MVF-Net: Multi-View 3D Face Morphable Model Regression

Fanzi Wu[2†*] Linchao Bao[1*] Yajing Chen[3] Yonggen Ling[1]
Yibing Song[1] Songnan Li[2] King Ngi Ngan[2,4] Wei Liu[1]

[1]Tencent AI Lab [2]The Chinese University of Hong Kong
[3]Shanghai Jiao Tong University [4]University of Electronic Science and Technology of China

Abstract

We address the problem of recovering the 3D geometry of a human face from a set of facial images in multiple views. While recent studies have shown impressive progress in 3D Morphable Model (3DMM) based facial reconstruction, the settings are mostly restricted to a single view. There is an inherent drawback in the single-view setting: the lack of reliable 3D constraints can cause unresolvable ambiguities. We in this paper explore 3DMM-based shape recovery in a different setting, where a set of multi-view facial images are given as input. A novel approach is proposed to regress 3DMM parameters from multi-view inputs with an end-to-end trainable Convolutional Neural Network (CNN). Multi-view geometric constraints are incorporated into the network by establishing dense correspondences between different views leveraging a novel self-supervised view alignment loss. The main ingredient of the view alignment loss is a differentiable dense optical flow estimator that can backpropagate the alignment errors between an input view and a synthetic rendering from another input view, which is projected to the target view through the 3D shape to be inferred. Through minimizing the view alignment loss, better 3D shapes can be recovered such that the synthetic projections from one view to another can better align with the observed image. Extensive experiments demonstrate the superiority of the proposed method over other 3DMM methods.

1. Introduction

Reconstructing 3D facial shapes from 2D images is essential for many virtual reality (VR) and augmented reality (AR) applications. In order to obtain fully-rigged 3D meshes that are necessary for subsequent steps like facial animations and editing, 3D Morphable Model (3DMM) [2] is often adopted in the reconstruction to provide a parametric representation of 3D face models. While conventional approaches recover the 3DMM parameters of given facial images through analysis-by-synthesis optimization [3, 25], recent work has demonstrated the effectiveness of regressing 3DMM parameters using convolutional neural networks (CNN) [40, 35, 32, 17, 12, 29, 28]. In spite of the remarkable progress in this topic, recovering 3DMM parameters from a single view suffers from an inherent drawback: the lack of reliable 3D constraints can cause unresolvable ambiguities, *e.g.*, the height of nose and cheekbones of a face is difficult to tell given only a frontal view.

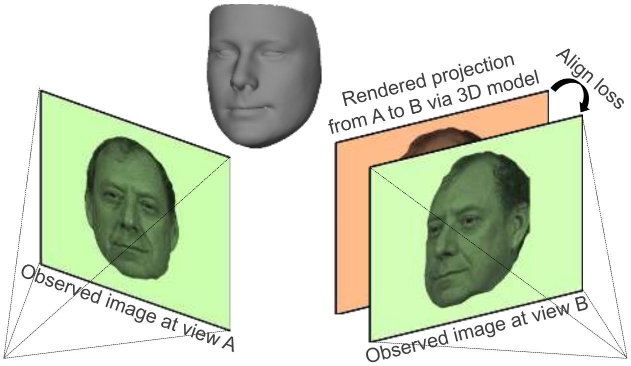

Figure 1. An illustration of the view alignment loss. The rendered projection from view A to B via the optimal underlying 3D model should align best with the image observed at view B.

A better way to reconstruct more faithful 3D shapes from 2D images is to exploit multi-view geometric constraints using a set of facial images in different views. In this case, structure-from-motion (SfM) and multi-view stereo (MVS) algorithms [9] can be employed to reconstruct an initial 3D model and then a 3DMM fitting can be performed using the 3D geometric constraints from the initial model [2]. However, the separated two steps are error-prone: the SfM/MVS step cannot utilize the strong human facial prior from 3DMM and hence its results are usually rather noisy, which further leads to erroneous 3DMM fitting. An alternative approach is to directly fit 3DMM parameters from multi-view images through analysis-by-synthesis optimization [25], but it requires a complicated, nonlinear optimization that can be difficult to solve in practice.

In this paper we propose a novel approach, which adopts an end-to-end trainable CNN to regress 3DMM parameters in the multi-view setting. Inspired by the photometric bun-

[*] Equal contributions. Emails: {wufanzi412, linchaobao}@gmail.com
[†] This work was done when Fanzi Wu was an intern at Tencent AI Lab.

dle adjustment method [6] for camera pose and 3D shape estimation in multi-view 3D reconstruction, our method is also based on the assumption that the underlying optimal 3D model should best explain the observed images in different views. That is, the photometric reprojection error between each observed image and a rendered image induced by the underlying 3D model for this view should be minimized (as illustrated in Fig. 1). To incorporate this constraint into our CNN, we sample textures from an input view using the predicted 3D model and camera pose, and then render the textured 3D model to another view to compute the loss between the rendered image and the observed image in the target view. In addition to the direct photometric loss between the two images, we propose a novel view alignment loss utilizing a differentiable dense optical flow estimator to backpropagate alignment errors, to avoid trapping into local minima during training. All the above procedures are differentiable and the whole network is end-to-end trainable. To the best of our knowledge, this is the first work that proposes an end-to-end trainable network to exploit both 3DMM and multi-view geometric constraints. We conduct extensive experiments to show the effectiveness of the proposed method.

2. Related Work

In this section, we briefly summarize the most related work to our approach. Please refer to the recent survey [41] for more detailed review.

2.1. Morphable 3D Face Model (3DMM)

Blanz and Vetter [2] introduced the 3D morphable model to represent textured 3D faces using linear combinations of a set of shape and texture bases, which is derived from collections of real 3D face scans. The model is later extended to include facial expressions by FaceWarehouse [5]. In this paper, we focus on recovering the underlying 3D shapes of human faces, hence we are only interested in regressing 3DMM parameters for shapes and expressions. We argue that more realistic textures for 3D meshes can be obtained with more advanced texture synthesis techniques [26] instead of the 3DMM texture representations.

