analysis: We evaluate the accuracy of the 3DMM regression of the shap on the MICC Florence dataset [27]. In this dataset, videos are taken on 53 subjects under three different conditions. These three conditions are defined as Indoor Cooperative, PTZ Indoor and PTZ Outdoor. The ground truth 3D scans are provided for 52 out of the 53 people. We used each video frame as the network input. Before evaluation, we remove the frames where the faces are not detected by the landmark detector [23]. The 3D shape model for each video sequence is obtained using the average of the 3DMM shape parameters in the remaining video frames. We follow the procedures mentioned in [8] to compute the point-to-plane error between the predicted 3D face models and ground truth scans. Table I lists the comparison of our results to two state-of-the-art 3DMM-based self-supervised learning approaches [6], [8]. Our method outperforms existing methods by combining low-level photometric loss and high-level perceptual identity loss.

We further show some visual comparisons of the results in Fig. 3 on two other datasets, the CelebA [24] and the LFW [25] datasets. Compared with MoFA [6] and Genova18 [8], our method is able to generate more faithful results. Note that our shape results captures more personalized facial characteristics compared to MoFA, while are more faithful than Genova18. For example, the generated face by Genova18 [8] in row 3 in Fig. 3 is too short along the vertical axis compared with the corresponding input image, while our result is more faithful. When focusing on the texture, we notice that MoFA [6] tends to generate smooth texture. The results from Genova18 [8] are more realistic but does not show sufficient color distinction between people from different races. In contrast, our method shows more color diversity for individuals.

