High-Fidelity 3D Digital Human Creation from RGB-D Selfies

XIANGKAI LIN*, YAJING CHEN*, LINCHAO BAO*†, HAOXIAN ZHANG, SHENG WANG, and XUEFEI ZHE, Tencent AI Lab
XINWEI JIANG, Tencent NExT Studios
JUE WANG, DONG YU, and ZHENGYOU ZHANG, Tencent AI Lab

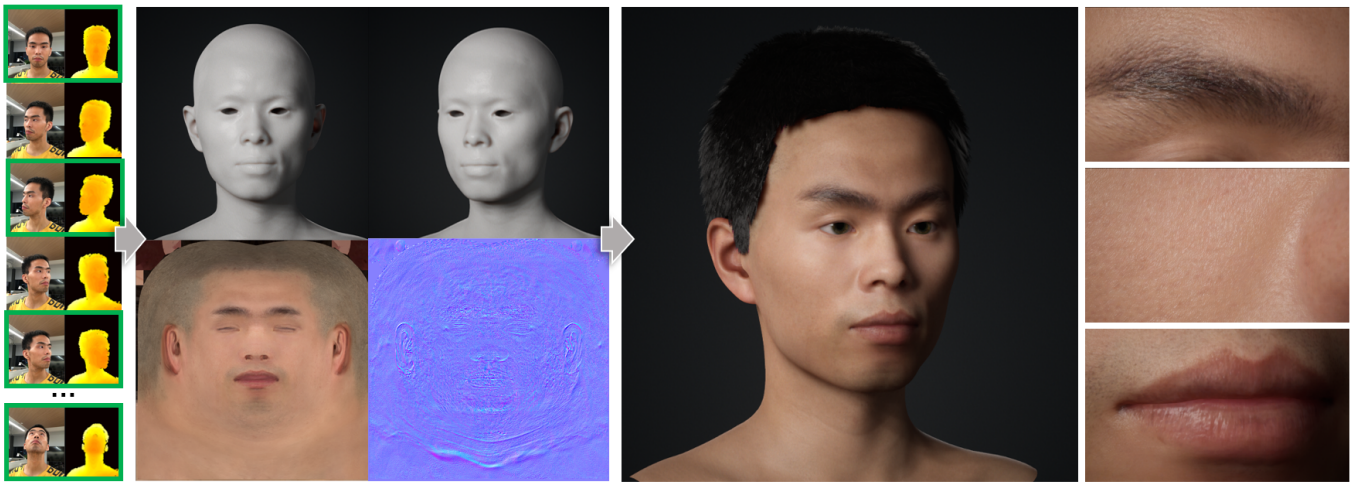


Fig. 1. Our system takes a user's RGB-D selfies as inputs and automatically produce a high-fidelity, riggable head model with high-resolution albedo map and normal map. The model faithfully preserves the user's facial identity features and can be rendered as a realistic digital human character.

We present a fully automatic system that can produce high-fidelity, photo-realistic 3D digital human characters with a consumer RGB-D selfie camera. The system only needs the user to take a short selfie RGB-D video while rotating his/her head, and can produce a high quality reconstruction in less than 30 seconds. Our main contribution is a new facial geometry modeling and reflectance synthesis procedure that significantly improves the state-of-the-art. Specifically, given the input video a two-stage frame selection algorithm is first employed to select a few high-quality frames for reconstruction. A novel, differentiable renderer based 3D Morphable Model (3DMM) fitting method is then applied to recover facial geometries from multiview RGB-D data, which takes advantages of extensive data generation and perturbation. Our 3DMM has much larger expressive capacities than conventional 3DMM, allowing us to recover more accurate facial geometry using merely linear bases. For reflectance synthesis, we present a hybrid approach that combines parametric fitting and CNNs to synthesize high-resolution albedo/normal maps with realistic hair/pore/wrinkle details. Results show that our system can produce faithful 3D characters with extremely realistic details. Code and the constructed 3DMM will be publicly available.

CCS Concepts: • **Computing methodologies** → **Reconstruction**; **Mesh models**.

Additional Key Words and Phrases: digital human, 3D face, avatar, 3DMM

*The first three authors contributed equally to the paper.
†Corresponding author: Linchao Bao (linchaobao@gmail.com).

Authors' addresses: Xiangkai Lin; Yajing Chen; Linchao Bao; Haoxian Zhang; Sheng Wang; Xuefei Zhe, Tencent AI Lab; Xinwei Jiang, Tencent NExT Studios; Jue Wang; Dong Yu; Zhengyou Zhang, Tencent AI Lab.

1 INTRODUCTION

Real-time rendering of realistic digital humans is an increasingly important task in various immersive applications like augmented and virtual reality (AR/VR). To render a realistic human face, high-quality geometry and reflectance data are essential. There exist specialized hardware like Light Stage [Alexander et al. 2009] for high-fidelity 3D faces capturing and reconstruction in the movie industry, but they are cumbersome to use for consumers. Research efforts have been dedicated to consumer-friendly solutions, trying to create 3D faces with consumer cameras, e.g., RGB-D data [Thies et al. 2015; Zollhöfer et al. 2011], multiview images [Ichim et al. 2015], or even a single image [Hu et al. 2017; Lattas et al. 2020; Yamaguchi et al. 2018]. While good results have been shown, the reconstructed 3D faces still contain artifacts and are far from satisfactory.

Indeed, faithful 3D facial reconstruction is a challenging problem due to the extreme sensitivity that human perception has towards faces. First, the recovered facial geometry needs to preserve all important facial features like cheek silhouettes and mouth shapes. Single-image based approaches [Hu et al. 2017; Lattas et al. 2020; Yamaguchi et al. 2018] can hardly achieve this due to the lack of reliable geometric constraints. With multiview RGB/RGB-D inputs, existing approaches [Ichim et al. 2015; Thies et al. 2015; Zollhöfer et al. 2011] do not fully leverage most recent advances in deep learning and differentiable rendering [Gecer et al. 2019; Genova et al. 2018], leading to inaccurate recovery that does not fully resemble the user's facial shape. Second, the synthesized facial reflectance maps need to be high-resolution with fine details like eyebrow hair,

lip wrinkles, and pore details on facial skin. Several recent work [Lattas et al. 2020; Saito et al. 2017; Yamaguchi et al. 2018] have tried to address these issues, but their results still lack natural facial details that are critical for realistic rendering.

In this paper, we present new facial geometry modeling and reflectance synthesis approaches that can produce faithful geometry shapes and high-quality, realistic reflectance maps, from multiview RGB-D data. Our geometry modeling algorithm extends differentiable renderer based 3DMM fitting, such as GANFIT [Gecer et al. 2019], from single image to multiview RGB-D data. Different from GANFIT, we employ conventional PCA-based texture bases instead of GAN to reduce the texture space, so that more data constraints can be exerted on geometric shaping. Additionally, we present an effective frame selection scheme, as well as an initial model fitting procedure, which can avoid enforcing conflicting constraints and increase system robustness. Moreover, we propose an effective approach that takes advantages of extensive data generation and perturbation to construct the 3DMM, which has much larger expressive capacity compared with previous methods. We show that even with the linear bases of the new 3DMM, our method can consistently recover accurate, personalized facial geometry.

For facial reflectance modeling, we use high-resolution 2K (2048 × 2048) UV-maps consisting of an albedo map and a normal map. We propose a hybrid approach that consists of a regional parametric fitting and CNN-based refinement networks. The regional parametric fitting is based on a set of novel pyramid bases constructed by considering variations in multi-resolution albedo maps, as well as high-resolution normal maps. Faithful but over-smoothed high-resolution albedo/normal maps can be obtained in this step. GAN-based networks are then employed to refine the albedo/normal maps to yield the final high-quality results. Our experiments show that even with the 680 × 480 resolution inputs, our method can produce high-resolution albedo/normal maps, where eyebrow hair, lip wrinkles and facial skin pores are all clearly visible. The high-quality reflectance maps significantly improve the realism of the final renderings in real-time physically based rendering engines.

With the recovered facial geometry and reflectance, we further present a fully automatic pipeline to create a full head rig, by completing a head model, matching a hair model, estimating the position/scale of eyeballs/teeth models, generating the expression blendshapes, etc. We conduct extensive experiments and demonstrate potential applications of our system.

Our major contributions include:

- A fully automatic system for producing high-fidelity, realistic 3D digital human characters with consumer-level RGB-D selfie cameras. Compared with previous avatar approaches, our system can generate higher quality assets for physically based rendering of photo-realistic 3D characters. The total acquisition and production time for a character is less than 30 seconds. The core code and the constructed 3DMM will be made publicly available[1].
- A robust procedure consisting of frame selection, initial model fitting, and differentiable renderer based optimization to recover faithful facial geometries from multiview RGB-D data, which can tolerate data inconsistency introduced during user data acquisition.
- A novel morphable model construction approach that takes advantages of extensive data generation and perturbation. The constructed linear 3DMM by our approach has much larger expressive capacity than conventional 3DMM.
- A novel hybrid approach to synthesize high-resolution facial albedo/normal maps. Our method can produce high-quality results with fine-scale realistic facial details.

[1]See our project page at: https://tencent-ailab.github.io/hifi3dface_projpage/

2 RELATED WORK

Creating high-fidelity realistic digital human characters commonly relies on specialized hardware [Alexander et al. 2009; Beeler et al. 2010; Debevec et al. 2000] and tedious artist labors like model editing and rigging [von der Pahlen et al. 2014]. Several recent work seek to create realistic 3D avatars with consumer devices like a smartphone using domain specific reconstruction approaches (i.e., with face shape/appearance priors) [Ichim et al. 2015; Lattas et al. 2020; Yamaguchi et al. 2018]. We mainly focus on prior arts along this line and briefly summarize the most related work in this section. Please refer to the recent surveys [Egger et al. 2020; Zollhöfer et al. 2018] for more detailed reviews.

