Riggable 3D Face Reconstruction via In-Network Optimization

Ziqian Bai[1]   Zhaopeng Cui[2*]   Xiaoming Liu[3]   Ping Tan[1*]

[1] Simon Fraser University   [2] State Key Lab of CAD & CG, Zhejiang University

[3] Michigan State University

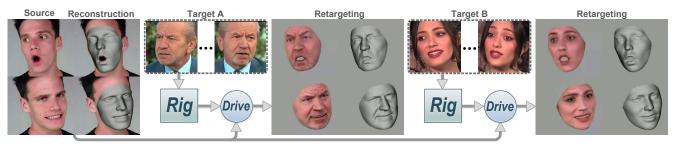{ziqianb, pingtan}@sfu.ca, zhpcui@zju.edu.cn, liuxm@cse.msu.edu

Figure 1. Our method estimates personalized face rigs and per-image reconstructions from monocular images with good reconstruction quality and supports video retargeting to different actors. The code is available at https://github.com/zqbai-jeremy/INORig.

Abstract

This paper presents a method for riggable 3D face reconstruction from monocular images, which jointly estimates a personalized face rig and per-image parameters including expressions, poses, and illuminations. To achieve this goal, we design an end-to-end trainable network embedded with a differentiable in-network optimization. The network first parameterizes the face rig as a compact latent code with a neural decoder, and then estimates the latent code as well as per-image parameters via a learnable optimization. By estimating a personalized face rig, our method goes beyond static reconstructions and enables downstream applications such as video retargeting. In-network optimization explicitly enforces constraints derived from the first principles, thus introduces additional priors than regression-based methods. Finally, data-driven priors from deep learning are utilized to constrain the ill-posed monocular setting and ease the optimization difficulty. Experiments demonstrate that our method achieves SOTA reconstruction accuracy, reasonable robustness and generalization ability, and supports standard face rig applications.

1. Introduction

3D face reconstruction has been an important research topic due to the increasing demands on 3D face understanding in fields like AR/VR, communication, games, and secu-

rity. Some approaches go beyond merely estimating static reconstructions and aim to reconstruct face rigs, which are personalized parametric models that can produce 3D faces under different expressions of a specific person. The rig can either be used on character animations such as face retargeting and voice puppetry, or on 3D face tracking serving as a personalized prior to ease the tracking difficulty.

When 3D data is available, various approaches [7,23,24] have been proposed to automatically reconstruct face rigs in the forms of blendshapes. Progress has also been made to develop more sophisticated rigs based on anatomical constraints [44] and deep neural networks [27, 45] to faithfully capture facial details. However, these methods heavily depend on the 3D data provided by specialized equipment such as dense camera/lighting arrays and depth sensors, which limits the application realms.

To release the restricted hardware requirements, methods were enhanced to work on monocular imagery. Given the ill-posedness of monocular reconstruction, algorithms usually use a low dimensional parametric face model as priors, *e.g.*, 3D morphable model (3DMM) [5] and multi-linear model [9, 43], whose parameters are estimated via the analysis-by-synthesis optimization [14, 16, 19, 49]. Additional components such as corrective basis [16], shading-based dynamic details [16, 21], image-based representation [10], as well as hair [10, 19] and other secondary components [21] are adopted to further personalize the estimated rig. However, these approaches may assume specific properties of the input, *e.g.*, requiring the subject to be static and in the neutral pose for a portion of the in-

put [10, 21]; need manual intervention [21]; and are often inefficient [16].

The recent boom in deep learning also advanced monocular 3D face reconstruction. Various learning-based methods were proposed to regress face model parameters or face shapes [30, 34, 42], learn with novel supervisions [13, 17, 32], build better face models [38–41], as well as integrate with traditional multi-view geometry [3]. Nevertheless, these methods mainly focus on static reconstructions and fail to produce personalized face rigs. Very recently, Chaudhuri *et al.* [11] used neural networks for regressing blendshape face rigs from monocular images. Despite the appealing textures produced by their method, the estimated 3D geometry, which is an important aspect for 3D reconstruction, still has considerable room for improvement.

In this paper, we propose a monocular riggable 3D face reconstruction algorithm. The riggable reconstruction consists of a personalized face rig and per-image parameters including expressions, poses, and illuminations. Our method is an end-to-end trainable network embedded with a differentiable in-network optimization. Two modules are involved. One is a neural decoder conditioned on the input images to parameterize the face rig into a latent code (termed as rig code) to control the person-specific aspects (*e.g.* identity). The other is a learnable optimization that estimates the rig code and the per-image parameters.

Our main novelty is the *integration* of deep learning and optimization for face rig. In contrast to prior static reconstruction methods [3], our riggable reconstruction can be re-animated by another face or even voices, enabling extra applications such as face retargeting and voice puppetry. Different from previous learning-based methods [11] that *directly regress* rig parameters, our in-network optimization iteratively solves rig parameters with *explicit constraints* governed by the first-principles (*e.g.* multi-view consistency, landmark alignment, and photo-metric reconstruction), achieving better geometry accuracy and good data generalization. Unlike traditional optimizations [16, 21] using hand-crafted priors, we adopt a learned deep rig model and a learned optimization to leverage deep priors to constrain the ill-posedness and ease the hardness of the optimization. Our method is able to achieve state-of-the-art (SOTA) reconstruction accuracy, reasonable robustness and generalization ability, and can be used in standard face rig applications as demonstrated in experiments.

2. Related Works

Personalized Modeling with 3D Inputs. Traditionally, blendshapes are typical choices to represent a personalized face model/rig, which are expressive 3D shapes that can be linearly combined to get novel expressions [22]. Due to the tremendous effort of manually creating blendshapes, approaches [7, 18, 23, 24] have been proposed to automate

the process by adaptively updating the initial blendshapes or adding correctives based on 3D inputs such as example meshes or depths. Progress has also been made to develop more sophisticated models. Wu *et al.* [44] proposed a local face model with anatomical constraints. Deep neural networks [27, 45] are also employed to capture person-specific geometry and appearance details. However, the application scenarios of these methods are largely limited by the dependency on 3D data, which requires specialized equipments such as dense camera/lighting arrays and depth sensors.