2.2. Single-view 3DMM-based Reconstruction

Conventional methods for single-view 3DMM fitting are mostly based on analysis-by-synthesis optimization [3, 25, 10, 34, 38, 39], by constraining the data similarities like pixel colors, facial landmarks, edges, *etc*., between observed images and the synthetic images induced by 3DMM. The optimization is usually sensitive to initial conditions and parameters, and hence brittle in practice. This leads to the recent interests in regression-based approaches with deep neural networks.

Zhu *et al.* [40] proposed a cascaded CNN to regress and progressively refine 3DMM parameters, trained with supervision data generated by fitting 3DMM parameters using conventional approaches and then augmented by their proposed face profiling technique. Later, Tran *et al*. [35] presented that more discriminative results could be obtained with deeper networks and 3DMM pooling over face identities. However, both methods require supervision obtained through optimization-based 3DMM fitting techniques. Dou *et al*. [8] proposed to train the regression network using real 3D scans together with synthetic rendered face images with a 3D vertex distance loss. Richardson *et al*. [23] showed that a 3DMM regression network can be trained using only synthetic rendered face images and later Kim *et al*. [17] proposed a bootstrapping algorithm to adapt the synthetic training data distribution to match real data. Recently, Tewari *et al*. [32] and Genova *et al*. [12] demonstrated impressive results by training 3DMM regression networks using only unlabeled images with a self-supervised photometric loss and a face recognition loss, respectively.

To model detailed facial geometries beyond the representation power of 3DMM, some recent studies proposed to supplement additional geometric representations such as displacement maps [24, 36] or parametric correctives [33] besides 3DMM representations. Some other work used volumetric representations [15] or non-regular meshes [27] instead of parametric representations. These types of representations are out of the scope of this paper.

2.3. Multi-view 3DMM-based Reconstruction

In the multi-view setting, a straightforward solution [14] for 3DMM-based reconstruction is to first perform traditional multi-view 3D reconstruction [9] and then fit a 3DMM using the reconstructed 3D model as constraints. However, the separated two steps are error-prone: the SfM/MVS step cannot utilize the strong human facial prior from 3DMM and hence its results are usually rather noisy, which further leads to erroneous 3DMM fitting. Dou *et al*. [7] recently proposed to address the problem using deep convolutional neural networks (CNNs) together with recurrent neural networks (RNNs). They used RNNs to fuse identity-related features from CNNs to produce more discriminative reconstructions, but multi-view geometric constraints are not exploited in their approach. Notice that there are some other 3DMM-based methods in multi-image settings [22], but in these work each input image is dealt individually, which is not the same as our multi-view setting.

3. Approach

3.1. Overview

We employ an end-to-end trainable CNN to regress 3DMM parameters from multiple facial images for the same person in different views. In order to establish multi-view geometric constraints like conventional multi-view 3D reconstruction approaches [9], for now we assume the facial images are taken at the same time under the same lighting condition. Later we will illustrate that our approach is able

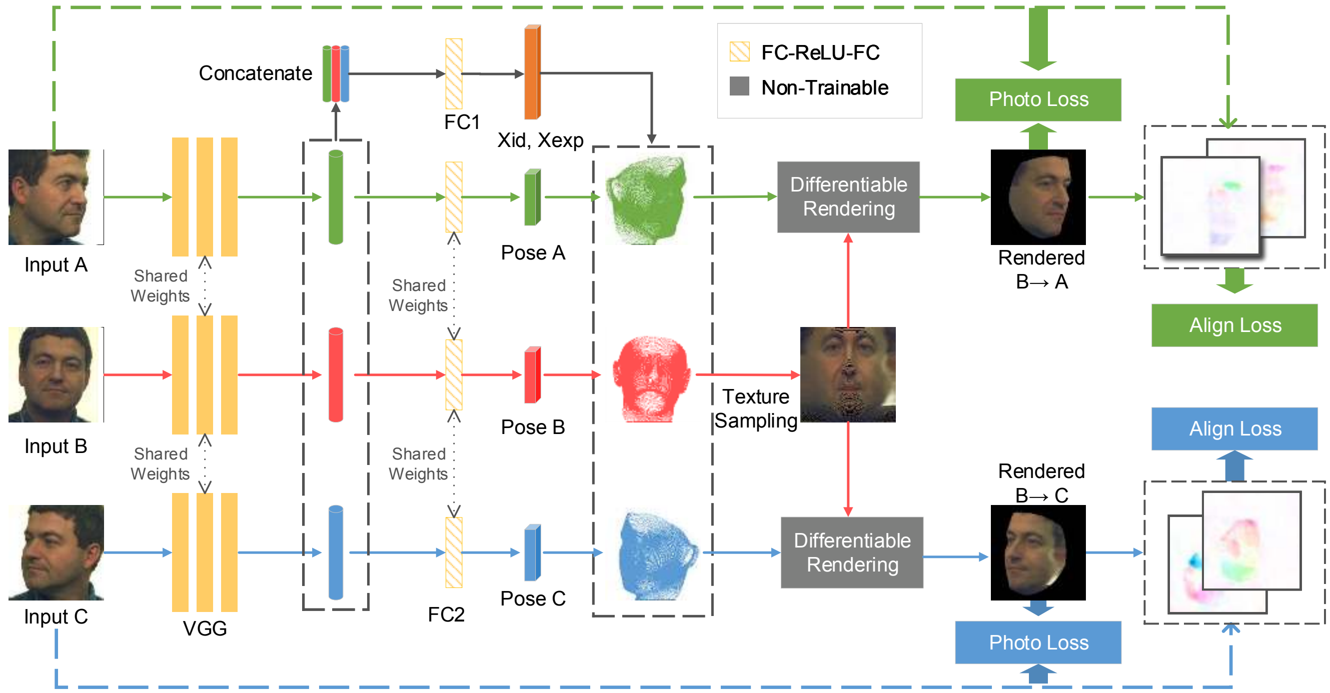

Figure 2. An overview of the proposed model.

to handle inputs with lighting variance. For simplicity, we adopt three-view setting to describe our approach. Note that it can be easily generalized to other number of input views.