2.1 Face 3D Morphable Model

The 3D morphable model (3DMM) is introduced in [Blanz and Vetter 1999] to represent a 3D face model by a linear combination of shape and texture bases. These bases are extracted with PCA algorithm on topological aligned 3D face meshes. To recover a 3D face model from observations, the 3DMM parameters can be estimated instead. Since the 3DMM bases are linear combinations of source 3D models, the expressive capacity of a 3DMM is rather limited. Researchers tried to increase the capacity either by automatically generating large amounts of topological aligned face meshes [Booth et al. 2016] or turn the linear procedure into a nonlinear one [Lüthi et al. 2017; Tran and Liu 2018]. However, the generated face models with these 3DMM models are usually flawed and not suitable for realistic digital human rendering. Another line to increase the expressive capacity of 3DMM is to segment the face into regions and then employ spatially localized bases to model each region [Blanz and Vetter 1999; Neumann et al. 2013; Tena et al. 2011]. We present a novel data augmentation approach that can effectively increase the capacity of either global or localized 3DMM with the same amount of source 3D face meshes as existing approaches.

2.2 Facial Geometry Capture

Capturing from Single Image. Given a single face image, the 3D face model can be recovered by estimating the 3DMM parameters with analysis-by-synthesis optimization approaches [Blanz and Vetter 2003; Garrido et al. 2013, 2016; Gecer et al. 2019; Hu et al. 2017; Romdhani and Vetter 2005; Thies et al. 2016; Yamaguchi et al. 2018]. A widely adopted approach among them is described in the Face2Face work [Thies et al. 2016], where the optimization objective consists of photo consistency, facial landmark alignment, and statistical regularization. Although there is a recent surge of deep learning based approaches to use CNNs to regress 3DMM parameters [Genova

et al. 2018; Tewari et al. 2017; Tran et al. 2017; Zhu et al. 2016], the results are commonly not in high fidelity due to lack of reliable geometric constraints. Some work go beyond the 3DMM parametric estimation to use additional geometric representations to model facial details [Chen et al. 2020; Guo et al. 2019; Jackson et al. 2017; Kemelmacher-Shlizerman and Basri 2011; Richardson et al. 2017; Sela et al. 2017; Shi et al. 2014; Tewari et al. 2018; Tran et al. 2018], but the results are generally not satisfactory for realistic rendering.

Capturing from Multiview Images. Ichim et al. [2015] present a complete system to produce face rigs by taking hand-held videos with a smartphone. The system relies on a multiview stereo reconstruction of the captured head followed by non-rigid registrations, which is slow and error-prone, especially when motion occurs or no reliable feature points can be detected in face regions. Besides, the two separated steps are very brittle: the reconstruction step cannot utilize the strong human facial prior from 3DMM and hence its results are usually rather noisy, which further leads to erroneous registration results. Recent research on multiview face reconstruction with deep learning approaches [Dou and Kakadiaris 2018; Wu et al. 2019] do not explicitly model geometric constraints and thus are not accurate enough for high-fidelity rendering.

Capturing from RGB-D Data. Modeling facial geometries from RGB-D data commonly consists of several separated steps [Bouaziz et al. 2013; Li et al. 2013; Weise et al. 2011; Zollhöfer et al. 2011, 2014]. First, accumulated point clouds are obtained with rigid registration [Newcombe et al. 2011]. Then a non-rigid registration procedure is employed to obtain a deformed mesh from the target mesh model [Bouaziz et al. 2016; Chen et al. 2013]. Finally, in order to obtain a 3DMM parametric representation, a morphable model fitting is applied using the deformed mesh as geometric constraints [Bouaziz et al. 2016; Zollhöfer et al. 2011]. Although the approach is widely adopted as standard practices, it suffers from accumulated errors due to the long pipeline. Thies et al. [2015] propose to use an unified parametric fitting procedure to directly optimize camera poses together with 3DMM parameters, taking into account both RGB and depth constraints. Their method achieves high-quality results in facial expression tracking, but is not specially designed for recovering personalized geometric characteristics.

2.3 Facial Reflectance Capture

Saito et al. [2017] propose to synthesize high-resolution facial albedo maps using CNN style features based optimization like the style transfer algorithm [Gatys et al. 2016]. However, the approach requires iterative optimization and needs several minutes of computation. Yamaguchi et al. [2018] further propose to inference albedo maps, as well as specular maps and displacement maps, using texture completion CNNs and super-resolution CNNs. GANFIT [Gecer et al. 2019] employs the latent vector of a Generative Adversarial Network (GAN) as the parametric representation of texture maps and then use an differentiable renderer based optimization to estimate the texture parameters. The most recent work AvatarMe [Lattas et al. 2020] propose to infer separated diffuse albedo maps, diffuse normal maps, specular albedo maps, and specular normal maps using a series of CNNs. We present a novel hybrid approach that can achieve high-quality results while at the same time is more robust than the above pure CNN-based approaches.

2.4 Full Head Rig Creation

To complete a full head avatar model, accessories beyond face region need to be attached to the recovered face model, e.g., hair, eyeballs, teeth, etc. Ichim et al. [2015] describe a simple solution to transfer these accessories (except hair) from a template model and adapt the scales/positions to the reconstructed face model. Cao et al. [2016] use image-based billboards to deal with eyes and teeth, and coarse geometric proxy to deal with hair model. Nagano et al. [2018] employ a GAN-based network to synthesis mouth interiors. Hu et al. [2017] propose to perform hair digitization by parsing hair attributes from the input image and then retrieving a hair model for further refinement. There are also some approaches working on modeling hairs in strand level [Chai et al. 2015; Hu et al. 2015; Luo et al. 2013; Saito et al. 2018; Wei et al. 2005]. For expression blendshape generation, Ichim et al. [2015] present a dynamic modeling process to produce personalized blendshapes, while Hu et al. [2017] adopt a simplified solution to transfer generic FACS-based blendshapes to the target model. The expression blendshapes can also be generated with a bilinear 3DMM model like FaceWarehouse [Cao et al. 2014], where the face identity and expression parameters are in independent dimensions.

3 OVERVIEW

We first introduce the 3D face dataset used in our system. Then we describe the goal of our system, followed by the user data acquisition process and a summary of the main processing steps.

3D Face Dataset. We use a specialized camera array system [Beeler et al. 2010] to scan 200 East Asians, including 100 males and 100 females, aged from 20 to 50 years old (with their permissions to use their face data). The scanned face models are manually cleaned and aligned to a triangle mesh template with 20,481 vertices and 40,832 faces. Each face model is associated with a 2K-resolution (2048 × 2048) albedo map and a 2K normal map, where pore-level details are preserved. A linear PCA-based 3DMM [Blanz and Vetter 1999] can be constructed from the dataset, which consists of shape bases, albedo map bases, and normal map bases. Note that we propose a novel approach to construct an augmented version of the 3DMM shape bases in Sec. 5.3. Besides, a novel pyramid version of the 3DMM albedo/normal maps is presented in Sec. 6.1.

Goal. The goal of our system is to capture high-fidelity users facial geometry and reflectance with RGB-D selfie data, which is further used to create and render full-head, realistic digital humans. For geometry modeling, we use 3DMM parameters to represent a face, since it is more robust to degraded input data and with more controllable mesh quality than deformation-based representations. For reflectance modeling, we synthesize 2K-resolution albedo and normal maps regardless of the input RGB-D resolution.

User Data Acquisition. We use an iPhone X to capture user selfie RGB-D data. Note that it is common nowadays for a smartphone to be equipped with a front-facing depth sensor and any such phone can be used. While a user is taking selfie RGB-D video, our capturing

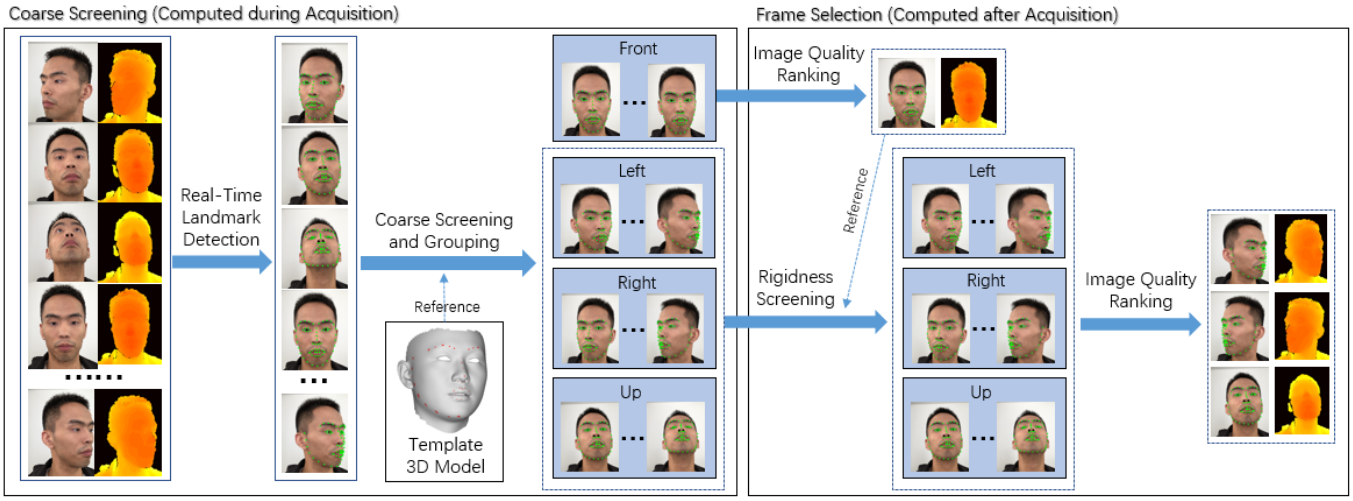


Fig. 2. Our two-stage frame selection procedure. Four frames are selected out of 200-300 frames considering both view coverage and data quality. Note that the reference model used for coarse screening and grouping is a template 3D face model, which may lead to inaccurate pose estimation. But the rough poses are sufficient for excluding extreme/invalid frames and categorizing the rest frames into pose groups. In the second stage, the reference model for rigidness screening is the lifted 3D landmarks from the front face data, which can result more accurate poses for more strict rigidness verification.

interface will guide the user to consecutively rotate his/her head to left, right, upward, and back to middle. The entire acquisition process takes less than 10 seconds, and a total of 200-300 frames of RGB-D images are collected, with resolution 640 × 480. The face region for computation is cropped (and resized) to 300 × 300. The camera intrinsic parameters are directly read from the device.