Our method only needs monocular RGB images as inputs, thus eliminates the dependency of bulky equipments.

Traditional Optimization. Methods were proposed to reconstruct personalized face rigs from monocular RGB data. Usually, some parametric face models, such as 3D morphable model (3DMM) [4, 5, 31] or multi-linear blendshapes [9, 43], are used as priors to constrain the ill-posed problem, while the model parameters are computed by various types of optimization [49]. Different algorithms were designed by extending this basic fitting pipeline. Ichim *et al.* [21] fit a 3DMM to a structure-from-motion reconstruction and personalize the expressions by updated blendshapes and a learned detail map regressor. Garrido *et al.* [16] proposed a multi-layer representation to reconstruct personalized face rigs from monocular RGB videos in a fully automatic fashion. People [10, 19] also added hair and other secondary components for more realistic face rigs. However, the 3DMM or multi-linear models involved usually cannot capture accurate face geometry due to their limited capacity. Though this can be alleviated by further adaptations or correctives, these methods usually contain a long pipeline with heavily engineered components, which may require manual intervention [21] and are often fragile and inefficient.

Recently, deep learning becomes a potential solution to address these issues. Our method leverages the powerful and more general data-driven priors captured by networks to constrain the ill-posedness and ease the optimization.

Learning-based Methods. Plenty of deep learning methods were designed to regress 3D shapes or face model parameters [30, 34, 42], learn with only 2D images [13, 17] and identity [32] supervisions, learn better face models from in-the-wild data [38–41], as well as integrate with traditional multi-view geometry [3]. However, most of them focus on static reconstructions instead of personalized face rigs. Very recently, Yang *et al.* [46] proposed to regress riggable displacement maps acting as textures of a bilinear blendshape model fitted by traditional optimizations. Though the displacement maps give better visual quality, they cannot address the limited capacity of linear models in terms of geometry accuracy. Chaudhuri *et al.* [11] proposed to use networks for regressing blendshape face rigs from monocular images in a self-supervised manner. Despite the appealing textures produced by their algorithm, their estimated 3D ge-
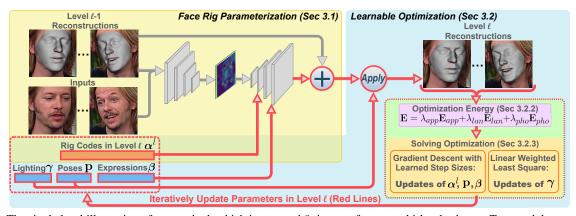
Figure 2. The single level illustration of our method, which is repeated 3 times to form a multi-level scheme. Two modules are involved: (1) **Face Rig Parameterization** that parameterizes the face rig into an optimisable latent code $\boldsymbol{\alpha}^l$ to control the person-specific aspects (*e.g.* identity) via a neural decoder; (2) An end-to-end **Learnable Optimization** to iteratively update the rig code $\boldsymbol{\alpha}^l$ and the per-image parameters including expressions, poses, and illuminations.

ometry, which is an important aspect for 3D reconstruction, still has considerable room for improvement.

Instead of direct regression, our method uses in-network optimization governed by the first-principles. This extra constraint, together with the learned deep priors, offer the potential to improve geometry accuracy and generalization, while address the limited capacity of linear face models.

3. Method

Given N monocular RGB images $\{\mathbf{I}_i\}_{i=1}^N$ of a person (*i.e.* unsynchronized images taken under different views and expressions), our method estimates riggable 3D face reconstructions composed of a personalized face rig $Rig(\cdot)$ as well as per-image parameters $\{\mathbf{x}_i = (\boldsymbol{\beta}_i, \mathbf{p}_i, \boldsymbol{\gamma}_i)\}_{i=1}^N$ including expressions $\boldsymbol{\beta}_i$, poses $\mathbf{p}_i$, and illuminations $\boldsymbol{\gamma}_i$. The per-image 3D reconstruction can be obtained by combining the estimated face rig and per-image parameters.

Our framework adopts a 3-level scheme to perform the reconstruction in a coarse-to-fine manner. For each level l, there are mainly 2 modules (see Fig. 2): (1) Face Rig Parameterization (Sec. 3.1): An image-conditioned network decoder to parameterize the face rig updates at level l into an optimizable latent code $\boldsymbol{\alpha}^l$; (2) Learnable Optimization (Sec. 3.2): An end-to-end learnable optimization to iteratively update the rig code $\boldsymbol{\alpha}^l$ and per-image parameters $\{\mathbf{x}_i\}_{i=1}^N$. Finally, our model is trained with registered ground truth 3D scans in a supervised manner (Sec. 3.3).

3.1. Face Rig Parameterization

The face rig is a parametric model that takes in an expression parameter $\boldsymbol{\beta}$ and outputs a colored 3D face mesh corresponding to the input expression, denoted as $(\mathbf{V}, \mathbf{A}) = Rig(\boldsymbol{\beta})$ where $\mathbf{V}$ is the mesh vertices and $\mathbf{A}$ is the albedo colors. To model the face rig $Rig(\cdot)$, multiple approaches have been proposed such as using a set of blendshapes [11], a neural network [45], or multi-layer representations [16].

However, these models are usually hard to be optimized [1] due to the ill-posedness of monocular 3D reconstruction. Inspired by previous works [3, 6, 37], we propose to learn a compact and optimisable latent code $\boldsymbol{\alpha}$ via the neural network to parameterize the face rig for constraining the ill-posed problem via data-driven priors.