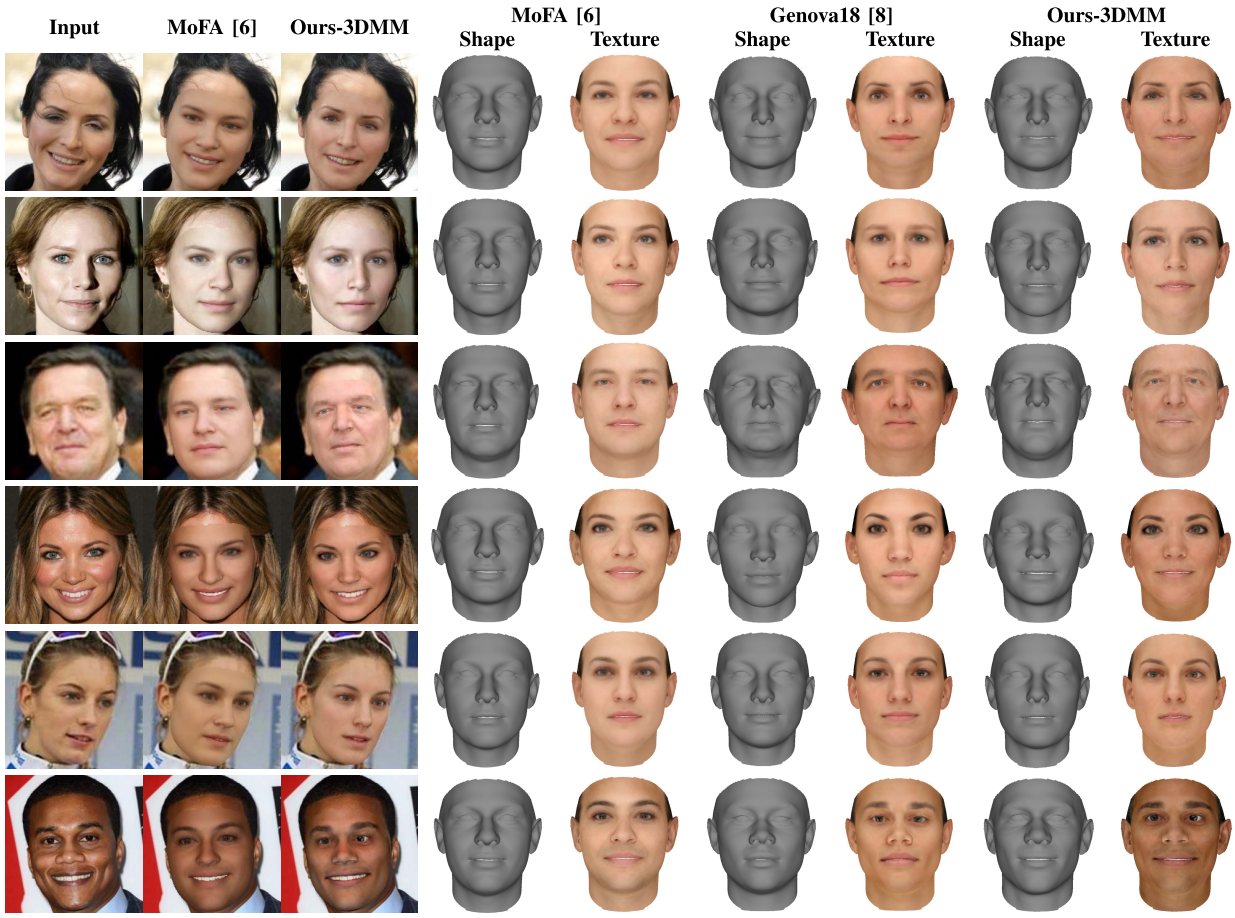


Fig. 3. Results of 3DMM regression on the CelebA [24] and LFW [25] datasets. For a fair comparison to Genova18 [8], The facial expressions obtained by MoFA [6] and our method are set to neutral.

TABLE II
MEAN AND STANDARD DEVIATION OF POINT-TO-POINT RMSE ON FACEWAREHOUSE [17] FOR 3DMM REGRESSION WITH EXPRESSIONS

Method	MoFA [6]	Ours
Error	2.26 ± 0.58	$\mathbf{1.81 \pm 0.43}$

2) Expression Analysis: We evaluate the expression reconstruction results on the FaceWarehouse dataset [17]. The dataset contains 150 subjects in 20 different expressions. We compare our method with MoFA [6]. We first use non-rigid registration to compute the vertex correspondence between FaceWarehouse data and 3DMM model. Then, we apply rigid transforms to align the predicted meshes and the ground truth scans provided in FaceWarehouse [17]. We compute the point-to-point RMSE errors (root-mean-square-error) for the corresponding vertices of the two meshes. Table II shows the mean and standard deviation of the point-to-point RMSE error on the FaceWarehouse [17] dataset. Since Genova18 [8] does not estimate expression parameters, we only compare to MoFA [6]. Our method produces more accurate results than MoFA. Fig. 4 shows some visual results of the two methods. When the commonly-seen expression (e.g., smiling with visible teeth) appears on the subjects of the input image as shown in the first row, both MoFA and our method are effective to reconstruct the 3D model. Meanwhile, the 3D model generated by our method contains more identity-specific details. When some uncommon expression appears (e.g., pouted mouths) on the second row, MoFA does not reconstruct the 3D model effectively while our method does. When the expression is

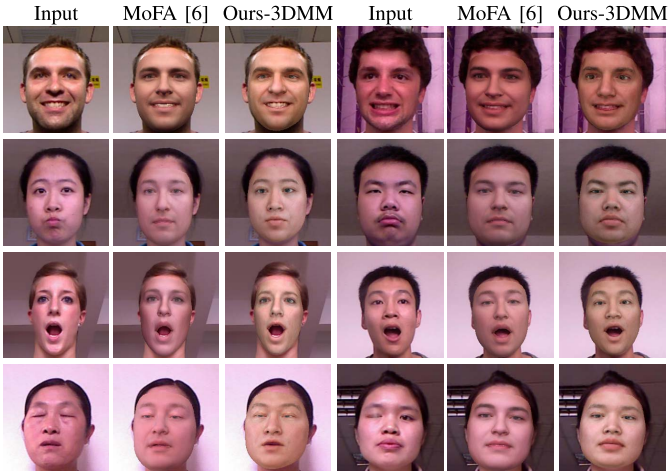


Fig. 4. 3DMM regression results of facial expressions on the FaceWarehouse dataset [17] for MoFA [6] and our method.