Processing Pipeline. We first employ an automatic frame selection algorithm to select a few high-quality frames that cover all sides of the user (Sec. 4). Then an initial 3DMM model fitting is computed with the detected facial landmarks in the selected frames (Sec. 5.1). Starting from the initial fitting, a differentiable renderer based optimization with multiview RGB-D constraints (Sec. 5.2) is applied to solve the 3DMM parameters as well as lighting parameters and poses. Based on the estimated parameters, high-resolution albedo/normal maps are then synthesized (Sec. 6). Finally, high-quality, realistic full head avatars can be created and rendered (Sec. 7).

4 FRAME SELECTION

There are typically 200-300 frames acquired from a user. For efficiency and robustness, we developed a robust frame selection procedure to select a few high-quality frames for further processing, which considers both view coverage and data quality. As shown in Fig. 2, the procedure consists of two stages as described below.

Coarse Screening and Preprocessing. We first apply a real-time facial landmark detector (a MobileNet [Howard et al. 2017] model trained on 300W-LP dataset [Zhu et al. 2016]) on RGB images to detect 2D landmarks for each frame. Then a rough head pose for each frame can be efficiently computed with the correspondences between the 2D landmarks and the 3D keypoints on a template 3D face model using PnP algorithm [Lepetit et al. 2009]. Frames with extreme/invalid poses or closed-eye/opened-mouth expressions can be easily identified and screened out with the 2D landmarks and rough head poses. We categorize the rest frames by poses into groups: *front*, *left*, *right*, and *up*. Each group only keeps 10-30 frames near the center pose of the group. Note that more groups can be obtained by categorizing the frames with finer-level angle partitioning. We experimented with different number of groups and found four is a good balance between accuracy and efficiency. The remaining depth images are preprocessed to remove depth values beyond the range between 40cm and 1m (the typical selfie distances). Bilateral filtering [Paris and Durand 2009] with a small spatial and range kernel is then applied to the depth images to attenuate noises.

Frame Selection. For each group, we further select one frame based on two criteria: image quality and rigidness. To measure the image quality of a frame, we compute the Laplacian of Gaussian (LoG) filter response and use the variance as a motion blur score (images with a larger score are sharper). A front face frame is first selected based on the motion blur score in the front group. We then compute the rigidness between each frame in the other groups and the front face with the help of depth data. Specifically, the detected 2D landmarks for each frame are lifted from 2D to 3D using depth data. Note that occluded landmarks are automatically removed according to the group that a frame belongs to, e.g., for a frame in the left group, the landmarks on the right side of the face are removed. We use RANSAC method to compute the relative pose between each frame in the other groups and the front face using the 3D-3D landmark correspondences [Arun et al. 1987]. Frames with too many outliers are considered as low rigidness and thus are excluded. Then a best

Fig. 3. The masks derived from the detected landmarks for texture blending. We use the verb "unwrap" to refer to the process of extracting partial texture maps from input photos and blending them into a complete texture map.

frame in each group can be found based on the motion blur score. The output of this step is four frames with the 3D landmarks.

5 FACIAL GEOMETRY MODELING

5.1 Initial Model Fitting

We use PCA-based linear 3DMM [Blanz and Vetter 1999] for parametric modeling. The shape and albedo texture of a face model is represented as

$$\mathbf{s} = \bar{\mathbf{s}} + S\mathbf{x}_{shp},$$
$$\mathbf{a} = \bar{\mathbf{a}} + A\mathbf{x}_{alb},$$

where $\bar{\mathbf{s}}$ is the vector format of the mean 3D face shape model, S is the shape identity basis, $\mathbf{x}_{shp}$ is the corresponding identity parameter vector to be estimated, $\bar{\mathbf{a}}$ is the vector format of the mean albedo map, A is the albedo map basis, $\mathbf{x}_{alb}$ is the corresponding albedo parameter vector to be estimated. The details of the bases are presented in Secs. 5.3 (shape) and 6.1 (albedo).

We fit an initial shape model with the detected 3D landmarks using a ridge regression [Zhu et al. 2015]. A partial texture map can be extracted by projecting the shape model onto each input image. With a predefined mask derived from landmarks for each view (see Fig. 3), the partial texture maps are then blended into a complete texture map using Laplacian pyramid blending [Burt and Adelson 1983]. The initial albedo parameters can be obtained with another ridge regression to fit the blended texture map.

5.2 Optimization

Fig. 4 shows our optimization framework. The parameters to be optimized are

$$\mathcal{P} = \{\mathbf{x}_{shp}, \mathbf{x}_{alb}, \mathbf{x}_{light}, \mathbf{x}_{pose}\},$$

and $\mathbf{x}_{shp} \in \mathbb{R}^{500}$ is the shape parameter, $\mathbf{x}_{alb} \in \mathbb{R}^{199}$ is the albedo parameter, $\mathbf{x}_{light} \in \mathbb{R}^{27}$ is the second-order spherical harmonics lighting parameter, $\mathbf{x}_{pose} \in \mathbb{R}^{6}$ include the rotation and translation parameters for rigid transformation. Note that we have only one $\mathbf{x}_{shp}$ and one $\mathbf{x}_{alb}$ for an user, while the number of $\mathbf{x}_{light}$ and $\mathbf{x}_{pose}$ equals to the number of views. With a set of estimated parameters and the 3DMM bases, a set of rendered RGB-D frames can be computed via a differentiable renderer [Gecer et al. 2019; Genova et al. 2018]. The distances between the rendered RGB-D frames and the input RGB-D frames can be minimized by backpropagating the errors to update parameters $\mathcal{P}$. The loss function to be minimized is defined as:

$$L(\mathcal{P}) = \omega_{rgb}L_{rgb}(\mathcal{P}) + \omega_{dep}L_{dep}(\mathcal{P}) + \omega_{id}L_{id}(\mathcal{P}) + \omega_{lan}L_{lan}(\mathcal{P}) + \omega_{reg}L_{reg}(\mathcal{P}), \quad (1)$$

where $L_{rgb}(\mathcal{P})$ denotes pixel-wise RGB photometric loss, $L_{dep}(\mathcal{P})$ indicates pixel-wise depth loss, $L_{id}(\mathcal{P})$ is identity perceptual loss, $L_{lan}(\mathcal{P})$ represents landmark loss, and $L_{reg}(\mathcal{P})$ means regularization terms. Note that the landmark loss, RGB photometric loss, and regularization term are similar to conventional analysis-by-synthesis optimization approaches [Thies et al. 2016]. The identity perceptual loss is also employed in recent differentiable renderer based approaches [Gecer et al. 2019; Genova et al. 2018]. We extend these losses into multiview setting and incorporate depth data for geometric constraints. The details of each term are as follows.

RGB Photo Loss. The pixelwise RGB photometric loss is:

$$L_{rgb}(\mathcal{P}) = \|I_{rgb} - I_{render}(\mathcal{P})\|_2,$$

where I_{rgb} is the input RGB image, I_{render} is the rendered RGB image from the differentiable renderer. We adopt $\ell_{2,1}$-norm because it is more robust against outliers than ℓ_2-norm.

Depth Loss. The depth loss is defined as:

$$L_{dep}(\mathcal{P}) = \rho(\|I_{dep} - I_z(\mathcal{P}))\|_2^2),$$

where $\rho(\cdot)$ defines a truncated ℓ_2-norm that clips the per-pixel mean squared error, I_{dep} is the input depth image, I_z is the rendered depth image from the differentiable renderer. The truncated function makes the optimization more robust to depth outliers.

Identity Perceptual Loss. To capture high-level identity information, we apply identity perceptual loss defined as

$$L_{id}(\mathcal{P}) = \|\psi(I_{rgb}) - \psi(I_{render})\|_2^2,$$

where $\psi(\cdot)$ is the deep identity features exacted from a pretrained face recognition model. Here we use features from the *fc7* layer of VGGFace model [Parkhi et al. 2015].

Landmark Loss. We define the landmark loss as the average distances between the detected 2D landmarks and projected landmarks from the predicted 3D model:

$$L_{lan}(\mathcal{P}) = \frac{1}{|\mathcal{F}|}\sum_{f_j \in \mathcal{F}} \omega_j \|f_j - \Pi(\Phi(v_j))\|_2^2,$$

where $f_j \in \mathcal{F}$ are the detected landmarks, $\Pi(\Phi(v_j))$ denotes that the vertex v_j is rigidly transformed by Φ and projected by camera Π. The weighting ω_j is to control the importance of each keypoint, where we set 50 for those located in eye, nose and mouth, while others are 1.

Regularization. To ensure the plausibility of the reconstructed faces, we apply regularization to shape and texture parameters:

$$L_{reg}(P) = \omega_{shp}\|x_{shp}\|_2^2 + \omega_{alb}\|x_{alb}\|_2^2,$$

where we set $\omega_{shp} = 0.4$ and $\omega_{alb} = 0.001$.