Specifically, we design a neural decoder that takes in the latent code $\boldsymbol{\alpha}$ (termed as "rig code") and the expression parameter $\boldsymbol{\beta}$, and outputs the colored mesh $(\mathbf{V}, \mathbf{A})$. We make the decoder conditioned on the input images $\{\mathbf{I}_i\}$ as in [3, 6, 37] to better leverage the visual clues. Note that we also need some sort of initial/intermediate reconstructions $\{\hat{\mathbf{V}}_i^{old}\}$ (*i.e.* per-image meshes of level $l-1$ shown in Fig. 2) to map the image information into UV space. We will describe how to get the initial/intermediate reconstructions $\{\hat{\mathbf{V}}_i^{old}\}$ in Sec. 3.2.3. Formally, we have

$$(\mathbf{V}, \mathbf{A}) = Rig(\boldsymbol{\beta}; \boldsymbol{\alpha}, \{\mathbf{I}_i\}, \{\hat{\mathbf{V}}_i^{old}\}). \qquad (1)$$

Inside the rig, there are mainly 4 components (each has 3 levels): image feature extraction, neutral shape decoding, expression deformation decoding, and albedo decoding, which will be described one by one. For each component (except Albedo), we will firstly describe its single level version, then its generalization to the 3-level scheme. More details can be found in the supplementary material.

3.1.1 Image Feature Extraction

As shown in the top left of Fig. 3, given the input images $\{\mathbf{I}_i\}_{i=1}^N$ and initial/intermediate reconstructions $\{\hat{\mathbf{V}}_i^{old}\}_{i=1}^N$, we aim to extract a feature map in UV space F_{uv} that encodes the priors derived from visual clues for the later face rig decoding. This process is similar to the adaptive basis generation in [3], with the output linear basis replaced by F_{uv}. We use N Siamese branches to extract N feature

---
[1]Here "optimize" refers to optimizing the model parameters over current input images, but not over the whole dataset (*i.e.* training).
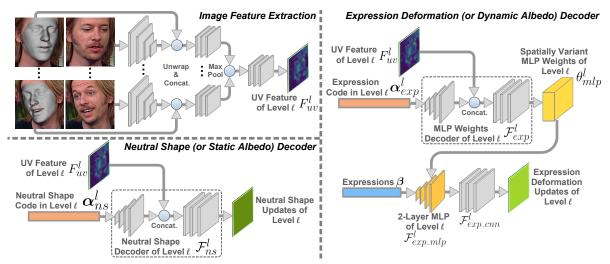
Figure 3. The illustration of different components in our neural rig decoder (Sec. 3.1). Top left: *Image Feature Extraction*. Bottom left: *Neutral Shape (or Static Albedo) Decoding*. Right: *Expression Deformation (or Dynamic Albedo) Decoding*.

maps in UV space from $\{\mathbf{I}_i\}$ and $\{\hat{\mathbf{V}}_i^{old}\}$, followed by max pooling and ResBlocks to get the desired UV feature F_{uv}.

Due to the 3-level scheme, we perform this feature extraction at the beginning of each level l with images $\{\mathbf{I}_i\}$ and reconstructed meshes of previous level $\{\hat{\mathbf{V}}_i^{l-1}\}$, resulting 3 UV feature maps of increasing resolutions $\{F_{uv}^l\}_{l=1}^3$.

3.1.2 Neutral Shape

As shown in the bottom left of Fig. 3, given the UV feature map F_{uv} and the neutral shape code $\boldsymbol{\alpha}_{ns}$ that is a portion of the rig code $\boldsymbol{\alpha}$, we aim to compute a neutral shape $\mathbf{V}_{ns}$ which is a 3D face mesh in neutral expression. A CNN structure $\mathcal{F}_{ns}(.)$ is used to decode $\boldsymbol{\alpha}_{ns}$ and F_{uv} into $\mathbf{V}_{ns}$ (or updates of $\mathbf{V}_{ns}$ in the 3-level scheme), which consists of several ResBlocks (w/o BatchNorm) and upsampling.

For the 3-level scheme, we repeat the mentioned decoding process for 3 times to increase resolutions and sum up the results. At the 1st level, we also include a PCA model from *Basel Face Model* (BFM) [28] to better leverage the statistical prior. Thus, we formulate the neutral shape as:

$$\mathbf{V}_{ns} = \overline{\mathbf{V}} + B_{bfm}\boldsymbol{\alpha}_{ns}^1 + \sum_{l=1}^3 \mathcal{F}_{ns}^l(\boldsymbol{\alpha}_{ns}^l; F_{uv}^l), \quad (2)$$

where $\overline{\mathbf{V}}$ and B_{bfm} are the mean shape and PCA bases from BFM [28]. Note that the 3 levels of decoding processes are gradually added into the formulation level-by-level during the multi-level optimization (Sec. 3.2.3).

3.1.3 Expression Deformation

As shown in the right of Fig. 3, given the UV feature map F_{uv}, the expression code $\boldsymbol{\alpha}_{exp}$ that is a portion of the rig code $\boldsymbol{\alpha}$, and the expression parameter $\boldsymbol{\beta}$, we aim to compute an expression deformation $\mathbf{D}_{exp}$ which are the per-vertex displacements added on the neutral shape. Three sub-networks are used to get $\mathbf{D}_{exp}$ including $\mathcal{F}_{exp}$, $\mathcal{F}_{exp\_mlp}$,

and $\mathcal{F}_{exp\_cnn}$. The networks $\mathcal{F}_{exp\_mlp}$ and $\mathcal{F}_{exp\_cnn}$ define a mapping from the expression parameter $\boldsymbol{\beta}$ to the final expression deformation $\mathbf{D}_{exp}$ as

$$\mathbf{D}_{exp} = \mathcal{F}_{exp\_cnn}(\mathcal{F}_{exp\_mlp}(\boldsymbol{\beta}; \theta_{mlp})), \quad (3)$$

where $\mathcal{F}_{exp\_mlp}$ is a 2-layer MLP with spatially variant weights θ_{mlp} and $\mathcal{F}_{exp\_cnn}$ is a CNN. Then, $\mathcal{F}_{exp}$ controls (or personalizes) this mapping by modifying the network weights θ_{mlp} according to the expression code $\boldsymbol{\alpha}_{exp}$ and the UV feature map F_{uv} as

$$\theta_{mlp} = \mathcal{F}_{exp}(\boldsymbol{\alpha}_{exp}; F_{uv}). \quad (4)$$