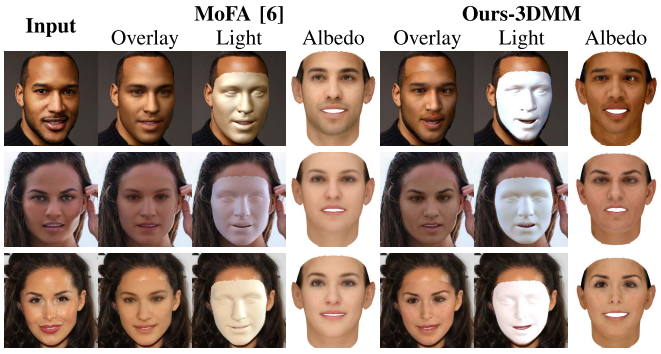


Fig. 5. Lighting and albedo results for MoFA [6] and our 3DMM regression.

extreme (e.g., largely-open mouths and closed eyes) as shown in the last row, neither of these two methods performs well. However, our method still performs favorably against MoFA.

3) Lighting and Albedo: Fig. 5 shows the visualization of the albedo and lighting reconstruction results from MoFA and our 3DMM regression. We set the meshes as white for a clear display. Though the overall color looks similar between these two results, the lighting and albedo are different. The colors in the overlay of MoFA [6] are mostly from lighting, which leads to smooth and fair albedo. In comparison, the albedo of our method is more faithful to the input faces.

C. Evaluation on Final Reconstruction

We first show some examples in Fig. 9 to demonstrate the differences between the coarse model obtained using our 3DMM regression and the fine model obtained using our full model. We can see that the fine model can retain more details and express more vivid facial characteristics than the 3DMM-based coarse model.

We now show visual comparisons of our method to state-of-the-art methods [10], [14], [21] for detailed 3D face reconstruction in Figs. 6 and 7. We observe that the 3D faces generated by Tran18 [10] are often noisy, where high frequency information are spread all over the meshes regardless of the input images. For example, on the third row of Fig. 6, the mouth region on the input face is smooth with salient texture, but the mouth of Tran18 is noisy. Meanwhile, the wrinkles on the cheeks are not reconstructed well according to the input image. Similar phenomena appear on other 3D face models which are not faithfully representing the input faces. The 3D models generated by Sela17 [14] are sometimes distorted, where the results can not be regarded as faces. In comparison, our method effectively generates 3D models preserving the global shape and structure. Fig. 8 shows several additional examples of the close-ups. While Guo19 [21] (Fig. 7) has nice shape reconstruction results, our method reconstructs meshes with more details and are more faithful to input images.

We perform quantitative evaluation on the MICC Florence dataset [27] and the Face Recognition Grand Challenge V2 (FRGC2) dataset [28]. In MICC, we evaluate the shape reconstruction precision and in FGRC2 we evaluate the depth estimation errors. We first evaluate the shape reconstruction precision on MICC dataset. As we aim to model facial details,

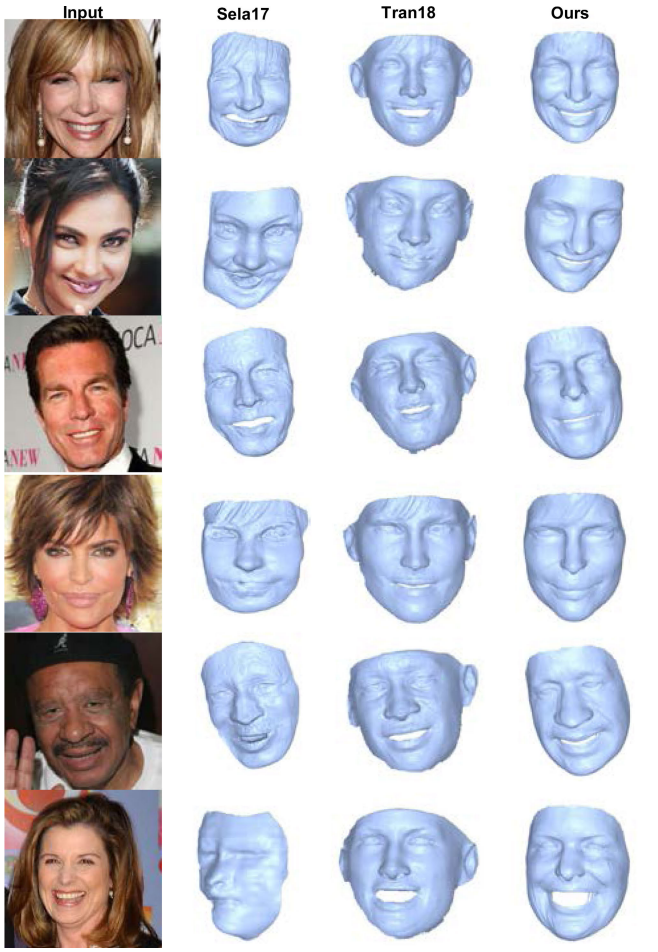


Fig. 6. Comparison to Sela17 [14] and Tran18 [10] of detailed reconstruction.