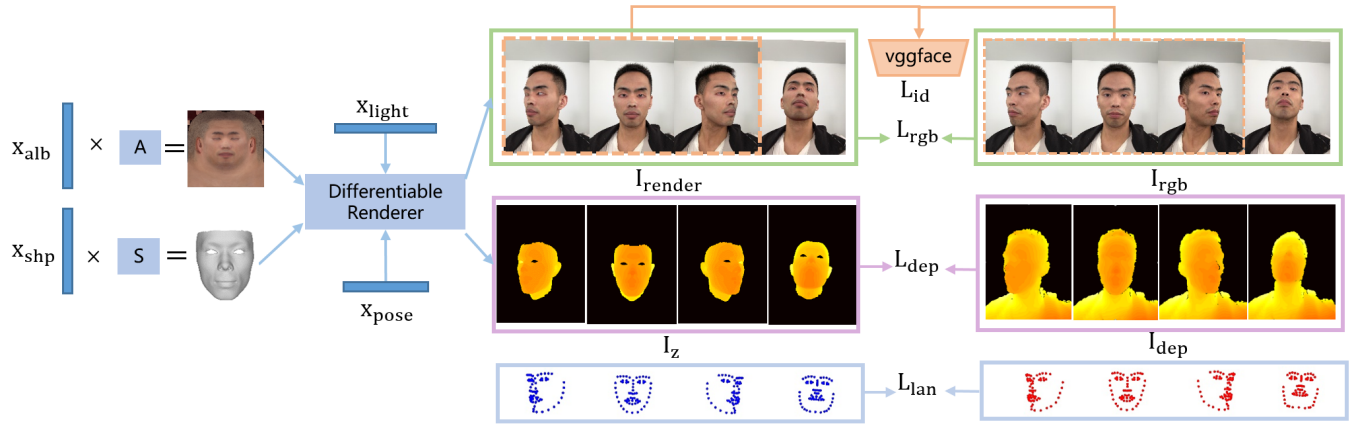


Fig. 4. Our optimization framework. The parameters to be solved include: 3DMM parameters $\mathbf{x}_{shp}$ and $\mathbf{x}_{alb}$ for a user, lighting parameters $\mathbf{x}_{light}$ and poses $\mathbf{x}_{pose}$ for each view. The constraints include: landmark loss L_{lan}, RGB photo loss L_{rgb}, depth loss L_{dep}, and identity perceptual loss L_{id}.

Implementation Details. For efficiency, we use albedo maps of a 512×512 resolution during the optimization. We render RGB-D images and compute the pixel losses in the same resolution as input depth images, which is 300×300. The weightings in Eq. (1) is set to $\omega_{rgb} = 1000.0$, $\omega_{depth} = 1000.0$, $\omega_{id} = 1.8$, $\omega_{lan} = 10$, $\omega_{reg} = 1.0$. We use Adam optimizer [Kingma and Ba 2014] in Tensorflow to update parameters for 150 iterations to get the results, with a learning rate 0.05 decaying exponentially in every 10 iterations.

Relation to Existing Approaches. The differences between our approach and state-of-the-art 3DMM fitting approaches are listed in Table 1. Result comparisons are presented in Sec. 8.2. Note that our implementation will be publicly available and can be easily configured into equivalent settings to other approaches by changing the combinations of input data and loss terms.

Method	Input	Loss Term	Optimizer
Ours	RGB-D	$L_{rgb}, L_{dep}, L_{id}, L_{lan}, L_{reg}$	DR-based
GANFIT	RGB	$L_{rgb}, L_{id}, L_{lan}, L_{reg}$	DR-based
Face2Face	RGB	$L_{rgb}, L_{lan}, L_{reg}$	Gauss-Newton
[Thies et al. 2015]	RGB-D	$L_{rgb}, L_{dep}, L_{lan}, L_{reg}$	Gauss-Newton

Table 1. Different 3DMM fitting approaches. "DR-based" stands for differentiable renderer based optimizer.

5.3 Morphable Model Augmentation

As the constraints incorporated in the optimization are rich, we found the expressive capacity of the linear 3DMM constructed using conventional approaches are very limited. We here present an augmentation approach to effectively boost the 3DMM capacity. Our approach is motivated by the observation that human faces are mostly not symmetrical. This will cause ambiguities when aligning face models. The reason is that during the alignment of two models, the relative rotation and translation between them is determined by minimizing the errors at some reference points on the models. Different reference points may lead to different alignment results. There are no perfect reference points due to the asymmetrical structures of human faces. This reminds us that we can perturb the relative pose between two aligned models to get an "alternative" alignment. In this way, we can actually get additional samples for PCA, since the new alignments introduces new morphing targets. Furthermore, we can use a set of perturbation operations including pose perturbation, mirroring, region replacement, etc., to augment the aligned models. Based on the large amount of generated data, we propose a stochastic iterative algorithm to construct a 3DMM that compresses more capacities into lower dimensions of the bases.

Fig. 5. Masks for region replacement.

Data Generation and Perturbation. Starting from the 200 aligned face shape models, our data generation and perturbation process consists of the following steps:

- *Region Replacement with Perturbation.* We first replace the nose region of each model with other models, with a rotation perturbation along the pitch angle (uniformly sampled within ±1 degree). Mouth region is also processed in the same way. For eye region, we apply replacement without perturbation. The different perturbations are empirically designed by minimizing the introduced visual defects during processing. The facial regions used in this step are shown in Fig. 5.
- *Rigid Transformation Perturbation.* We then apply rigid transformation perturbations to each face model, where the uniformly sampled range is set to: ±1 degree along yaw/pitch/roll angles for rotation, ±1% along each of the three axes for translation, ±1% for scale.
- *Mirroring.* Finally, we apply a mirroring for all the generated face models along model local coordinate system. In this way, we get over 100,000 face models in total.

Stochastic Iterative 3DMM Construction. Our iterative 3DMM construction algorithm is presented in Alg. 1. There are two levels of

ALGORITHM 1: Iterative 3DMM Construction Algorithm

Params : $n = 1000, m = 25, Thresh$
begin
  Face model set $\mathcal{S} \leftarrow$ initial 200 models;
  repeat
    Randomly sample (without replacement) a test set $\mathcal{D}$ with n face models from the whole dataset (over 100,000 models);
    $k \leftarrow 0$;
    $\xi \leftarrow \infty$;
    while $\xi > Thresh$ **do**
      Apply Principal Component Analysis (PCA) on $\mathcal{S}$;
      Select the principal components with 99.9% cumulative explained variance to get the 3DMM bases S^k;
      Fit the models in $\mathcal{D}$ using bases S^k;
      Select the m models with largest fitting errors as set $\mathcal{M}$;
      Add the the m corresponding mirrored models into $\mathcal{M}$;
      $\xi \leftarrow$ the median error of the m models;
      $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{M}$;
      $k \leftarrow k + 1$;
    end
  until *the whole dataset is sampled*;
end
Output : PCA bases S^k.

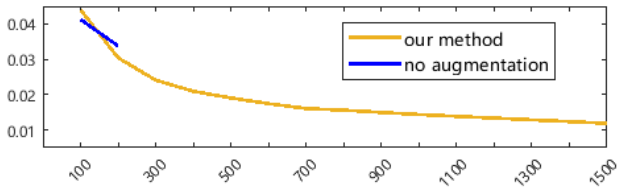


Fig. 6. Fitting errors (in cm) with different versions of bases. Note that the bases without augmentation are constructed from 200 models and thus have a maximum number of dimensions 199. The results show the expressive power of our bases is much larger than original bases.

loops in our algorithm. We maintain a model set $\mathcal{S}$ for 3DMM construction and update it inside the loops. In each iteration of the outer loop, we sample a test set $\mathcal{D}$ with $n = 1000$ models from the whole generated dataset. In each iteration of the inner loop, we use the constructed 3DMM from $\mathcal{S}$ to fit models in $\mathcal{D}$, and add $m = 25$ models with largest fitting errors in $\mathcal{D}$ into $\mathcal{S}$. The convergence threshold is empirically set such that the inner loop is usually converged in less than 5 iterations. Note that in the inner loop, a model sample in $\mathcal{D}$ could be repeatedly added into $\mathcal{S}$ for several times. In this case, constructing a 3DMM from the final $\mathcal{S}$ is different from directly performing PCA on the whole dataset as the data population is changed. Our algorithm encourages more data variance to be captured using fewer principal components (note that in each iteration we construct 3DMM using only the principal components with 99.9% cumulative explained variance).

Evaluation. In order to validate the effectiveness of our technique, we design a numerical evaluation experiments with the help of BFM 2009 model [Paysan et al. 2009]. Note that the source face models in our dataset are all East Asians, while those in BFM are mostly not Asians. The domain gap between the two datasets provides us

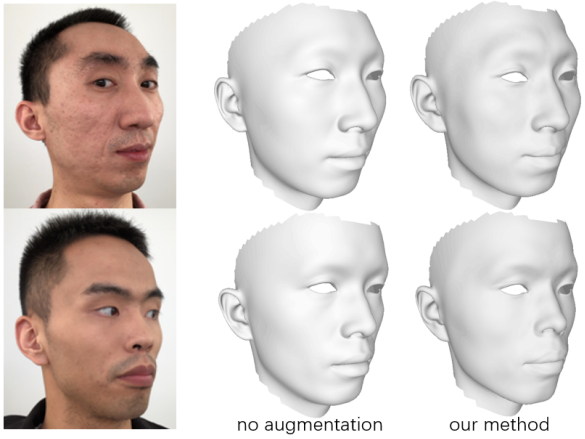


Fig. 7. The recovered geometries with two versions of bases. The bases obtained with our method can preserve more personalized facial geometries (note the regions of facial silhouette, mouth shape, and the nose shape).

a good benchmark for cross validation (our goal is not to model cross-ethnicity fitting, but use the relative fitting errors between different versions of 3DMM to evaluate their expressive power). For each BFM basis, we compute two 3D face models using the positive and negative standard deviation values. A total of 398 BFM face models are obtained in this way. We register the BFM face models to our mesh topology using Wrap3 software [R3ds 2020]. We use the extracted PCA bases from our dataset to fit the obtained BFM face models and measure the fitting errors. Fig. 6 shows the comparison of fitting errors between the augmented bases and the original bases. If only 100 bases are used, the augmented version has no advantage against the original version. As the number of bases grows, the augmented version clearly outperforms the original version. Note that the maximum number of bases of the original version is 200 since there are only 200 source models for 3DMM construction. For the augmented version, thousands of bases could be obtained since there are over 100,000 models after augmentation. Since our iterative algorithm emphasizes the expressive power of the principal components with 99.9% cumulative explained variance, most of the expressive capacities are compressed into these components. In our experiments, we found the final number of the components with 99.9% cumulative explained variance in different runs is generally around 500. Thus we use 500 bases through this paper. Fig. 7 shows a comparison of the facial geometries obtained using our optimization algorithm in Sec. 5.2 with different versions of PCA bases.