For the 3-level scheme, we repeat the mentioned decoding process for 3 times in increasing resolutions and sum up the results. Note that to utilize the statistical prior, we adopt the expression PCA bases [3, 13], built from Facewarehouse [9], in the 1st level, thus resulting in some modifications on the network architecture. More specifically, $\mathcal{F}_{exp\_mlp}^1$ is a 2-layer MLP with spatially invariant weights θ_{mlp}^1, and $\mathcal{F}_{exp\_cnn}^1$ is replaced by a matrix multiplication with the expression PCA bases B_{exp}. Formally, we have

$$\mathbf{D}_{exp} = B_{exp}\mathcal{F}_{exp\_mlp}^1(\boldsymbol{\beta}; \theta_{mlp}^1)$$
$$+ \sum_{l=2}^3 \mathcal{F}_{exp\_cnn}^l(\mathcal{F}_{exp\_mlp}^l(\boldsymbol{\beta}; \theta_{mlp}^l)), \quad (5)$$

where $\theta_{mlp}^l = \mathcal{F}_{exp}^l(\boldsymbol{\alpha}_{exp}^l; F_{uv}^l)$, $l = 1, 2, 3$. The final mesh can be obtained by $\mathbf{V} = \mathbf{V}_{ns} + \mathbf{D}_{exp}$. Similar to the neutral shape, the 3 levels of decoding processes are gradually added into the formulation level-by-level during the multi-level optimization (Sec. 3.2.3).

3.1.4 Albedo

Following [11], we also estimate dynamic albedo maps to better capture facial details such as wrinkles. Given the UV feature map F_{uv}, the albedo code $\boldsymbol{\alpha}_{alb}$, and the expression

parameter $\boldsymbol{\beta}$, we aim to compute the per-vertex albedo $\mathbf{A}$. Since only a small amount of high-frequency details could vary with expressions, we first estimate a static albedo at the 1st and 2nd levels similar to the neutral shape, then add the dynamic components at the 3rd level similar to the expression deformation. Formally, we have

$$\mathbf{A} = \overline{\mathbf{A}} + \sum_{l=1}^{2} \mathcal{F}_{alb}^{l}(\boldsymbol{\alpha}_{alb}^{l}; F_{uv}^{l})$$
$$+ \mathcal{F}_{alb\_cnn}^{3}(\mathcal{F}_{alb\_mlp}^{3}(\boldsymbol{\beta}; \theta_{mlp}^{3})), \quad (6)$$

where $\theta_{mlp}^{3} = \mathcal{F}_{alb}^{3}(\boldsymbol{\alpha}_{alb}^{3}; F_{uv}^{3})$ and $\overline{\mathbf{A}}$ is the average albedo map from BFM [28]. The 3 levels of decoding processes are also gradually added into the formulation level-by-level during the multi-level optimization (Sec. 3.2.3).

3.2. Learnable Optimization

Given the parameterization of the face rig as in Sec. 3.1, the next step is to optimize the rig code $\boldsymbol{\alpha}$ and per-image parameters $\{\mathbf{x}_i = (\boldsymbol{\beta}_i, \mathbf{p}_i, \boldsymbol{\gamma}_i)\}_{i=1}^{N}$ (*i.e.* expressions $\boldsymbol{\beta}_i$, poses $\mathbf{p}_i$, and illuminations $\boldsymbol{\gamma}_i$) to obtain the final riggable 3D reconstructions as shown in Fig. 2. The estimation is done by an energy minimization with end-to-end learnable components. We will first introduce how to to get the per-image reconstructions from the parameters $\boldsymbol{\alpha}$ and $\{\mathbf{x}_i\}$ (Sec. 3.2.1), then describe the energy formulation used to optimize the parameters (Sec. 3.2.2), and finally, solve the optimization in a multi-level fashion (Sec. 3.2.3).

3.2.1 Per-image Reconstruction from parameters

Given the rig code $\boldsymbol{\alpha}$ and per-image parameters $\{\mathbf{x}_i\}_{i=1}^{N}$, we aim to obtain the per-image reconstructions (*i.e.* one colored 3D mesh $(\hat{\mathbf{V}}_i, \hat{\mathbf{C}}_i)$ for each image), on which the objective energy is computed. For each image, we first decode the rig code $\boldsymbol{\alpha}$ and the expression parameter $\boldsymbol{\beta}_i$ to a mesh with albedo $(\mathbf{V}_i, \mathbf{A}_i)$ by the neural decoder as in Equ. (1). Then, this mesh is transformed and projected to the image plane by the weak perspective camera model with pose $\mathbf{p}_i = (s, \mathbf{R}, \mathbf{t})$ (*i.e.* scale s, rotation $\mathbf{R} \in SO(3)$, and 2D translation $\mathbf{t} \in \mathbb{R}^2$) as $\hat{\mathbf{V}}_i = s\mathbf{R}\mathbf{V}_i + \mathbf{t}$ and $\mathbf{\Pi}(\hat{\mathbf{V}}_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \hat{\mathbf{V}}_i$, where $\mathbf{\Pi}(\cdot)$ is the projection function. Following [36, 38], we assume Lambertian surface and adopt the Spherical Harmonics (SH) illumination model [29] as $\hat{\mathbf{c}}_i = \mathbf{a}_i \cdot \sum_{b=1}^{9} \boldsymbol{\gamma}_{i,b} H_b$ to obtain the final mesh color $\hat{\mathbf{C}}_i$, where $\hat{\mathbf{c}}_i / \mathbf{a}_i$ is the per-vertex color/albedo.

3.2.2 Energy Formulation

We define the objective to measure how well the reconstructions $\{(\hat{\mathbf{V}}_i, \hat{\mathbf{C}}_i)\}_{i=1}^{N}$ explain the input images $\{\mathbf{I}_i\}_{i=1}^{N}$:

$$\mathbf{E}(\boldsymbol{\alpha}^l, \{\mathbf{x}_i\}) = \lambda_{app}\mathbf{E}_{app} + \lambda_{lan}\mathbf{E}_{lan} + \lambda_{pho}\mathbf{E}_{pho}, \quad (7)$$

with multi-view appearance consistency $\mathbf{E}_{app}$, landmark alignment $\mathbf{E}_{lan}$, and photo-metric reconstruction $\mathbf{E}_{pho}$.