Fig. 2 illustrates the overview of our proposed model in the case of three input views. We learn features from each input image by a shared weight CNN, and then concatenate the features together to regress a set of 3DMM parameters for the person. Differently, we regress the pose parameters for each input view from its individual features (Sec. 3.3). With the pose parameters and 3DMM parameters, we are able to render a textured 3D face model from each input image by sampling textures from the image (Sec. 3.4). Note that in the three-view setting, there will be three textured 3D face models, with the same underlying 3D shape but with different textures. After obtaining the rendered 3D face models of different views, we then project each of them to a different view from the view where the textures are sampled (Sec. 3.5). For instance, we project the 3D model with textures sampled from image at view A to view B. Then we can compute losses between the projected image with the input image at the target view. We will present the details of the adopted losses in Sec. 3.6. Please be noted that the rendering layer is non-parametric yet differentiable, like that in previous self-supervised approaches [32, 12], and the gradients can thus be backpropagated to the trainable layers.

3.2. Model

The 3DMM parameters to be regressed in this work include both identity and expression parameters like [40]. A 3D face model s can be represented as

$$\mathbf{s} = \bar{\mathbf{s}} + E_{\text{id}}\mathbf{x}_{\text{id}} + E_{\text{exp}}\mathbf{x}_{\text{exp}}, \tag{1}$$

where $\bar{\mathbf{s}}$ is the vector format of the mean 3D face model, E_{id} and E_{exp} are the identity basis from BFM 2009 [19] and expression basis from FaceWarehouse [5] respectively, $\mathbf{x}_{\text{id}}$ and $\mathbf{x}_{\text{exp}}$ are the corresponding 199-dimension identity vector and 29-dimension expression vector to be regressed.

To project 3D model onto 2D image plane, we employ the weak perspective projection model. Given a 3D point $\mathbf{v}$, its 2D projection can be computed with a set of camera pose parameters $\mathcal{P}$ as follows

$$\Pr(\mathbf{v}, \mathcal{P}) = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \end{bmatrix} \cdot R \cdot \mathbf{v} + \mathbf{t}, \tag{2}$$

where f is the scaling factor, R is the rotation matrix, and $\mathbf{t}$ is the 2D translation $[t_x, t_y]^{\mathrm{T}}$. Since the rotation matrix R can be minimally parameterized as three Euler angles α, β, γ, the pose to be regressed contains 6 parameters in total, which reads as $\mathcal{P} = \{f, \alpha, \beta, \gamma, t_x, t_y\}$.

3.3. Parametric Regression

We denote the three-view input images as $\mathbf{I}_A$, $\mathbf{I}_B$, and $\mathbf{I}_C$. We assume $\mathbf{I}_B$ is the image taken from the frontal view, $\mathbf{I}_A$ and $\mathbf{I}_C$ are taken from the left and right views respectively. Note that we do not need the images to be taken from precise known view angles. Each input image is sent through several convolutional layers (borrowed from VGG-Face [30] in our implementation) and pooled to a 512-dimentional feature vector. Then a set of pose parameters $\mathcal{P} = \{f, \alpha, \beta, \gamma, t_x, t_y\}$ is regressed for each view via two fully-connected layers. The three 512-dimentional feature vectors are concatenated together to regress the 228-dimentional 3DMM parameters $\mathcal{X} = \{\mathbf{x}_{\text{id}}, \mathbf{x}_{\text{exp}}\}$ (199 for identity and 29 for expression) using another two fully-connected layers. Note that for each set of inputs, we regress one $\mathcal{X}$ and three pose parameters $\mathcal{P}_A$, $\mathcal{P}_B$, and $\mathcal{P}_C$. The networks to extract features and regress pose parameters for the three views have shared weights.

3.4. Texture Sampling

With the predicted 3DMM parameters $\mathcal{X}$, as well as the known identity basis E_{id} and expression basis E_{exp}, we can compute the 3D face model using Eq. (1). Three different texture maps can be obtained by sampling textures from each image individually using its own pose parameters predicted by the network. For each vertex $\mathbf{v}$ of the 3D model, we apply Eq. (2) to project the vertex to the image plane and fetch the texture color from each input image for the vertex using differentiable sampling scheme, as adopted in Spatial Transformer Networks [16]. For 3D point within a triangle on the mesh, we utilize barycentric interpolation to get its texture color from surrounding vertices. Note that since the texture sampling scheme does not handle occlusions, the textures sampled for occluded regions in each image are erroneous. We deal with this problem using visibility masks which will be detailed in Sec. 3.5. Suppose now we have obtained three differently textured 3D models in this step.

3.5. Rendered Projection and Visibility Masks

The textured 3D model can be projected to an arbitrary view to render a 2D image, via the differentiable rendering layer introduced in [12]. For example, given a 3D model with textures sampled from image $\mathbf{I}_A$, we can render it to the view of $\mathbf{I}_B$ using the pose parameters $\mathcal{P}_B$, which we denote as $\mathbf{I}_{A\to B}$. Formally, for any 3D point $\mathbf{v}$ on the mesh surface (including points within triangles), the color of its projected pixel in the rendered image can be computed as

$$\mathbf{I}_{A\to B}[\Pr(\mathbf{v}, \mathcal{P}_B)] = \mathbf{I}_A[\Pr(\mathbf{v}, \mathcal{P}_A)], \tag{3}$$

where we use $[\cdot]$ to denote the pixel selection in an image. In practice, the rendering is implemented through rasterization on the target image plane, that is, denoting an arbitrary pixel in the target image as $\mathbf{u}$, then Eq. (3) can be written as

$$\mathbf{I}_{A\to B}[\mathbf{u}] = \mathbf{I}_A[\Pr(\Pr^{-1}(\mathbf{u}, \mathcal{X}, \mathcal{P}_B), \mathcal{P}_A)], \tag{4}$$

where we use $\Pr^{-1}(\cdot)$ to denote the back projection from a 2D point to 3D space. Note that since the back projection is essentially a ray in 3D space, we need the 3D surface of the face model, which can be induced by 3DMM parameters $\mathcal{X}$, in order to locate the back projection ray to a 3D point. Thus the back projection operator $\Pr^{-1}(\cdot)$ in the above equation takes $\mathcal{X}$ as input in addition to camera pose $\mathcal{P}_B$. Ideally, with the optimal underlying 3D model and camera poses, the observed image $\mathbf{I}_B$ should be the same as the rendered image $\mathbf{I}_{A\to B}$ in non-occluded facial regions,

$$\mathbf{I}_{A\to B}(\mathcal{X}^*, \mathcal{P}_B^*, \mathcal{P}_A^*)[\mathbf{u}] \equiv \mathbf{I}_B[\mathbf{u}], \quad \text{for } \mathbf{u} \in \mathcal{M}, \tag{5}$$

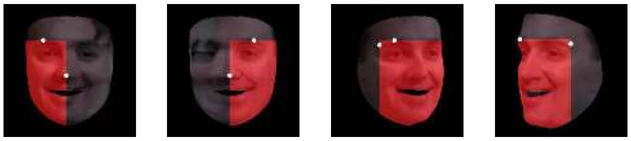
(a) (b) (c) (d)

Figure 3. Visibility masks for rendered images: (a) $\mathbf{I}_{A\to B}$; (b) $\mathbf{I}_{C\to B}$; (c) $\mathbf{I}_{B\to A}$; (d) $\mathbf{I}_{B\to C}$. The dark regions are excluded using 3D landmarks on nose tip and eyebrows (the white points).