Relation to Localized 3DMM. There are some approaches constructing separate 3DMM for each facial region [Blanz and Vetter 1999; Neumann et al. 2013; Tena et al. 2011]. The localized 3DMMs obtain more capacities compared with global models by separating the deformation correlations between different facial regions. The region replacement augmentation in our approach is in the same spirit as localized 3DMM by explicitly generating samples with possible combinations of facial regions from different subjects. Compared with localized models, our 3DMM avoids online fusion of facial regions and thus is more efficient. Besides, our perturbation scheme and iterative 3DMM construction algorithm can be applied to localized

models to improve their capacities as well. In this paper, we employ global model for efficiency consideration.

6 FACIAL REFLECTANCE SYNTHESIS

In this section, we present a our hybrid approach to synthesis high-resolution albedo and normal maps. We notice that super-resolution based approaches [Lattas et al. 2020; Yamaguchi et al. 2018] cannot yield high-quality, hair-level details of the eyebrows. On the other hand, directly synthesizing high-resolution texture maps [Saito et al. 2017] may lead to overwhelming details, which also makes the rendering not realistic. Our approach addresses the problems with the help of a pyramid-based parametric representation. Fig. 8 shows the pipeline of our approach, which we explain as follows.

6.1 Regional Pyramid Bases

Fig. 9 illustrates the process to construct our regional pyramid bases. We first compute image pyramids consisting of two resolutions (512×512 and 2048×2048) for the 200 albedo maps in our dataset. We divide facial regions into 8 sub-regions indicated as the different colors in the left UV-map. The region partitioning is based on the fact that different regions have different types of skin/hair details. Denote the set of all regions as $\mathcal{K}$. For each region $k \in \mathcal{K}$, we construct a linear PCA-based blending model. We define each sample in our dataset as a triplet $(\mathbf{a}^k_{512}, \mathbf{a}^k_{2048}, \mathbf{g}^k_{2048})$, where $\mathbf{a}^k_{512}$ stands for 512×512 albedo map of region k, and $\mathbf{a}^k_{2048}$ and $\mathbf{g}^k_{2048}$ are the albedo map and normal map in 2048×2048 resolution. Then the triplet is vectorized into a vector format by fetching and concatenating all pixel values together from the three maps in the region. Note that during the process, the pixel indices in the three maps are recorded such that the vectorized sample can be "scatter back[2]" into UV-map format. For each region k, we apply a PCA on the the 200 vectorized samples to get the bases. Finally, the vectorized bases can be scattered back into UV-map format to obtain the blending bases $\{A^k_{512}, A^k_{2048}, G^k_{2048}\}_{k\in\mathcal{K}}$, where $A^k_{512} \in \mathbb{R}^{n_k \times 199}$ is the low-resolution albedo basis, $A^k_{2048} \in \mathbb{R}^{16n_k \times 199}$ is the high-resolution albedo basis, $G^k_{2048} \in \mathbb{R}^{16n_k \times 199}$ is the high-resolution normal basis. and n_k is the number of pixels within region k in the 512-resolution.

The constructed regional pyramid bases have several advantages compared with conventional bases. First, the expressive capacity is larger than global linear bases, while each region can be processed individually to accelerate the runtime. Second, the bases capture variations in both albedo and normal map. Third, the incorporated multiple resolutions can emphasize more structural information in the extracted bases. With the pyramid basis, we can perform parametric fitting on the low resolution, and directly apply the same parameters on high-resolution bases to obtain high-resolution albedo and normal maps. This not only reduces computation, but also generally yields higher-quality results than directly fitting on high resolution. Note that the albedo bases employed in our geometric fitting procedure (Sec. 5.2) is the conventional global model in 512-resolution, while the bases used in this section dedicated for reflectance synthesis are the regional pyramid model. The reason is that we found the less powerful albedo bases would make the optimization constraints more imposed on the geometric parameter estimation rather than texture parameter estimation. Otherwise more powerful albedo bases tends to result in overfitted textures but underfitted geometries.

[2]We use the "scatter_nd" function in Tensorflow as the "scatter back" operation.

6.2 Regional Fitting

Since the albedo parameters $\mathbf{x}_{alb}$ obtained in Sec. 5.2 are based on conventional global bases, the resulting albedo maps are not satisfactory due to limited expressive power. Here we directly extract textures from the source images using the estimated shape and poses from Sec. 5.2. Then a model-based delighting using the estimated lighting parameters from Sec. 5.2 is applied on the extracted textures, followed by an unwrapping and blending to yield an initial 512×512 albedo map I_{init}. We use the 512-resolution regional bases $\{A^k_{512}\}_{k\in\mathcal{K}}$ to fit the initial albedo map:

$$L(\mathbf{x}_{alb}) = \|I_{fit}(\mathbf{x}_{alb}) - I_{init}\|_2 + \omega_{tv} f_{tv}(I_{fit}(\mathbf{x}_{alb})) + \omega_{alb}\|\mathbf{x}_{alb}\|_2^2,$$

where $I_{fit}(\mathbf{x}_{alb}) = \sum_{k\in\mathcal{K}} A^k_{512}\mathbf{x}^k_{alb}$, f_{tv} denotes the total variation function, $\omega_{tv} = 0.0001$ and $\omega_{alb} = 0.001$. Note that the total variation term is essential to eliminate the artifacts in the resulting albedo maps near boundaries between regions. After obtaining $\mathbf{x}_{alb}$, we can directly compute a high-resolution albedo map $\hat{\mathbf{a}}_{2048}$ and a normal map $\hat{\mathbf{g}}_{2048}$ as

$$\hat{\mathbf{a}}_{2048} = \sum_{k\in\mathcal{K}} A^k_{2048}\mathbf{x}^k_{alb}$$
$$\hat{\mathbf{g}}_{2048} = \sum_{k\in\mathcal{K}} G^k_{2048}\mathbf{x}^k_{alb}.$$

With the help of regional pyramid bases, different types of skin/hair details in different regions can be separately preserved via the high-resolution bases, while the fitting process on low resolution makes the algorithm focus on major facial structures, e.g., the shape of the eyebrows and lips. Since the parametric representation is based on linear blending model, the results are usually over-smoothed (see Fig. 17). We next present our detail synthesis step to refine the albedo and normal maps.

6.3 Detail Synthesis

We adopt two refinement networks to synthesize details for albedo and normal map respectively. The refinement networks employ the architecture of a GAN-based image translation model, *pix2pix* [Isola et al. 2017]. As shown in Fig. 8, for albedo refinement, the network takes the fitted 2048-resolution albedo map as input and outputs a refined albedo map in the same resolution. For normal refinement, the refined albedo map and the fitted normal map are concatenated along channel dimension. The refinement network takes the concatenation as inputs and outputs a refined normal map.

During training, we first use facial region replacement and skin color transfer [Reinhard et al. 2001] to augment the 200 high-quality

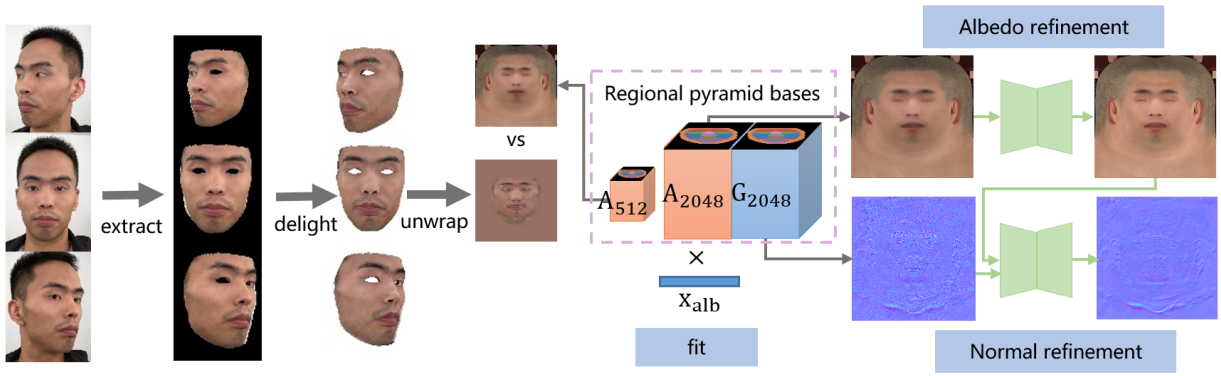


Fig. 8. Our albedo/normal map synthesis pipeline.

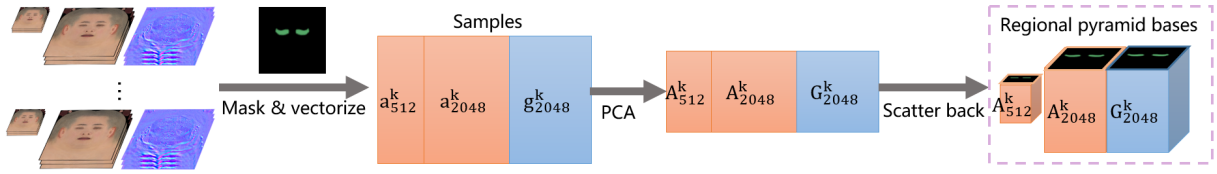


Fig. 9. Construction of the regional pyramid bases.