For the multi-view appearance consistency, we follow the formulation of Bai *et al.* [3] and define this term in a feature-metric manner. For each image $\mathbf{I}_i$, we project the reconstructed mesh $\hat{\mathbf{V}}_i$ onto the feature map of $\mathbf{I}_i$ extracted by a FPN [25] and do sampling via bilinear interpolation to get per-vertex feature vectors $F(\hat{\mathbf{v}}_i^k)$, where $\hat{\mathbf{v}}_i^k$ denotes the the k-th vertex of the mesh $\hat{\mathbf{V}}_i$. We then compute the L_2 differences of feature vectors between the corresponding vertices of pairs of images. Formally, we have

$$\mathbf{E}_{app} = \frac{2}{N(N-1)} \sum_{i \neq j} \frac{1}{M} \sum_{k=1}^{M} \|F(\hat{\mathbf{v}}_i^k) - F(\hat{\mathbf{v}}_j^k)\|_2^2, \quad (8)$$

where M is the number of vertices per mesh. We exclude invisible vertices according to the z-buffer in rendering.

For landmark alignment, we use a similar formulation as [38, 39] with sliding landmarks on the contour, reads as

$$\mathbf{E}_{lan} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{68} \sum_{k=1}^{68} \|\mathbf{u}_i^k - \mathbf{\Pi}(\hat{\mathbf{v}}_i^{\mathbf{u}^k})\|_2^2, \quad (9)$$

where $\hat{\mathbf{v}}_i^{\mathbf{u}^k}$ denotes the mesh vertex corresponding to the k-th landmark of image $\mathbf{I}_i$ and $\{\mathbf{u}_i^k\}$ are computed with the state-of-the-art face landmark detector [8].

For the photo-metric reconstruction, we first sample the per-vertex colors $\mathbf{c}_i^k$ from the image $\mathbf{I}_i$ in a similar way for computing the per-vertex feature $F(\hat{\mathbf{v}}_i^k)$. Then, we measure the L_2 consistency between the sampled image colors $\mathbf{c}_i^k$ and the reconstructed colors $\hat{\mathbf{c}}_i^k$ from $\hat{\mathbf{C}}_i$. Formally, we have

$$\mathbf{E}_{pho} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \sum_{k=1}^{M} \|\mathbf{c}_i^k - \hat{\mathbf{c}}_i^k\|_2^2, \quad (10)$$

where visibility is handled in the same way as for $\mathbf{E}_{app}$.

3.2.3 Solving Optimization

Initialization. At the beginning, an initialization is needed as the starting of optimization. We obtain the initial face rig by removing all levels of rig decoders in Equs. (2)(5)(6), resulting in a constant rig $(\overline{\mathbf{V}}, \overline{\mathbf{A}}) = Rig(\cdot)$. Thus, the decoded mesh with albedo is $(\mathbf{V}_i^0, \mathbf{A}_i^0) = (\overline{\mathbf{V}}, \overline{\mathbf{A}})$ for each image. The initial pose $\mathbf{p}_i^0$ is regressed by a pre-trained network as in [3], which is used to get the per-image mesh $\hat{\mathbf{V}}_i^0$ by transforming $\mathbf{V}_i^0$. Finally, the initial illumination is estimated by solving a linear weighted least square problem with a modified version of $\mathbf{E}_{pho}$ as the objective defined as

$$\mathbf{E}_{pho}' = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \sum_{k=1}^{M} w_i^k \|\mathbf{c}_i^k - \hat{\mathbf{c}}_i^k\|_2^2, \quad (11)$$

where w_i^k is a constant set as $(\mathbf{a}_i^k)^{-0.5}\|\mathbf{c}_i^k - \hat{\mathbf{c}}_i^k\|_2^{-1.5}$ and $\mathbf{a}_i^k$ is the per-vertex albedo to gain better robustness to outliers. For convenience, we treat this initialization as level 0.

Multi-level Scheme. For each level $l \geq 1$, we have 3 steps to update the face rig and per-image parameters. First, given the images $\{\mathbf{I}_i\}$ and the per-image meshes of the pre-
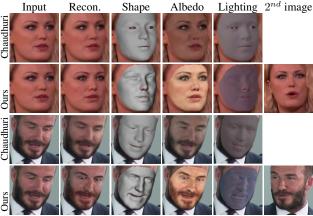
Figure 4. Qualitative comparison with Chaudhuri *et al.* [11]. Our estimated geometries are superior and textures are comparable.
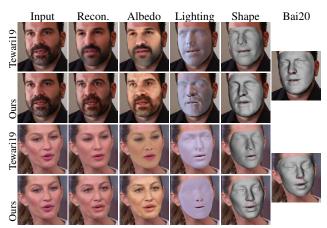

Figure 5. Qualitative comparison with Tewari *et al.* [38] and Bai *et al.* [3]. We obtain more faithful geometries that better reflect the corresponding personnel and higher resolution textures.

vious level $\{\hat{\mathbf{V}}_i^{l-1}\}$, we extract the level l UV feature F_{uv}^l (Sec. 3.1.1). Then the level l rig decoders in Equs. (2)(5)(6) are added into the rig formulation to parameterize the face rig updates of level l into $\boldsymbol{\alpha}^l = (\boldsymbol{\alpha}_{ns}^l, \boldsymbol{\alpha}_{exp}^l, \boldsymbol{\alpha}_{alb}^l)$. $\boldsymbol{\alpha}^l$ is set to zeros and $\{\mathbf{x}_i\}$ are set to the outputs of level $l-1$. Second, we solve the optimization $\arg\min_{\boldsymbol{\alpha}^l, \{\boldsymbol{\beta}_i, \mathbf{p}_i\}} \mathbf{E}$ via gradient descent with step sizes regressed by a network as in [3]. Finally, the illumination $\boldsymbol{\gamma}_i$ is updated according to Equ. (11) as the initialization.