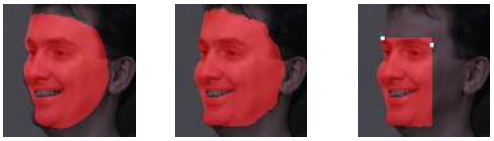
(a) Initial mask (b) After filtering (c) After cropping

Figure 4. The mask processing for an observed image. The initial mask is essentially the texture sampling regions. It is then filtered using a joint edge-preserving filering with the image as guidance. The final mask (c) is obtained by excluding occluded regions using 2D detected landmarks on eyebrows (the white points).

where $\mathcal{M}$ denotes the set of pixels in non-occluded facial regions. We will use this assumption to design our self-supervised losses in Sec. 3.6.2.

Till now, we are discussing the rendered projections without considering occlusions. To exclude occluded facial regions, we employ visibility masks to obtain $\mathcal{M}$. Note that Eq. (5) is for the ideal case, where the visibility mask is the same for both rendered image and observed image. In practice, with imperfect 3DMM and pose parameters, we need different masks for rendered image and observed image to enforce the photometric consistency (see Sec. 3.6.2 for details). For rendered image, we simply extract a visibility mask by excluding regions that may be occluded in other views using 3D vertices corresponding to 2D facial landmarks (the correspondences between 3D vertices and 68-points 2D facial landmarks are provided by [40]). Fig. 3 illustrates an example of the visibility masks for all three views. For the observed real image, we obtain an initial mask using the texture sampling regions. Then a joint edge-preserving filtering [11] is performed on the initial mask, with the input real image as guidance, to force the edges of the mask aligned well with the facial regions of the input image. Finally the regions that may be occluded in other views are excluded using 2D detected landmarks, similar to the processing of masks for rendered images (see Fig. 4). Note that for the frontal observed image, there are two different visibility masks when viewed from left and right sides, respectively. We denote the set of pixels in the corresponding masks as $\mathcal{M}_B^{(A)}$ and $\mathcal{M}_B^{(C)}$.

3.6. Losses and Training

In order to obtain a good initialization and avoid trapping into local minima, we first pretrain the CNN using supervised labels on the 300W-LP dataset [40], where ground-truth 3DMM and pose parameters are obtained via conventional 3DMM fitting algorithms and multi-view images are generated by face profiling augmentation. After the pretraining converges, we then perform self-supervised training on the Multi-PIE dataset [13], where multi-view facial images are taken in controlled indoor settings. The training losses are detailed in the following section.

3.6.1 Supervised Pretraining

In supervised pretraining, the ground-truth landmarks, 3DMM and pose parameters are provided. In the dataset 300W-LP, for each real facial image, several synthetic rendered views are generated. During the training stage, we randomly select a set of multi-view images for each face, which contains left, frontal, and right views. We use ground-truth landmarks, 3DMM and pose parameters as supervision, as well as regularizations on 3DMM parameters. The supervised training loss is

$$L_{\text{sup}} = \lambda_1 L_{\text{landmark}} + \lambda_2 L_{\text{pose}} + \lambda_3 L_{\text{3DMM}} + \lambda_4 L_{\text{reg}}, \quad (6)$$

where L_{landmark} is the landmark alignment loss similar to [32], L_{pose} and L_{3DMM} are L2 losses between predictions and ground-truths, L_{reg} is the regularization loss on 3DMM parameters also similar to [32]. The weighting $\lambda_{1,2,3,4}$ are hyper-parameters controlling the trade-off between losses.

3.6.2 Self-supervised Training

During the self-supervised training stage, we enforce the photometric consistency between observed image and synthetic rendered image to incorporate multi-view geometric constraints. From Eq. (5) we derive the photometric loss

$$L_{\text{photo}}(\mathbf{I}_B, \mathbf{I}_{A\to B}) = \sum_{\mathbf{u}\in\mathcal{M}_B^{(A)}\cup\mathcal{M}_{A\to B}} \|\mathbf{I}_B[\mathbf{u}] - \mathbf{I}_{A\to B}[\mathbf{u}]\|_2^2, \quad (7)$$

where $\mathcal{M}_B^{(A)}$ and $\mathcal{M}_{A\to B}$ are the sets of pixels in visibility masks for $\mathbf{I}_B$ (viewed from the left side) and $\mathbf{I}_{A\to B}$ respectively. Note that here we use the union of $\mathcal{M}_B^{(A)}$ and $\mathcal{M}_{A\to B}$ such that misalignment errors can be taken into considerations. Unfortunately, we find that using only the photometric loss could lead to bad alignment in practice. The reason is that the pixels within facial regions are similar to each other such that mis-matching easily happens. In order to increase the reliability of the dense correspondences between observed image and rendered image, we introduce an additional novel alignment loss into the training.