Fig. 7. Comparison to Guo19 [21] of detailed reconstruction.

we select frontal video frames of each subject for evaluation. The frontal frames only exist in the Indoor Cooperative condition. We compute the point-to-point error between the reconstructed 3D faces and the ground truth scans. We first crop the ground truth scan to 95mm around the tip of the nose. Then, we run ICP algorithm with isotropic scale to find an alignment between the ground truth and the reconstruction before computing the point-to-point distances.

Table III shows the reconstruction error on MICC dataset compared with state-of-the-art methods [10], [14]. The lower error demonstrates that our method can reconstruct fine details with higher accuracy and stability than the other methods. During the evaluation process, Sela17 [14] fails to reconstruct 5 subjects, while Tran18 [10] and our method successfully

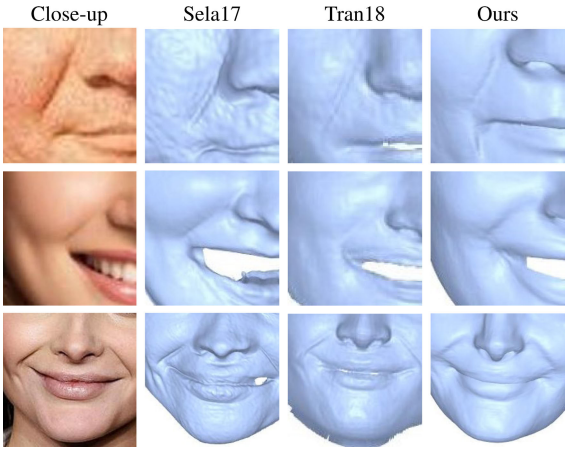


Fig. 8. Close-ups of detailed reconstruction results.

TABLE III
POINT-TO-POINT ERROR ON MICC FLORENCE [27] FOR DIFFERENT DETAILED RECONSTRUCTION APPROACHES

Method	Sela17 [14]	Tran18 [10]	Ours
Error(mm)	3.19 ± 0.58	2.61 ± 0.66	$\mathbf{2.41 \pm 0.62}$

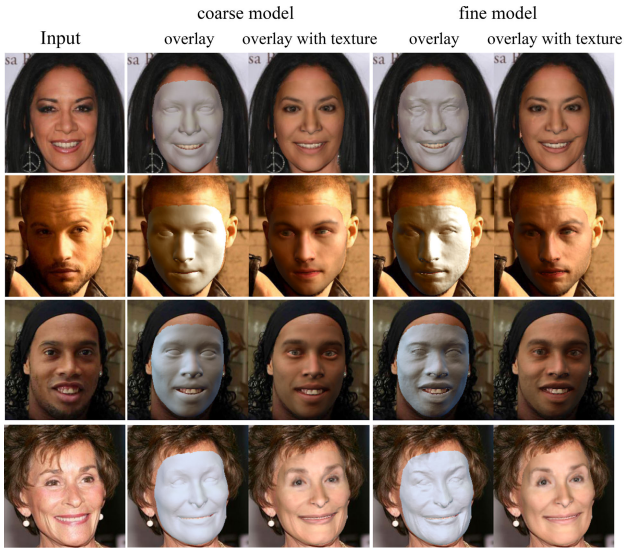


Fig. 9. Visual comparison of the coarse model (3DMM regression) and fine model (detailed reconstruction), both obtained using our method.