3.3. Training Losses

Our model is trained with registered ground truth 3D scans in a supervised manner, with ground truth in 3D scans in the camera space, identity label, and 3D scans in neutral expression for each identity. The complete loss is

$$L = L_{pose} + L_{recon\_geo} + L_{ns\_geo}$$
$$+ \lambda_1 L_{recon\_pho} + \lambda_2 L_{\boldsymbol{\beta}} + \lambda_3 L_{ns\_con}. \quad (12)$$

L_{pose} and L_{recon_geo} are losses supervising the per-image meshes, each of which contains 2 terms as $L_{pose} = 0.025 L_{lan} + L_{dep\_v}$ and $L_{recon\_geo} = L_{den\_v} + 1000 L_{norm}$. We define these 4 terms following [3]. Please see the supplementary for detailed loss definitions.

L_{ns_geo} is a geometry loss supervising the neutral shape reconstruction. It has the same formulation as L_{recon_geo}, except being computed between the ground truth neutral shape and the estimated one.

L_{recon_pho} is a photo-metric loss supervising the per-image reconstructions. Following [11], we use the differentiable rendering to obtain the reconstructed image, then compute the image intensity loss L_{img} and image gradient loss L_{grad} in $L_{2,1}$ norm [11,40]. These two losses are added together as $L_{recon\_pho} = L_{img} + L_{grad}$.

$L_{\boldsymbol{\beta}}$ is an L_2 loss to encourage a small expression parameter $\boldsymbol{\beta}_i$ when the per-image reconstruction should be in neutral expression. Formally, we have $L_{\boldsymbol{\beta}} = \|\boldsymbol{\beta}_i\|_2^2$ when the ground truth $\mathbf{V}_i^{gt}$ is in neutral expression.

L_{ns_con}, termed as neutral shape consistency loss, is computed on the estimated neutral shapes to encourage small intra-identity differences. For two estimated neutral shapes of the same identity in a mini-batch, we compute the L_2 vertex loss $L_{ns\_con\_v}$ (same form as L_{dep_v}) and the cosine normal loss $L_{ns\_con\_n}$ (same form as L_{norm}) between them to enforce the consistency, which reads as $L_{ns\_con} = L_{ns\_con\_v} + 1000 L_{ns\_con\_n}$.

4. Experiments

Training Data. Our model is trained on Stirling/ESRC 3D face database [1], containing textured 3D scans of $100+$ subjects, each with up to 8 expressions. We synthesize training data by rendering 3D scans of 116 subjects (31 for validation). For each training/validation sample, we first randomly sample 10 expressive scans with replacement (*i.e.* can have repeated expressions) of the same identity, then each scan is associated with a random pose and an illumination sampled from the training data of Sengupta *et al.* [35]. Finally, we use the selected scans, poses, and illuminations to render 10 images as a training/validation sample. Our training losses require dense vertex correspondences between the reconstructions and scans (*i.e.* registered scans). Following [3], we first fit the 3DMM (BFM + expression bases) to the landmarks of each scan, then perform Non-Rigid ICP [2] to obtain the dense correspondences.

Implementation. Our algorithm is implemented via Pytorch. To optimize each level, we perform 3 iterations of parameter updates with weights $\lambda_{app} = 0.25$, $\lambda_{lan} = 0.025$, and $\lambda_{pho} = 1$. During training, we randomly select 2-7 out of 10 images for each data sample to make our model adapt to different numbers of images. When computing $\mathbf{E}_{app}$ during training, we randomly sample vertices to save GPU memory. To focus our model on reconstruction quality, we remove the neutral shape consistency loss L_{ns_con} for now (*i.e.* $\lambda_3 = 0$). The rest loss weights are $\lambda_1 = \lambda_2 = 10$, and

	Bosphorus		BU3DFE	
	Mean	STD	Mean	STD
Protocol of Bai et al. [3]				
Single Tewari18 [39]	-	-	1.78	0.49
View Deng19 [13]	1.47	0.40	1.38	0.37
Two Tewari19 [38]	-	-	1.74	0.45
Views Bai20 [3]	1.44	**0.38**	1.11	0.29
Ours(R)	1.37	0.44	1.09	0.31
Ours	**1.36**	**0.38**	**1.00**	**0.27**

Table 1. Geometry accuracy on Bosphorus [33] and BU3DFE [47] with testing protocol of Bai *et al.* [3].

	BU3DFE	
	Mean	STD
Protocol of Tewari et al. [38]		
Single Tewari18 [39]	1.83	0.39
View Shang20 [36]	1.55	0.31
Two Tewari19 [38]	1.78	0.45
Views Chaudhuri20 [11]	1.61	0.32
Ours(R)	1.27	0.26
Ours	**1.21**	**0.25**

Table 2. Geometry accuracy on BU3DFE [47] with protocol of Tewari *et al.* [38].

	NoW Dataset	
	Mean	STD
Single View		
Tuan17 [42]	2.31	0.42
Feng18 [15]	1.97	0.68
Sanyal19 [32]	1.52	0.50
Multi Views		
Ours	**1.33**	**0.28**

Table 3. Geometry accuracy of neutral shapes on NoW [32].

the batch size is 1 with a learning rate 2.0×10^{-5}.

4.1. Per-image Reconstruction

Our method can estimate a 3D face per input image. We evaluate 3D reconstruction quantitatively and qualitatively. **Quantitative Evaluation**. Two datasets BU3DFE [47] and Bosphorus [33] are used to evaluate the 3D reconstructions. Following [3], we first roughly align the predicted mesh to the 3D scan and crop the scan based on landmarks (8 for BU3DFE and 5 for Bosphorus), then perform ICP [48] to improve the alignment. Finally, per-vertex point-to-plane distances from 3D scans to reconstructions are computed. On BU3DFE, we also follow the protocol of Tewari *et al.* [38] to align and compute error based on dense correspondences for fairly comparison with Chaudhuri *et al.* [11], whose numbers are cited from their paper. As in Tab. 1 and Tab. 2, our method outperforms various single- and multi-image monocular methods, achieving the state of the art. **Qualitative Evaluation**. We also visually compare with previous works on VoxCeleb2 [12] dataset. Fig. 4 shows the comparison with Chaudhuri *et al.* [11]. In terms of geometry, our method outperforms [11] by capturing more medium level details such as smiling lines, better nose shapes, and geometry around eyes. In terms of texture, we obtain comparable results, with more high-frequency details but also slightly more artifacts. Note that our method additionally takes in the second image shown in Fig. 4. Compared to [3] and [38] in Fig. 5, our estimated shapes are more faithful against the input images and better reflects the corresponding personnel. Also, our estimated textures are in a higher resolution comparing to Tewari *et al.* [38], while Bai *et al.* [3] does not estimate textures. More video and image results can be found in the supplementary material.