We employ a differentiable dense optical flow estimator to compute the flow between observed image and rendered image, and then use the sum of squared flow magnitudes at all pixels as the alignment loss. Since the dense optical flow estimator tends to estimate smoothed flow fields, individual mis-matchings can be largely suppressed. For example, to enforce the photometric consistency between $\mathbf{I}_B$ and $\mathbf{I}_{A\to B}$, we compute the alignment loss as

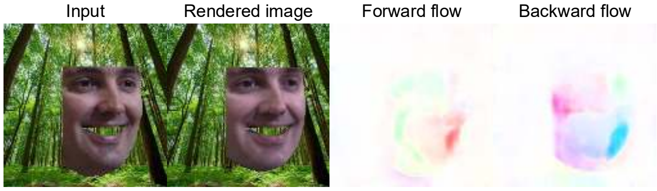

Figure 5. Optical flows between observed and rendered images.

$$L_{\text{align}}(\mathbf{I}_B, \mathbf{I}_{A\to B}) = |\mathbf{F}(\mathbf{I}_B, \mathbf{I}_{A\to B})| + |\mathbf{F}(\mathbf{I}_{A\to B}, \mathbf{I}_B)|, \quad (8)$$

where $\mathbf{F}(\cdot)$ denotes the optical flow estimator. Note that here bi-directional optical flows are employed. Besides, in order to reduce the distractions of optical flow estimation errors in uninterested regions, we fill in the the regions outside visibility masks with textures whose flow can be easily estimated (see Fig. 5 for an example).

For the three-view setting, we compute the photometric loss and alignment loss between 4 pairs of images: $(\mathbf{I}_B, \mathbf{I}_{A\to B})$, $(\mathbf{I}_B, \mathbf{I}_{C\to B})$, $(\mathbf{I}_A, \mathbf{I}_{B\to A})$, and $(\mathbf{I}_C, \mathbf{I}_{B\to C})$. Additionally, to increase the training stability, we also adopt the landmark loss L_{landmark} during self-supervised training, where the landmarks are detected via a state-of-the-art landmark detector from [4] automatically. To sum, the self-supervised training loss is

$$L_{\text{self-sup}} = \lambda_5 L_{\text{landmark}} + \lambda_6 L_{\text{photo}} + \lambda_7 L_{\text{align}}, \quad (9)$$

where both photometric loss L_{photo} and alignment loss L_{align} are computed from the above 4 pairs of images. The hyper-parameters $\lambda_{5,6,7}$ control the trade-off between losses.

4. Experiments

In this section, we first introduce the datasets, evaluation metrics, and implementation details for conducting the experiments (Sec. 4.1 and 4.2). We then demonstrate the effectiveness of the proposed approach with extensive ablation studies in Sec. 4.3. Finally, quantitative and qualitative comparisons to state-of-the-art single-view 3DMM-based approaches are presented in Sec. 4.4.

4.1. Datasets and Metrics

Training Datasets. 1) Our supervised pretraining is performed on 300W-LP dataset [40], which contains over 60,000 images derived from 3,837 face images by varying poses using face profiling synthesis method [40]. Ground-truth landmarks, 3DMM and pose parameters are provided by the dataset. We sample triplet consists of a front, left, and right view image from 300W-LP dataset using the provided yaw angles, which results in 140k training triplets in total. 2) Our self-supervised training is performed on Multi-PIE dataset [13], which contains over 750,000 images recorded from 337 subjects using 15 cameras in different directions under various lighting conditions. We take frontal-view images as anchors and randomly select side-view images (left or right) to get 50k training triplets and 5k testing triplets, where the subjects in testing split do not appear in training split. Note that whether an image is in frontal, left, or right view can be determined by the provided camera ID.

Evaluation Datasets. 1) We mainly perform quantitative and qualitative evaluations on the MICC Florence dataset [1], which consists of 53 identities of persons with neutral expression and ground-truth 3D scans are available. Each person contains three videos of "indoor-cooperative", "indoor", and "outdoor" respectively. To experiment with the multi-view setting addressed in this paper, we manually select a set of multi-view frames for each person, such that his/her expressions are consistent in different views. Since it is difficult to select such sets of frames in the "outdoor" videos, we only perform evaluations on the "indoor-cooperative" and "indoor" videos. 2) Qualitative evaluations are further performed on Color FERET dataset [20, 21] and MIT-CBCL face recognition database [37], where multi-view facial images are available.

Evaluation Metrics. In the quantitative evaluations on MICC dataset, we follow the evaluation metrics from [12], which compute point-to-plane L2 errors between predict 3D models and ground-truth 3D scans. Here, we abandon subjects of ID 2 and 27 as their ground-truth 3D scans are flawed and also excluded in other work [32, 12].

4.2. Implementation Details

We use PWCNet [31] as our differentiable optical flow estimator in the self-supervised training step. Note that during our training, the weights of PWCNet is fixed. We crop input images according to bounding boxes of facial landmarks (either ground-truth or detected with [4]) and resize them to 224×224. To augment the training data, we add random shift with 0∼0.05 of input size to the bounding box. We adopt Adam [18] as the optimizer. The batchsize is set to 12. The supervised pretraining is trained on 300W-LP for 10 epoches with learning rate 1e-5, and the self-supervised training is trained on Multi-PIE for 10 epoches with learning rate 1e-6. The default weights for balancing losses are set to $\lambda_1 = 0.1$, $\lambda_2 = 10$, $\lambda_3 = 1$, $\lambda_4 = 1$, $\lambda_5 = 1$, $\lambda_6 = 10$, and $\lambda_7 = 0.1$. We set different weights for different loss terms to make their numbers in a similar scale. The weights λ_1 and λ_7 are set to relatively smaller values as they represent pixel distances. The weights λ_2 and λ_6 are set to larger values as pose parameters and pixel values of input images are normalized to $[0, 1]$.

4.3. Ablation Study

We conduct a series of experiments on MICC dataset to demonstrate the effectiveness of each component in our approach. Table 1 shows the mean errors of different versions of our model. From the results we observe that, compared with the supervised pretrained model (v1), the self-supervised trained model with only photometric loss (v2) reduces the mean error by 0.026 for "indoor-cooperative" but none for "indoor" images, while the model with only alignment loss (v3) reduces the mean error by 0.039 for "indoor-cooperative" and 0.007 for "indoor" images, which is a moderate improvement over photometric loss. Combining the photometric loss and alignment loss (v4) gives the best results, an error reduction of 0.046 and 0.024.

Ours	Self-supervised Loss			INC		IND	
	L_{landmark}	L_{photo}	L_{align}	Mean	Std	Mean	Std
v1	–	–	–	1.266	0.297	1.252	0.285
v2	√	√	×	1.240	0.258	1.252	0.245
v3	√	×	√	1.227	0.248	1.245	0.240
v4	√	√	√	**1.220**	**0.247**	**1.228**	**0.236**

Table 1. Mean error of our approach on the MICC dataset. The versions: v1 for the supervised pretrained model; v2-v4 for the self-supervised trained model with different losses.