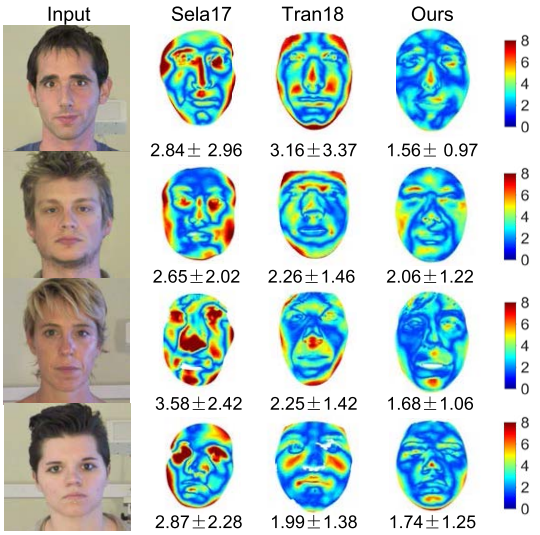


Fig. 10. Examples of error maps for detailed reconstruction.

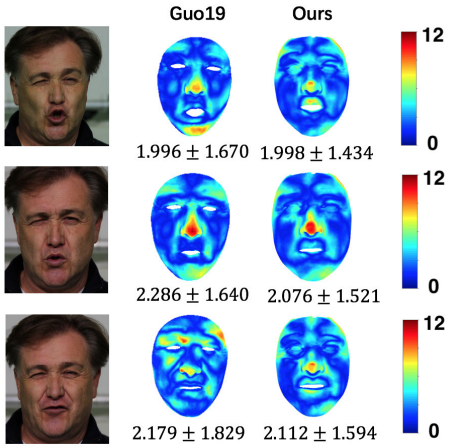


Fig. 11. Examples of error maps for detailed reconstruction.

TABLE IV
DEPTH ERROR ON FRGC2 [28] FOR DIFFERENT DETAILED RECONSTRUCTION APPROACHES

Method	Sela17 [14]	Tran18 [10]	Ours
Error(mm)	10.55 ± 2.34	6.73 ± 1.85	$\mathbf{5.05 \pm 1.44}$

reconstruct all subjects. Fig. 10 shows some error maps and Fig. 12 shows the individual errors. Sela17 is not robust compared with Tran18 and our method with higher error and standard deviation metrics. On the contrary, our method stably produces state-of-the-art results.

Besides MICC, we also evaluate on the FRGC2 datasets where the depth of the input face images are estimated. The Face Recognition Grand Challenge V2 (FRGC2) dataset [28] includes 4,950 high resolution images of 688 identities with corresponding depth maps. We evaluate the depth estimation results generated by all the methods on the FRGC2. To calculate the depth error, we first scale the depth estimation of each method to fit the ground truth depths in min-max ranges. Then the mean distance between the two depth maps at valid pixel positions provided by a fixed binary mask are computed as depth error. Table IV shows the depth estimation results for these methods, while Fig. 13 displays the corresponding histogram. our method achieve lowest mean and standard deviation in depth errors compared with the other approaches. The low depth errors indicate that the proposed model generate 3D faces with higher accuracy.

Fig. 11 shows several error map examples of our method and Guo19 [21] on FaceCap [29]. The dataset consists of stereo captures and the corresponding 3D reconstructions, which are used as ground truth data. our method achieves comparable results with Guo19 in 3D face detail reconstruction, though both approaches have large error on noses.

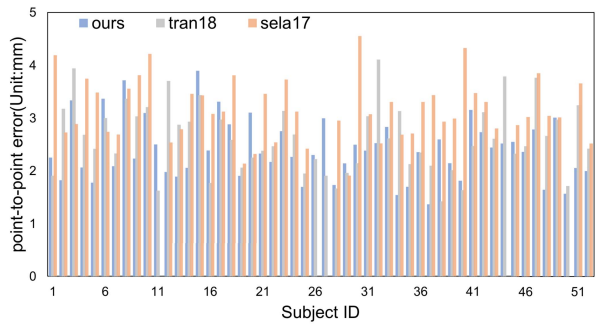


Fig. 12. Barplots on MICC for different detailed reconstruction approaches.