4.2. Neutral Shape Reconstruction

One natural property of face rigs is the ability to disentangle neutral shapes and expression deformations. Therefore, we also evaluate the reconstructed neutral shape as a measure of the face rig quality, by using standard geometry accuracy as well as 3D face verification rate. **Geometry Accuracy**. We evaluate our neutral shapes on the NoW Dataset [32], which contains 2,054 2D images of 100 subjects and a separate neutral shape 3D scan for each

subject. Under each subject, the images are classified into 4 categories (*i.e.* neutral, expression, occlusion and selfie).

Since the dataset is originally designed for single-image methods, we use the following protocol to adapt it to our multi-image setting. For each subject, we run our algorithm separately for images of different categories, resulting in 4 neutral shapes. The motivation is to make our setting closer to the original single-image setting. When only 1 image is available in some cases, we horizontally flip it to generate the second input image. Finally, geometry errors between reconstructed neutral shapes and ground truth scans are computed as in [32] in a per-image manner (*i.e.* one neutral shape is tested as many times as the number of images used to reconstruct it) to be consistent with the single-image setting. As in Tab. 3, our approach outperforms prior monocular methods. Note that we only compare with single-image methods since we were not able to find multi-image methods that can separate neutral shapes and expressions while having a public implementation. **3D Face Verification**. Quantitative evaluations in Tab. 1, 2, and 3 are performed on datasets of a small number of subjects with images taken in controlled setting. However, it is important to evaluate how 3D face reconstruction works on in-the-wild images from a large number of subjects, a challenge without ground truth 3D scans. Thus, instead of computing geometry errors, following [26], we resort to 3D face verification rate to measure the neutral shape quality.

To this end, we test on the Labeled Faces in the Wild (LFW) [20] benchmark. We estimate the neutral shape of each image via the tested method, then train a network to perform 3D face verification on the estimated shapes. Our method inputs the original image and its flipped one. Please see supplementary for details of the verification setup.

To increase the robustness and discriminativeness, we finetune our model with $\lambda_3 = 5$ and augment images with synthetic degradation. Limitated by GPU memory, we only select 2 out of 10 images per sample and set batch size to 2. We denote this version of model as *Ours(R)*.

We compare *Ours(R)* to Shang20 [36], a self-supervised monocular 3D face reconstruction method. Note that [36] is trained on a large amount of in-the-wild faces, serving as a strong baseline on generalization and robustness. From the verification accuracy (Ours(R): 81.4%, [36]: 81.0%)
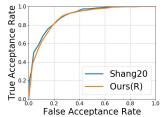
Figure 6. 3D face verification on LFW [20].

	Mean	STD
W/o $\boldsymbol{\alpha}_{exp}$ & $\boldsymbol{\alpha}_{ns}$	1.48	0.41
W/o $\boldsymbol{\alpha}_{ns}$	1.41	0.42
W/o $\boldsymbol{\alpha}_{exp}$	1.45	0.40
Full	**1.36**	**0.38**

Table 4. Comparing with regression baselines on Bosphorus [33].

Exp. Types	Viewed	Novel	
Methods	Ours	B1	B2
Mean	0.99	1.07	1.22	1.50
STD	0.31	0.35	0.52	0.56

Table 5. Geometry errors of novel and viewed expressions in self-retargeting. Key words: (B1) Use PCA bases of [36] and our estimated neutral shape. (B2) Replace the neutral shape of rig in B1 with averaged neutral shape obtained from [36].

and Fig. 6, two methods perform comparably, demonstrating that even on in-the-wild faces of diverse identities, our method still has reasonable generalization and robustness.

4.3. Retargeting

Retargeting is a standard application of riggable 3D face reconstruction, where the reconstructed rig of the target actor is re-animated by an image/video of a source actor. We quantitatively evaluate self-retargeting and qualitatively evaluate cross-actor video retargeting.

Self-Retargeting. Similar to Sec. 4, we evaluate on synthetic data rendered from 31 test subjects. We render 284 samples, each has 8 images with different expressions. We use 7 images to build the face rig and use the left one as a novel expression for self-retargeting. Specifically, we run our method on the left image and its flipped version to obtain the expression parameter (*exp param*) of this novel expression. Then we apply *exp param* to the estimated rig and compute geometry errors. We also include reconstruction errors of 7 viewed expressions as a reference. Since the code of [11] is not published, we design two baselines based on [36]: 1) B1: Use PCA bases of [36] to model the expression space, along with our estimated neutral shape, to form a rig. We obtain *exp param* by the regressor in [36]; 2) B2: Replace the neutral shape of rig in B1 with averaged neutral shape obtained from [36]. As in Tab. 5, our method has better accuracy than the baselines on novel expressions. Our error on novel expressions is also close to the viewed expressions, indicating good generalization of expressions. **Video Retargeting**. As in Fig. 1, our method outputs good results for different targets in both shapes and textures that faithfully reflect the target identity, and reasonably transfers the expressions. We also visually compare with Chaudhuri *et al*. [11] on their demo video. Selected frames are shown in Fig. 7. Our method has superior shape quality better re-
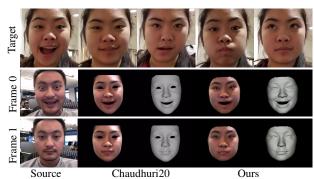

Figure 7. Video retargeting comparing to Chaudhuri *et al*. [11].

flecting the personal characteristics, such as the round chin instead of the sharp one from [11], and achieves reasonable expression transfer results. Note that in Fig. 7 we obtain the rig albedo by removing shadings from image colors of the target actor for better visual quality, and the target rig is built by *Ours(R)* for better robustness. However, our method has a few limitations on unusual expressions, eyelid motion, and the amplitude of transferred expressions. Video results and more analysis can be found in the supplementary.