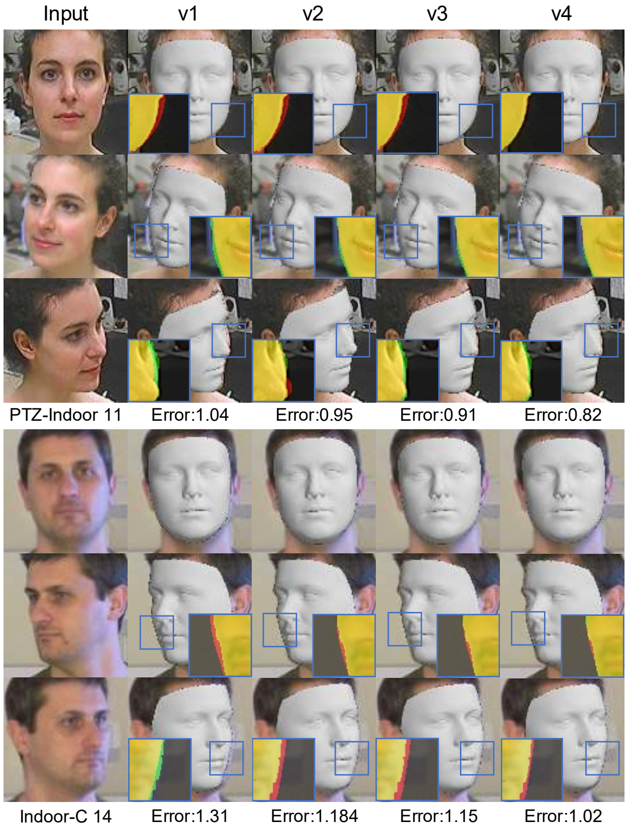

Figure 6. Visual examples of ablation study on the MICC dataset. The meanings of the colors in the close-ups are as follows. Red: the projection area from 3D to 2D exceeds the observed facial boundary. Green: the projection area is smaller than the facial area. Yellow: overlap between projection and facial areas.

Fig. 6 shows two visual examples of the ablation study. From the close-ups we can clearly observe the performance improvements from v1 to v4. Specifically, take the right-side view of the bottom person as an example, we can ob-

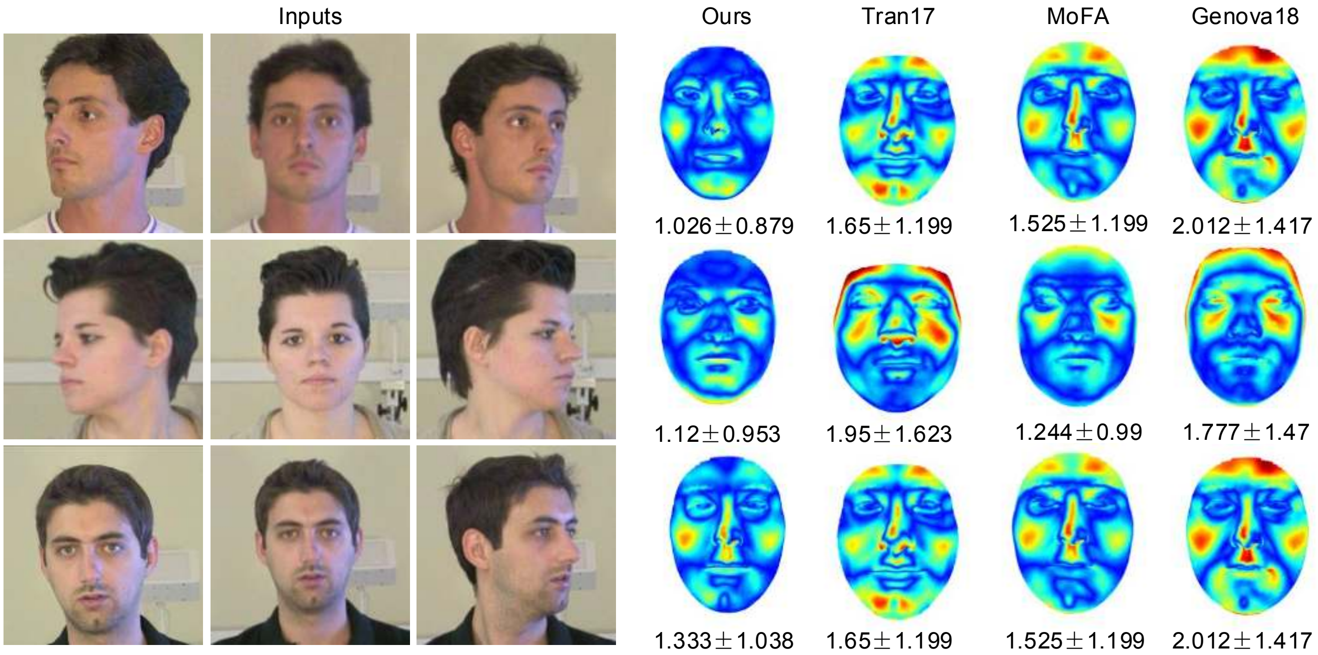

Figure 7. Examples of error map comparison on the MICC dataset.

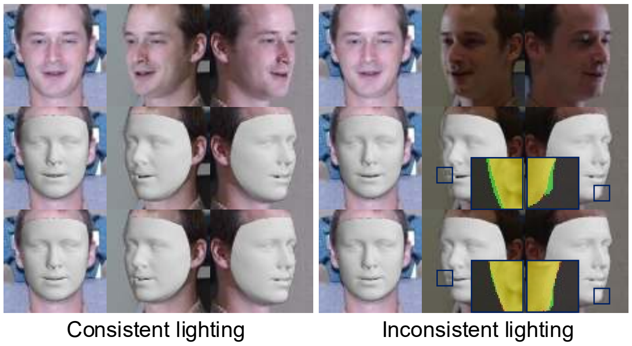

Figure 8. Experiments on inconsistent lighting conditions across views. First row: input. Second row: results obtained with only the photometric loss. Third row: results obtained with both photometric loss and alignment loss.