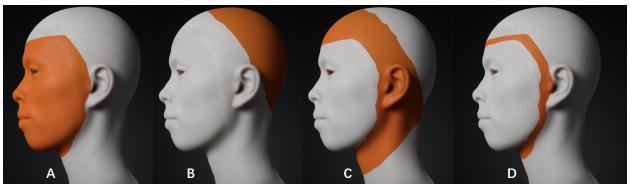
albedo/normal maps (from the dataset for constructing the 3DMM) into 4000 maps, which serve as ground-truth supervision for training the two networks. Then we perform regional fitting (Sec. 6.2) on the 4000 maps to get the fitted albedo/normal maps, which serve as inputs of the networks during training. We only use the facial regions out of the whole UV maps for computing training losses. Similar to *pix2pix* [Isola et al. 2017], we keep $L1$ loss and GAN loss in both networks. For albedo refinement, we additionally apply total variation loss to reduce artifacts and improve skin smoothness. The weights for $L1$, GAN and total variation losses are 100, 1, 0.001. For normal refinement, we additionally employ a pixel-wise cosine distances between the predictions and ground-truth maps to increase the accuracy of normal directions. The weights for $L1$, GAN and cosine distance losses are 100, 1, 0.001. The networks are trained with Adam optimizer for 75000 iterations.

Relation to Existing Approaches. There are three recent CNN-based approaches that can be adopted to synthesize high-resolution facial UV-maps, which are Yamaguchi et al. [2018], GANFIT [Gecer et al. 2019], and AvatarMe [Lattas et al. 2020]. However, these approaches cannot produce satisfactory results in our case. GANFIT [Gecer et al. 2019] needs about 50 times more training data than ours to train a GAN as the nonlinear parametric representation of texture maps. In their work, the 10,000 texture maps are obtained using unwrapped photos, where shadings and specular highlights are not removed. In our system, the 200 albedo maps and normal maps are created with very high-quality artistic efforts, where shadings and specular highlights are completely removed and hair-level details are preserved. It is rather difficult to extend our data amount to theirs while keeping such high data quality. Regarding the other two approaches, we also tried super-resolution based network as in Yamaguchi et al. [2018] and pure CNN-based synthesis in AvatarMe [Lattas et al. 2020], and found the results obtained with their approaches are generally inferior to ours. We present some comparison in Sec. 8.3.

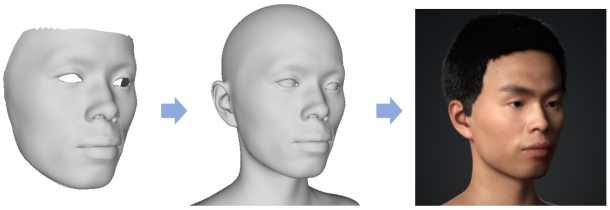
7 FULL HEAD RIG CREATION AND RENDERING

Head Completion. Although the scanned 200 models in out dataset are full head models, there are usually no reliable geometric constraints beyond facial regions for a RGB-D selfie user. We employ an algorithm to automatically complete a full head model given the recovered facial model. The regions involved in our algorithm are:

- A: facial region;
- B: back head region;
- C: intermediate region;
- D: overlapped region between A and C.



Our goal is to compute a full head shape such that region A matches the facial shape and region B matches a reference back head shape. The reason of using a reference shape for back head region B is it can further ease the difficulties to attach accessories like hair models. To this end, we construct a head morphable model using only regions B ∪ C of the 200 source models. Note that this model does not need strong expressive power as face models, thus we do not employ the technique in Sec. 5.3 but directly use PCA to extract bases. Given the recovered facial shape of a user, we apply a ridge regression similar to Sec. 5.1 to get the head 3DMM parameters, using the constraints of region B ∪ D. Then the full head model is obtained by combining the resulting shape (B ∪ C) with facial region A.



Accessories. We perform a hairstyle classification on the user's front photo (using a MobileNet [Howard et al. 2017] image classification model trained on labeled front photos) and attach the corresponding hair model (created by artists in advance) to the head model according to the predicted hairstyle label. There are in total 30 hairs models in different hairstyles in our system (see supplementary materials). For eyeballs, we use template models and calculate the positions and scales based on reference points on the head model. For teeth, we employ an upper teeth model and a lower teeth model. The upper teeth model is placed according to reference points near nose, and it remains still when facial expression changes. The lower teeth model is placed according to reference points on the chin. When mouth opens or closes, the lower teeth model moves with the chin. Note that the accessory models are not the focus of this work, their modeling and animation can be found in dedicated research work [Bérard et al. 2016, 2019; Velinov et al. 2018; Wu et al. 2016; Zoss et al. 2019, 2018].

Expression Rigging. We adopt a simple approach similar to Hu et al. [2017] to transfer generic FACS-based blendshapes to the target model to obtain expression blendshapes. Note that our approach can be extended to further acquire user's expression data and construct personalized blendshapes like Ichim et al. [2015].

Rendering. The recovered full head mesh model, as well as the high-quality albedo map and normal map, can be rendered with any physically based renderer. In this work, we show rendered results using Unreal Engine 4 (UE4), with the material composition templates provided by the engine [2020]. Since we do not model specular map in our approach, we use a same specular map from the material template for rendering all the results in this paper.

8 RESULTS AND EVALUATION

8.1 Acquisition and Processing Time

The selfie data acquisition typically takes less than 10 seconds (200-300 frames). The total processing time after data acquisition is about 15 seconds. Note that some of the processing steps like real-time landmark detection, coarse screening, and bilateral filtering can be computed on the smartphone client while the user is taking selfie. The preprocessed data are streamed to a server via WiFi during acquisition. The rest steps of the processing are computed on the server. Table 2 shows the runtime on our server with an Nvidia Tesla P40 GPU and an Intel Xeon E5-2699 CPU (22 cores). Note that the frame selection and expression blendshape generation are implemented with multi-thread acceleration and the total runtime is largely reduced thanks to parallel processing. Table 3 shows a time comparison with other avatar creation systems. In terms of total acquisition and processing time, our system provides a convenient and efficient solution for users to create high-quality digital humans.

Processing Step	Runtime
Landmark Detection	–
Coarse Screening	–
Bilateral Filtering	–
Frame Selection	0.2s
Initial Model Fitting	0.1s
Initial Texture	0.5s
Optimization	10s
Regional Parametric Fitting	1.5s
Detail Enhancement	1s
Head Completion	0.5s
Accessories	0.1s
Expression Rigging	1s
Total	~15s

Table 2. The runtime for each step in our system. Note that the first three steps are computed during data acquisition and thus do not need additional processing time. GPU and multi-thread CPU are used.

Avatar Creation System	Acquisition Time	Processing Time	Manual Interaction
[Ichim et al. 2015]	10 minutes	~1 hour	15 minutes
[Cao et al. 2016]	10 minutes	~1 hour	needed
[Hu et al. 2017]	<1 second	~6 minutes	–
Ours	<10 seconds	~15 seconds	–

Table 3. Time comparison with other avatar creation systems.

Method	Average ranking percentage
Face2Face [Thies et al. 2016]	30.3%
GANFIT [Gecer et al. 2019]	27.3%
N-ICP [Bouaziz et al. 2016]	16.6%
Ours	14.2%

Table 4. Identity verification results using rendered images. Lower ranking percentage means the rendered images are more similar to user photos from the point of view of a face recognition network.

8.2 Quality of Facial Geometry

We evaluate the quality of our recovered facial shapes in extensive experimental settings. The experiments include quantitative and qualitative comparisons of different variants of our approach and existing methods such as Face2Face [Hu et al. 2017; Thies et al. 2016; Yamaguchi et al. 2018], GANFIT [Gecer et al. 2019; Lattas et al. 2020], N-ICP [Bouaziz et al. 2016; Weise et al. 2011], etc. Note that all the results are obtained with our 3DMM bases for fair comparisons.

Quantitative Evaluation. We use the same workflow as the production of our dataset to manually create the ground-truth models of two users. Since the ground-truth obtained in this way is very expensive, we only perform numerical evaluations on the two models to get quantitative observations. The corresponding geometries recovered from RGB-D selfie data with different approaches are evaluated. The results are in Fig. 11. It can be seen from the results that our method yields the lowest mean errors, closely followed by N-ICP and the single-view variants of our method (*single rgbd+id*). Compared with N-ICP, our method performs better on detailed