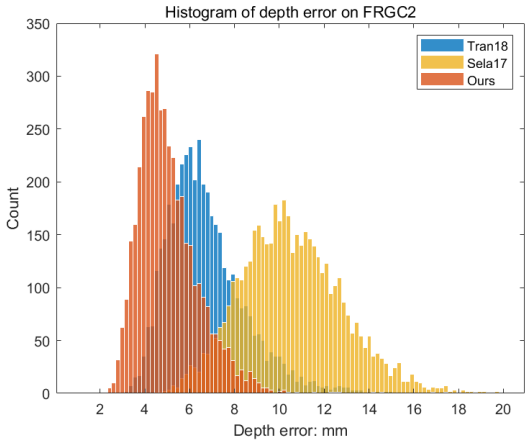


Fig. 13. Depth error histogram on FRGC2 [28] for different approaches.

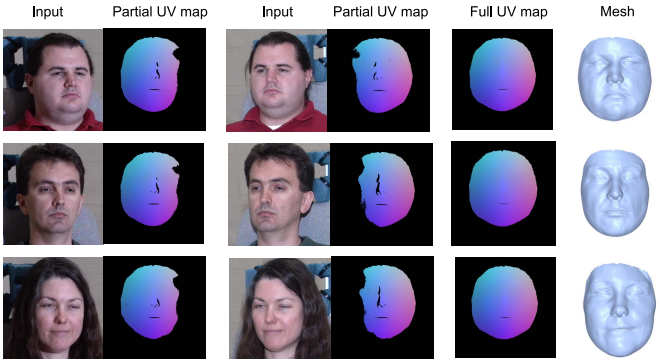


Fig. 14. Though our method is trained with single-view images, it can be adopted for multi-view reconstruction by merging two partial UV position maps into a full UV position map.

D. Application

An additional advantage of employing UV position maps [30] for reconstruction is that it enables easier integration of 3D reconstructions from different views of a same face. The UV maps from different views can be easily combined by a simple blending in UV-space. Thus the combined full UV map can represent a complete 3D face model that are visible in different views. The induced detailed 3D reconstruction are more complete compared to depth map based representations. Fig. 14 shows several examples of blending two partial UV maps from two views of a same face. In practice, to handle difficult lighting conditions or blurry faces, some preprocessing approach can be used [31].

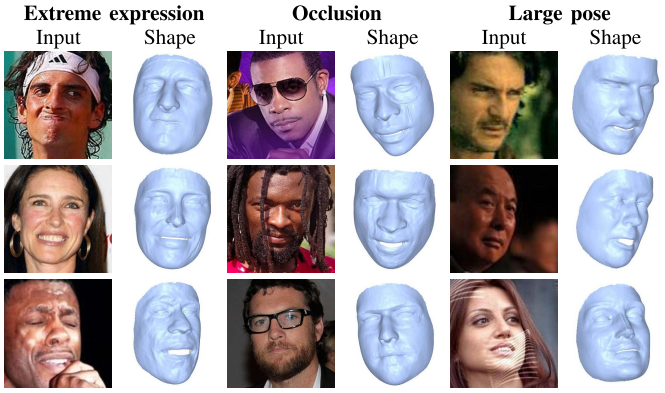


Fig. 15. Limitations of our method. The proposed model have troubles dealing with partial occlusion, extreme poses and expressions.

E. Limitation Analysis

Fig. 15 shows some failure cases of the proposed model under scenarios including large expression, occlusion and large pose. Since the proposed model are not trained with specific policy to deal with these conditions, the reconstruction quality can not be guaranteed if these situation occur.

V. Conclusion

We propose a detailed 3D face reconstruction framework with self-supervised learning. We first use a coarse 3DMM encoder to regress 3DMM parameters. When learning the 3DMM encoder, we incorporate measurements of multiple loss terms ranging from the pixel-wise similarity to the global facial perception. After learning the 3DMM parameters, we unwrap both input face image and 3DMM texture into UV space where all the faces are precisely aligned. The details from the inputs are effectively transferred to the 3D model as the aligned facial details facilitate the learning process. Experiments on various datasets demonstrate our method performs favorably against state-of-the-art 3D face modeling approaches.

References

[1] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1999, pp. 187–194.

[2] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, Sep. 2003.

[3] S. Romdhani and T. Vetter, "Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 986–993.

[4] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3D solution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 146–155.

[5] A. T. Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3D morphable models with a very deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5163–5172.