Method	INC		IND	
	Mean	Std	Mean	Std
Tran *et al.* [35]	1.443	0.292	1.471	0.290
Tran *et al.* + pool	1.397	0.290	1.381	0.322
Tran *et al.* + [22]	1.382	0.272	1.430	0.306
MoFA [32]	1.405	0.306	1.306	0.261
MoFA + pool	1.370	0.321	1.286	0.266
MoFA + [22]	1.363	0.326	1.293	0.276
Genova *et al.* [12]	1.405	0.339	1.271	0.293
Genova *et al.* + pool	1.372	0.353	1.260	0.310
Genova *et al.* + [22]	1.360	0.346	1.246	0.302
Ours	**1.220**	**0.247**	**1.228**	**0.236**

Table 2. Comparison of mean error on the MICC dataset.

serve that the facial silhouette of the input face is flat, while in the result from v1 it seems a little bit plump and it becomes much more flatter in the result from v4. The same trends can be found in other examples by inspecting the alignment of 3D models to the facial silhouettes.

We further conduct studies under varying lighting conditions across views to demonstrate the effectiveness of the proposed alignment loss to handle lighting changes. Fig. 8 shows an example. In this example, when the lighting is consistent across the three views (left), the model trained with only photometric loss performs almost as good as the model trained with both photometric loss and alignment loss. But when the lighting is inconsistent across the views, the result obtained from only photometric loss is much worse than that from both losses. The reason why the alignment loss is robust to lighting changes is due to the optical flow estimator, which is already trained to deal with lighting changes of input images.

4.4. Comparisons to State-of-the-art Methods

We first compare our results on MICC dataset with state-of-the-art single-view 3DMM reconstruction methods. To evaluate single-view methods on our three-view evaluation triplets for each person, we first use their model to predict 3D model a 3D model for each input image. Then three different evaluation settings are employed to ensure fair comparisons. The first one is to calculate the point-to-plane errors for each 3D model and then average the errors. The second one is to average the three predicted 3D models in a triplet and then compute the point-to-plane errors between the pooled 3D model with ground-truth model (shown in Table 2 as "+pool" entries). The third one is to compute the weighted average of three predicted 3D models as [22] and then compute the point-to-plane errors (shown in Table

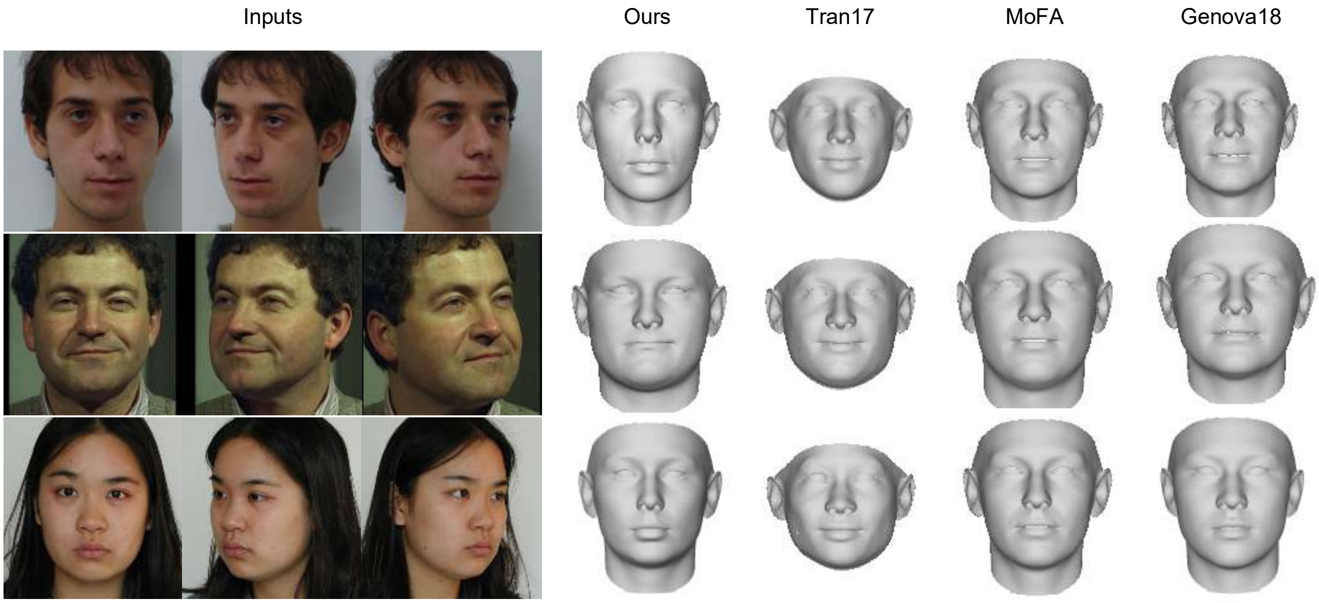

Figure 9. Examples of visual comparison with the other methods. More examples are in supplementary materials.

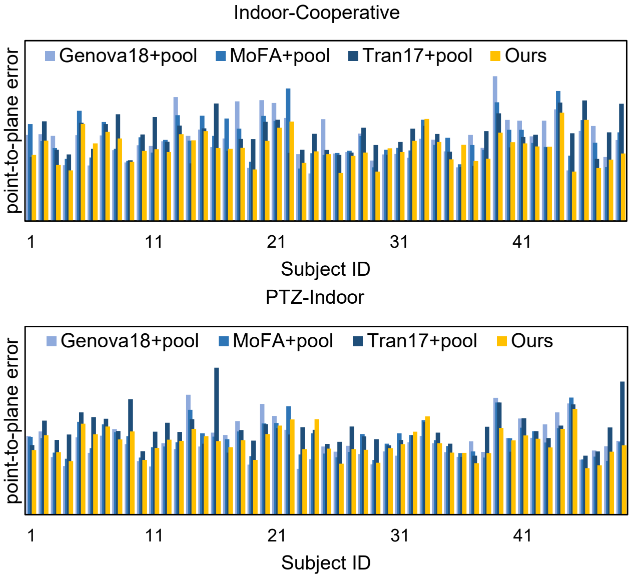

Figure 10. Detailed comparisons for each subject in MICC dataset.