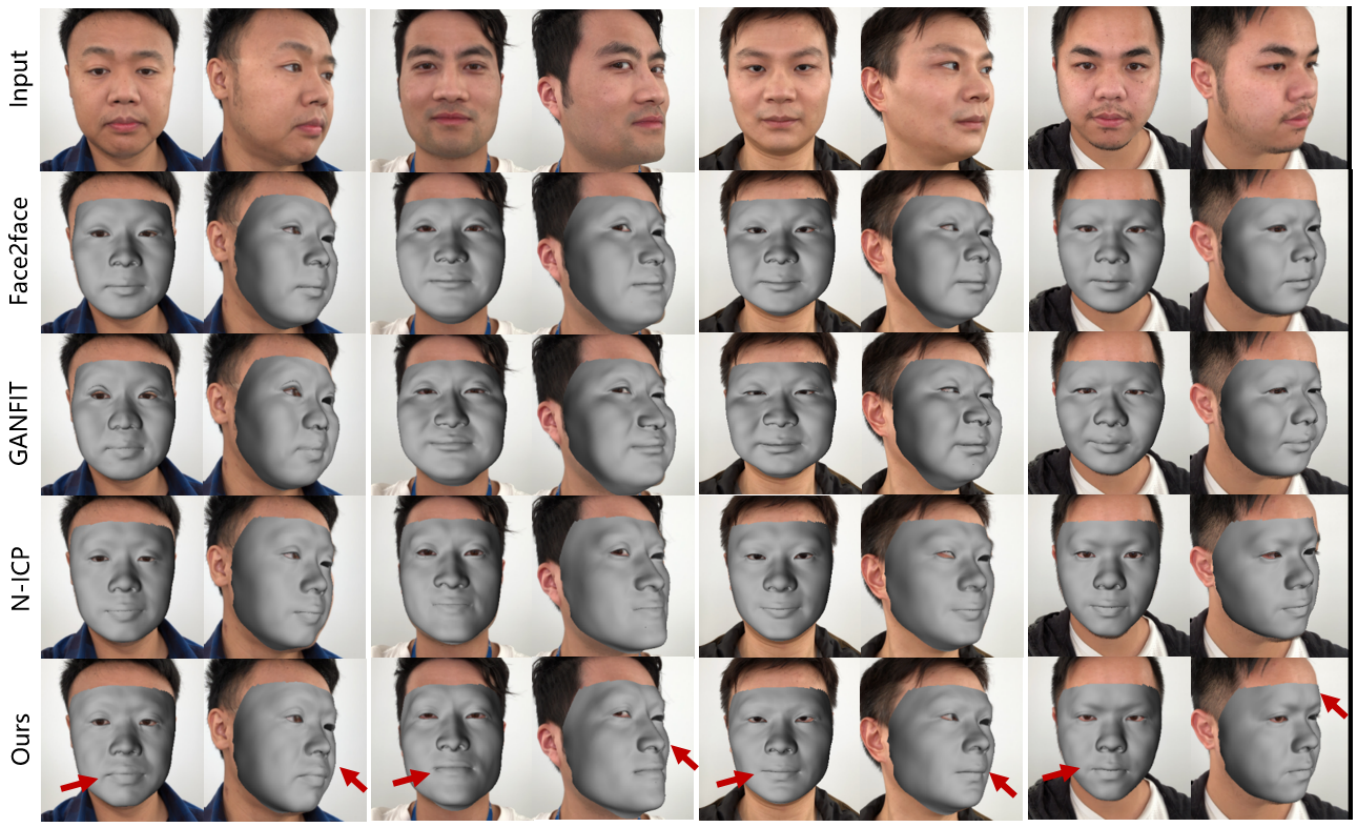


Fig. 10. Visual comparison with state-of-the-art approaches. As pointed out by the red arrows, our method is able to generate face geometries with more accurate cheek silhouettes and more faithful mouth shapes to the input photos. In comparison, the mouth shapes obtained by N-ICP lack personalized features and are similar to each other among all the subjects. For fair comparison, all the results are obtained with our 3DMM.

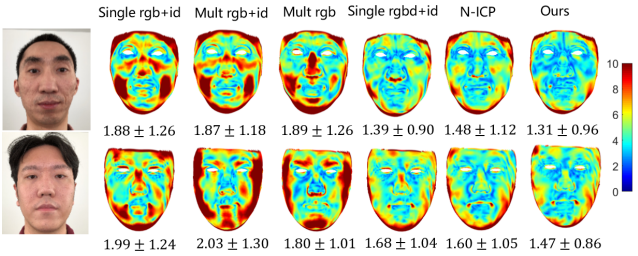


Fig. 11. Error maps for different approaches (mm).

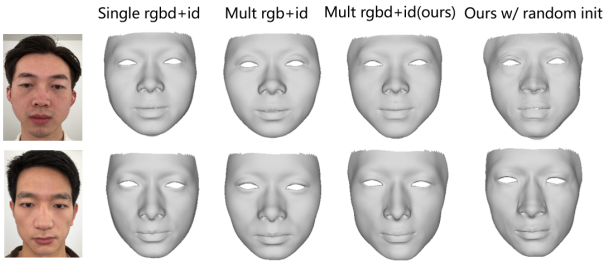


Fig. 12. Visual comparison for different variants of our method. The results obtained with multiview RGB-D data and identity loss (the second column from right) are generally more faithful than other results. We also show the results obtained without initial model fitting (the rightmost column), which are usually flawed and inferior to the full algorithm.

facial geometries near eyes, nose, and mouth. It conforms to our motivation that appearance constraints (photo loss and identity perceptual loss) help capture more accurate facial features.

Identity Verification using Rendered Images. To demonstrate the ability of our method to capture high-level perceptual identity features, we design a novel face verification experiment for further numerical evaluation. We collect selfie data from 30 subjects and put their selfie photos into a large face image dataset with over 40,000 photos of Asian people. Then the geometry models of the 30 subjects are obtained using different approaches. We use UE4 to render the recovered models into realistic images. For fair comparison of geometry models, we use the same albedo and normal maps for models obtained using different approaches for a user. The rendered realistic images are compared with all the images in the large dataset using a face recognition network [Deng et al. 2019]. We then calculate the ranking of each user's rendered image to his/her real selfie photo among all the face images. The average ranking percentage are shown in Table 4. Our method generally yields more recognizable shapes than other approaches.

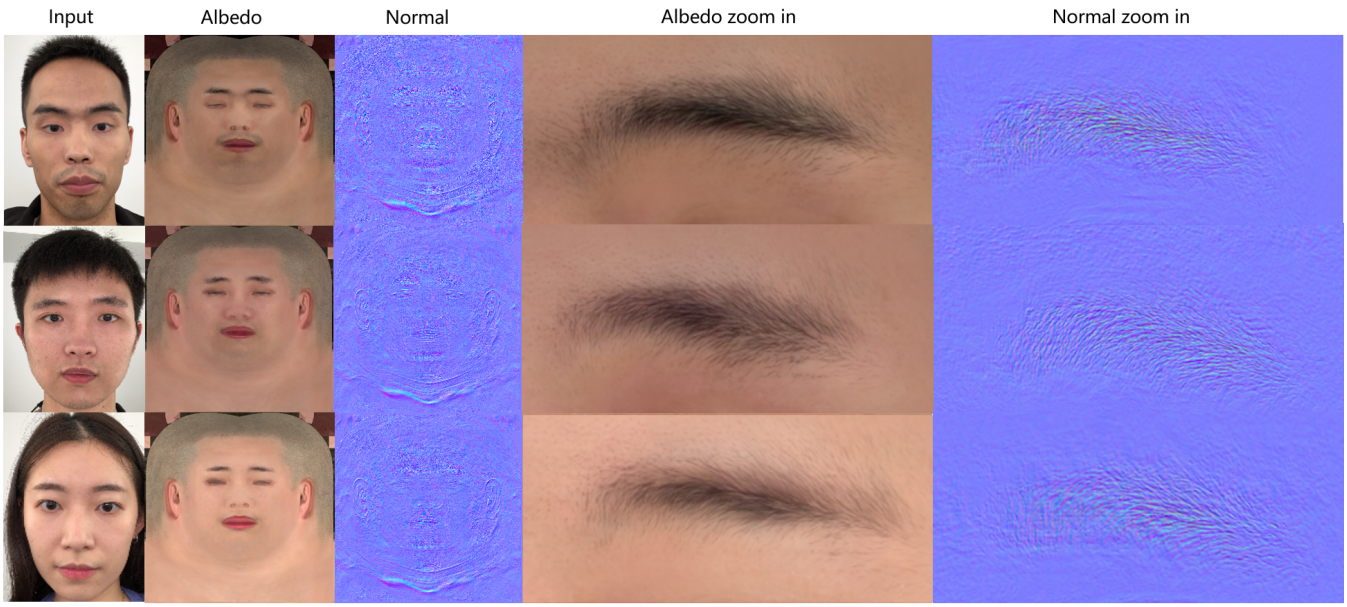


Fig. 13. Examples of our synthesized albedo and normal maps.

Qualitative Evaluation. Fig. 12 shows two examples of the shape models obtained using different variants of our method. The version with multiview RGB-D data generally outperforms other variants. Besides, as shown in the figure, random initialization of our optimization can lead to flawed models. The initial fitting in Sec. 5.1 improves the system robustness. We further show results comparisons between our method with Face2Face [Hu et al. 2017; Thies et al. 2016; Yamaguchi et al. 2018], GANFIT [Gecer et al. 2019; Lattas et al. 2020], and N-ICP [Bouaziz et al. 2016; Weise et al. 2011] in Fig. 10. In general, both our method and N-ICP can reconstruct more accurate facial shapes than the methods using only RGB data (Face2face and GANFIT). This can be clearly observed from the silhouettes near the cheek region in the results. Watching more closely, our method can recover more faithful and personalized facial shapes than N-ICP, especially near mouth region. The mouth shapes obtained with N-ICP are similar among all the faces, while our results preserve personalized mouth features and are more faithful to the photos. This can be explained by the additionally incorporated photometric loss and identity loss in our method.

8.3 Quality of Facial Reflectance

Results. Our method can produce albedo and normal maps with high-quality, realistic details while preserving major facial features of the users. Fig. 13 shows several examples of our obtained albedo and normal maps. Hair-level details in the albedo and normal maps are clearly visible. Fig. 15 shows the overlay results with synthesized albedo maps. The major facial features are consistent between the synthesized albedo maps and the input photos, especially in the eyebrow region. More importantly, the hair-level details near eyebrow region and the pore-level skin details are clearly visible. Note that the input RGB-D images in our method are in 640×480 resolution, where actual skin micro/meso-structures and hair-level details are hardly visible (see Fig. 14 left column). The synthesized skin/hair details by our approach are actually plausible hallucination, which is critical for realistic rendering of digital humans. We further show the intermediate results of our albedo synthesis algorithm in Fig. 14. The unwrapped result (left) is extracted from the input photos. It can be seen from the close-up display that it is blurry and low-quality. Moreover, the overall map seems dirty due to the imperfect lighting

Fig. 14. The intermediate results of our albedo synthesis algorithm. The unwrapped result (left) is extracted from the low-resolution input photos. It seems dirty due to imperfect lighting removal. The result obtained after regional fitting (middle) seems much cleaner and higher-quality. The final result (right) contains more realistic details.