4.4. Optimization vs Regression

The main novelty of our method is the optimizable neural parameterization of the face rig coupled with the learnable optimization. This design introduces optimization into network inference thus explicitly enforces constraints such as multi-view appearance consistency, landmark alignment, and photo-metric reconstruction, which are derived from the first principles based on the domain knowledge. This additional prior information has the potential to improve the 3D reconstruction quality. We investigate this advantage by comparing our method with regression baselines, where the components of the face rig are directly predicted by the neural network instead of being optimized. Please refer to the supplementary for more details of the regression baselines. As shown in Tab. 4, the performance drops when one or more rig components are regressed, demonstrating the effectiveness of explicit optimization during inference.

5. Conclusion

We solve riggable 3D face reconstruction from monocular RGB images by an end-to-end trainable network embedded with a in-network optimization. The network contains an optimisable neural face rig parameterization coupled with a learnable optimization. The optimization explicitly enforces first-principal constraints during inference, while the learning components leverage deep priors to constrain the ill-posedness and alleviate the optimization difficulty. Experiments demonstrate that our method achieves state-of-the-art reconstruction accuracy, reasonable robustness and generalization ability, and can be applied to the standard face rig application such as retargeting.

References

[1] *Stirling/ESRC 3D face database.* 6

[2] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–8, 2007. 6

[3] Ziqian Bai, Zhaopeng Cui, Jamal Ahmed Rahim, Xiaoming Liu, and Ping Tan. Deep facial non-rigid multi-view stereo. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5850–5860, 2020. 2, 3, 4, 5, 6, 7

[4] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, 2003. 2

[5] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *Proc. of ACM SIGGRAPH*, volume 99, pages 187–194, 1999. 1, 2

[6] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2560–2568, 2018. 3

[7] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Trans. Graph.*, 32(4):40, 2013. 1, 2

[8] 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 *Int. Conf. Comput. Vis.*, pages 1021–1030, 2017. 5

[9] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graph.*, 20(3):413–425, 2013. 1, 2, 4

[10] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.*, 35(4), 2016. 1, 2

[11] Bindita Chaudhuri, Noranart Vesdapunt, Linda Shapiro, and Baoyuan Wang. Personalized face modeling for improved face reconstruction and motion retargeting. In *Eur. Conf. Comput. Vis.*, pages 142–160, 2020. 2, 3, 4, 6, 7, 8

[12] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *Proc. Interspeech 2018*, pages 1086–1090, 2018. 7

[13] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 4, 7

[14] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Trans. Graph.*, 39(5):1–38, 2020. 1

[15] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Eur. Conf. Comput. Vis.*, pages 534–551, 2018. 7

[16] Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Trans. Graph.*, 35(3):28, 2016. 1, 2, 3

[17] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8377–8386, 2018. 2

[18] Pei-Lun Hsieh, Chongyang Ma, Jihun Yu, and Hao Li. Unconstrained realtime facial performance capture. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1675–1683, 2015. 2

[19] Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.*, 36(6):1–14, 2017. 1, 2

[20] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Technical Report 07-49, UMass, Amherst*, 2008. 7, 8

[21] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph.*, 34(4):1–14, 2015. 1, 2

[22] John P Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Frederic H Pighin, and Zhigang Deng. Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8):2, 2014. 2

[23] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *ACM Trans. Graph.*, 29(4):1–6, 2010. 1, 2

[24] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013. 1, 2

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2117–2125, 2017. 5

[26] Feng Liu, Qijun Zhao, Xiaoming Liu, and Dan Zeng. Joint face alignment and 3d face reconstruction with application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(3):664–678, 2018. 7

[27] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *ACM Trans. Graph.*, 37(4):1–13, 2018. 1, 2

[28] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009. 4, 5

[29] Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 117–128, 2001. 5

[30] Elad Richardson, Matan Sela, and Ron Kimmel. 3d face reconstruction by learning from synthetic data. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 460–469. IEEE, 2016. 2

[31] Joseph Roth, Yiying Tong, and Xiaoming Liu. Adaptive 3d face reconstruction from unconstrained photo collections. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4197–4206, 2016. 2

[32] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7763–7772, 2019. 2, 7

[33] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56. Springer, 2008. 7, 8

[34] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *Int. Conf. Comput. Vis.*, pages 1576–1585, 2017. 2

[35] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of faces in the wild'. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6296–6305, 2018. 6

[36] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3d face reconstruction by occlusion-aware multiview geometry consistency. In *Eur. Conf. Comput. Vis.*, 2020. 5, 7, 8

[37] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *Int. Conf. Learn. Represent.*, 2019. 3

[38] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Fml: face model learning from videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10812–10822, 2019. 2, 5, 6, 7

[39] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2549–2559, 2018. 2, 5, 7

[40] Luan Tran, Feng Liu, and Xiaoming Liu. Towards high-fidelity nonlinear 3d face morphable model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1126–1135, 2019. 2, 6

[41] Luan Tran and Xiaoming Liu. On learning 3d face morphable model from in-the-wild images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):157–171, 2019. 2

[42] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5163–5172, 2017. 2, 7

[43] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popovic. Face transfer with multilinear models. In *ACM SIGGRAPH 2006 Courses*, pages 24–es. 2006. 1, 2

[44] Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Trans. Graph.*, 35(4):1–12, 2016. 1, 2

[45] Chenglei Wu, Takaaki Shiratori, and Yaser Sheikh. Deep incremental learning for efficient high-fidelity face tracking. *ACM Trans. Graph.*, 37(6):1–12, 2018. 1, 2, 3

[46] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 601–610, 2020. 2

[47] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006. 7

[48] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 7

[49] 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 *Comput. Graph. Forum*, volume 37, pages 523–550, 2018. 1, 2