2 as "+[22]" entries). Table 2 shows the mean errors of the comparison. The proposed method outperforms all single-view methods in both settings. Fig. 10 shows the detailed numerical comparisons for each subject in the dataset. Several examples of the comparison of detailed error maps are presented in Fig. 7.

We further present some visual comparisons using images from other datasets such as Color FERET dataset [20, 21] and MIT-CBCL face recognition database [37], where multi-view facial images are available. Fig. 9 shows several examples of the visual comparisons to single-view methods in neutral expression. Fig. 11 shows several examples of the visual comparisons to MoFA in different facial expressions. The superiority of our method over single-view methods can be observed in these comparisons.

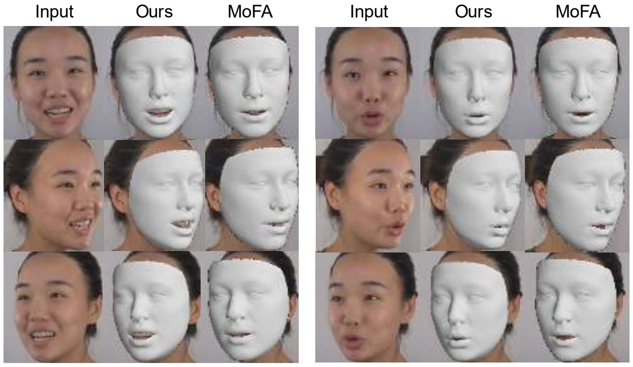

Figure 11. Examples of visual comparison to MoFA in different facial expressions. Our method can produce more accurate shapes and expressions. More examples are in supplementary materials.

5. Conclusions

In this paper, we presented a novel approach to regress 3DMM parameters from multi-view facial images with an end-to-end trainable CNN. Different from single-view 3DMM-based CNNs, our approach explicitly incorporates multi-view geometric constraints as the photometric loss and alignment loss between different views with the help of rendered projections via predicted 3D models. The alignment loss was computed via a differentiable dense optical flow estimator, which enables the flow errors to backpropagate to the 3DMM parameters to be predicted. The effectiveness of the proposed approach was validated through the extensive experiments. Our study essentially explores model-based multi-view reconstruction using deep learning, which we believe will inspire more future research.

References

[1] Andrew D. Bagdanov, Alberto Del Bimbo, and Iacopo Masi. The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*, J-HGBU 11, page 7980, New York, NY, USA, 2011. ACM. 6

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, pages 187–194. ACM, 1999. 1, 2

[3] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003. 1, 2

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *ICCV*, volume 1, page 4, 2017. 5, 6

[5] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014. 2, 3

[6] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *CVPR*, pages 1486–1493, 2014. 2

[7] Pengfei Dou and Ioannis A Kakadiaris. Multi-view 3d face reconstruction with deep recurrent neural networks. *Image and Vision Computing*, 80:80–91, 2018. 2

[8] Pengfei Dou, Shishir K Shah, and Ioannis A Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. In *CVPR*, pages 21–26, 2017. 2

[9] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 1, 2

[10] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph.*, 32(6):158–1, 2013. 2

[11] Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In *ACM Trans. Graph*, volume 30, page 69. ACM, 2011. 4

[12] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *CVPR*, pages 8377–8386, 2018. 1, 2, 3, 4, 6, 7

[13] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 5

[14] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph*, 34(4):45, 2015. 2

[15] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *ICCV*, pages 1031–1039. IEEE, 2017. 2

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 4

[17] Hyeongwoo Kim, Michael Zollhöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Christian Theobalt. Inversefacenet: Deep monocular inverse face rendering. In *CVPR*, pages 4625–4634, 2018. 1, 2

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *AVSS*, pages 296–301. IEEE, 2009. 3

[20] P Jonathon Phillips, Hyeonjoon Moon, Syed A Rizvi, and Patrick J Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1090–1104, 2000. 6, 8

[21] P Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998. 6, 8

[22] Marcel Piotraschke and Volker Blanz. Automated 3d face reconstruction from multiple images using quality measures. In *CVPR*, pages 3418–3427, 2016. 2, 7, 8

[23] Elad Richardson, Matan Sela, and Ron Kimmel. 3d face reconstruction by learning from synthetic data. In *3DV*, pages 460–469. IEEE, 2016. 2

[24] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *CVPR*, pages 5553–5562. IEEE, 2017. 2

[25] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *CVPR*, volume 2, pages 986–993. IEEE, 2005. 1, 2

[26] Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. Photorealistic facial texture inference using deep neural networks. In *CVPR*, volume 3, 2017. 2

[27] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *ICCV*, pages 1585–1594. IEEE, 2017. 2

[28] Yantao Shen, Hongsheng Li, Tong Xiao, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Deep group-shuffling random walk for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2265–2274, 2018. 1

[29] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Person re-identification with deep similarity-guided graph neural network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 486–504, 2018. 1

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3

[31] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 6

[32] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, volume 2, page 5, 2017. 1, 2, 3, 5, 6, 7

[33] Ayush Tewari, Michael Zollhfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Prez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *CVPR*, June 2018. 2

[34] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, pages 2387–2395, 2016. 2

[35] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *CVPR*, pages 1493–1502. IEEE, 2017. 1, 2, 7

[36] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard Medioni. Extreme 3D face reconstruction: Seeing through occlusions. In *CVPR*, 2018. 2

[37] Benjamin Weyrauch, Bernd Heisele, Jennifer Huang, and Volker Blanz. Component-based face recognition with 3d morphable models. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, pages 85–85. IEEE, 2004. 6, 8

[38] Fanzi Wu, Songnan Li, Tianhao Zhao, and King Ngi Ngan. Model-based face reconstruction using sift flow registration and spherical harmonics. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1774–1779. IEEE, 2016. 2

[39] Fanzi Wu, Songnan Li, Tianhao Zhao, King Ngi Ngan, and Lv Sheng. 3d facial expression reconstruction using cascaded regression. *arXiv preprint arXiv:1712.03491*, 2017. 2

[40] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *CVPR*, pages 146–155, 2016. 1, 2, 3, 4, 5

[41] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. In *Computer Graphics Forum*, volume 37, pages 523–550. Wiley Online Library, 2018. 2