[6] A. Tewari *et al.*, "MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1274–1283.

[7] H. Kim, M. Zollhofer, A. Tewari, J. Thies, C. Richardt, and C. Theobalt, "InverseFaceNet: Deep monocular inverse face rendering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4625–4634.

[8] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman, "Unsupervised training for 3D morphable model regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8377–8386.

[9] A. Tewari *et al.*, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2549–2559.

[10] A. T. Tran, T. Hassner, I. Masi, E. Paz, Y. Nirkin, and G. Medioni, "Extreme 3D face reconstruction: Seeing through occlusions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3935–3944.

[11] E. Richardson, M. Sela, and R. Kimmel, "3D face reconstruction by learning from synthetic data," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 460–469.

[12] P. Dou, S. K. Shah, and I. A. Kakadiaris, "End-to-end 3D face reconstruction with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5908–5917.

[13] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3D face reconstruction from a single image via direct volumetric CNN regression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1031–1039.

[14] M. Sela, E. Richardson, and R. Kimmel, "Unrestricted facial geometry reconstruction using image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1576–1585.

[15] E. Richardson, M. Sela, R. Or-El, and R. Kimmel, "Learning detailed face reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1259–1268.

[16] M. Zollhöfer *et al.*, "State of the art on monocular 3D face reconstruction, tracking, and applications," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 523–550, May 2018.

[17] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 3, pp. 413–425, Mar. 2014.

[18] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–10, Nov. 2013.

[19] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2387–2295.

[20] F. Wu *et al.*, "MVF-net: Multi-view 3D face morphable model regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 959–968.

[21] Y. Guo, J. Zhang, J. Cai, B. Jiang, and J. Zheng, "CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1294–1307, Jun. 2019.

[22] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–12.

[23] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial Landmarks)," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1021–1030.

[24] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[25] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, 2007.

[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[27] A. D. Bagdanov, A. Del Bimbo, and I. Masi, "The florence 2D/3D hybrid face dataset," in *Proc. Joint ACM Workshop Hum. Gesture Behav. Understand. (J-HGBU)*, 2011, pp. 79–80.

[28] P. J. Phillips *et al.*, "Overview of the face recognition grand challenge," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 947–954.

[29] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–10, Nov. 2013.

[30] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 534–551.

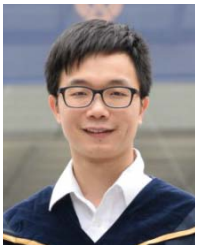
[31] Y. Song *et al.*, "Joint face hallucination and deblurring via structure generation and detail enhancement," *Int. J. Comput. Vis.*, vol. 127, nos. 6–7, pp. 785–800, Jun. 2019.

Yajing Chen received the B.S. degree from the Department of Computer Science and Technology, Shanghai Jiao Tong University, in 2015. She is currently a Researcher with Tencent AI Lab. Her research interests include 3D face reconstruction and image synthesis.

Fanzi Wu received the B.S. degree from the Department of Electronic Engineering, Tianjin University, China, in 2014. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong. Her research interest includes 3D vision.

Zeyu Wang received the B.S. degree in automation from Shanghai Jiao Tong University in 2020. He is currently a Graduate Student with Columbia University, majoring in computer science. His research interests include machine learning, computer vision, and recommend systems.

Yibing Song received the bachelor's degree from the University of Science and Technology of China in 2011 and the M.Phil. and Ph.D. degrees from the City University of Hong Kong in 2014 and 2018, respectively. He is currently a Researcher with Tencent AI Lab. His research interests include visual recognition and visual generation.



Yonggen Ling received the Ph.D. degree from the Robotics Institute, The Hong Kong University of Science and Technology, in 2017. He is currently a Senior Researcher with Robotics X, Tencent. His research interests include visual odometry, mapping, 3D reconstruction, visual-inertial fusion, SLAM, and autonomous navigation.

Linchao Bao received the M.S. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in computer science from the City University of Hong Kong in 2015. He was a Research Intern with Adobe Research from November 2013 to August 2014 and worked for DJI as an Algorithm Engineer from January 2015 to June 2016. He is currently a Principal Research Scientist with Tencent AI Lab. His research interests include computer vision and graphics.