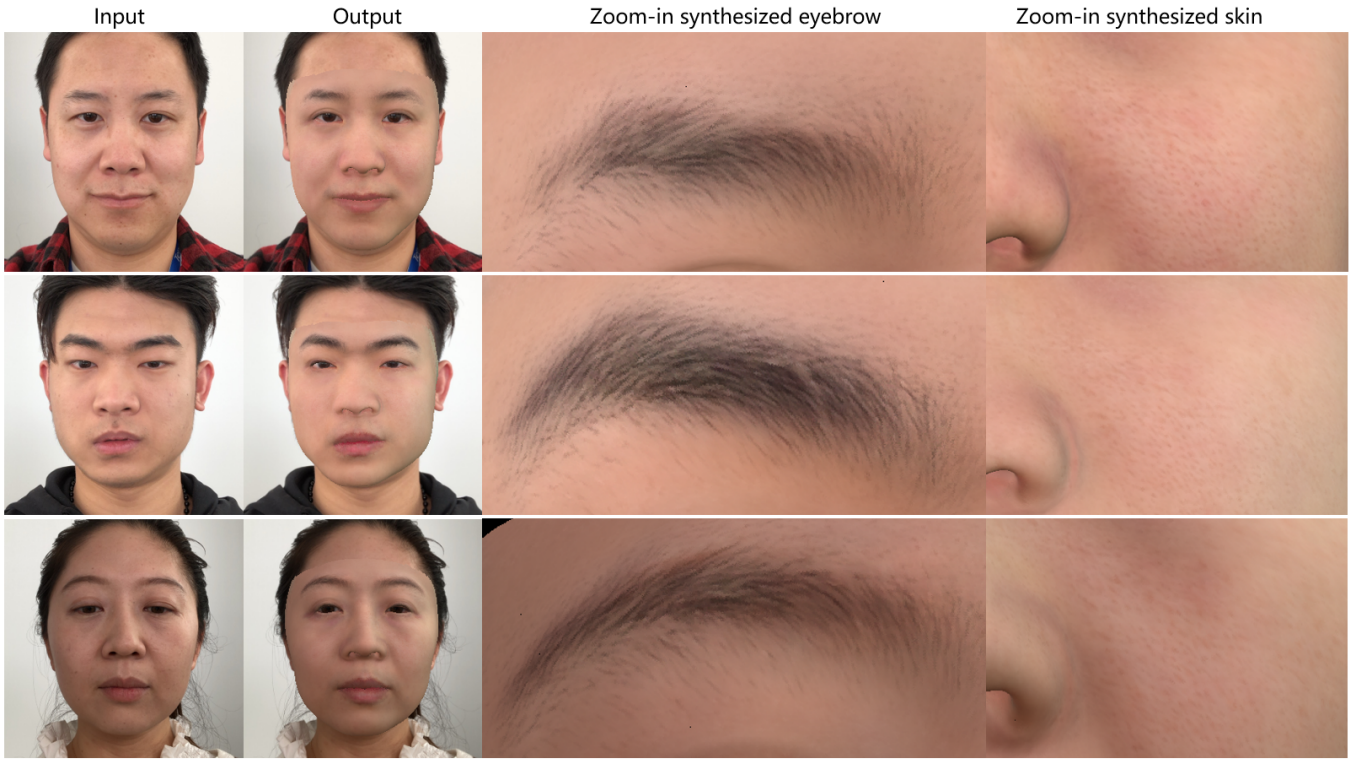


Fig. 15. Overlay results with our synthesized albedo maps. Note that the eyebrow and mouth shapes in our albedo maps are faithful to input photos.

removal step. After the regional fitting step, the result is much more clean but without hair/pore details. The final refined albedo map preserves the major facial features (e.g., the shape of eyebrows), while containing more high-quality, realistic details.

Pix2pix vs pix2pix-HD. We first compare two variants of our detail enhancement models, i.e., with *pix2pix* [Isola et al. 2017] and *pix2pix-HD* [Wang et al. 2018]. An example is shown below. The detail enhancement with *pix2pix* generally produces superior results than *pix2pix-HD*, especially around the eyebrow and cheek regions. Besides, we found *pix2pix-HD* is more difficult to train and the results are generally worse than *pix2pix*, possibly due to the small amounts of training data. Note that *pix2pix-HD* is adopted in AvatarMe [Lattas et al. 2020].

Comparison with Super-resolution Approach. Fig. 17 shows a comparison between our method and a state-of-the-art super-resolution network [Zhang et al. 2018], which is trained on the same dataset as our detail enhancement network. The super-resolution network generally cannot yield hair-level details around eyebrows. Our regional fitting algorithm produces over-smoothed eyebrow strands, which can be further refined into clear hair-level details while preserving major eyebrow shape. Note that Yamaguchi et al. [2018] employed a super-resolution model for high-resolution texture synthesis.

Comparison with State-of-the-art Approaches. We compare our result to two state-of-the-art high-resolution reflectance synthesis approaches [Yamaguchi et al. 2018], [Lattas et al. 2020] in Fig. 18.

Fig. 16. Comparison between pix2pix and pix2pix-HD. The results obtained with pix2pix-HD are generally with uneven skin colors and obvious artifacts.

The hair-level details around eyebrows in our result are clearly with higher quality. Besides, the pore-level details on the cheek are more

Fig. 17. Comparison with super-resolution (SR) approach. SR-based method tends to produce unnatural high-frequency details rather than natural details. In comparison, our method (right) can produce realistic details.

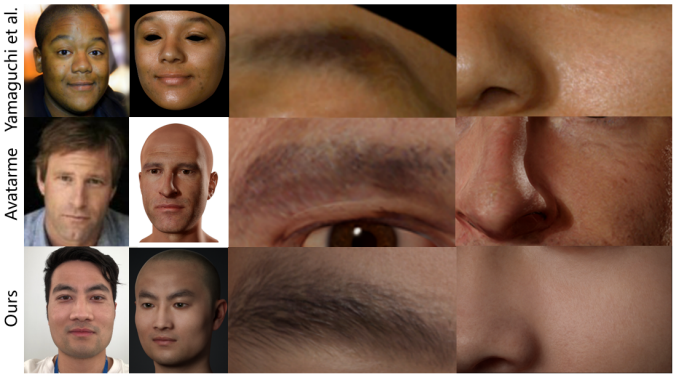


Fig. 18. Comparison with state-of-the-art high-resolution facial texture synthesis approaches. Our result contains more realistic hair/pore details.

realistic in our result, while those in the other approaches seem noisy and unnatural.

8.4 Comparison to Model-free Reconstruction

We notice some commercial systems (e.g., Bellus3D [2020]) utilize RGB-D selfies to reconstruct static 3D face models and directly extract texture maps from input photos. Their systems commonly employ a model-free reconstruction approach like KinectFusion [Newcombe et al. 2011] and the results usually seem very faithful to input photos. However, there are several drawbacks in their results. First, as shown in Fig. 20, their reconstructed meshes are not topologically consistent and are prone to flaws. It would be difficult to attach accessories and animate them. Moreover, the extracted texture maps contain shadows and highlights, which are undesired since they cause severe unnatural issues when the rendered lighting is different from the captured lighting. In comparison, our results are high-quality and ready for realistic rendering and animation (see Fig. 20).

8.5 Robustness to Different Inputs

We conduct experiments for a user taking selfies in different lighting conditions (Fig. 19). The recovered shape and reflectance remain consistent regardless of different lighting conditions and poses. Note that there is an inherent decoupling ambiguity between skin color and illumination. The resulting skin color in the right of the figure is actually a little bit more yellowish due to the yellower input photo. However, the facial structures (like the eyebrow shape) in the resulting reflectance map are consistent.

8.6 Rendered Results

Fig. 24 shows some of our rendered results with UE4. Thanks to the high-fidelity geometry and reflectance maps, the rendered results

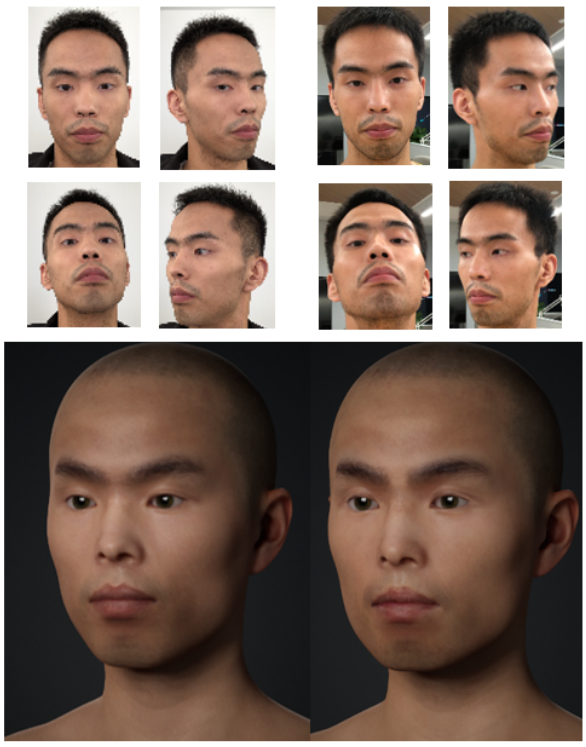


Fig. 19. Results obtained using RGB-D data captured in different lighting conditions, but rendered in an identical lighting setting. The recovered facial structures are consistent in the two lighting conditions. Note that the skin color result in the right rendered image is a little bit more yellowish than the left. This is because the photos in the right is captured with a different white balance setting from the left. There is an inherent decoupling ambiguity between skin color and illumination in our approach.

are realistic and faithful to input faces. Note the wrinkle details on the lips, the pore-level details on the cheek, the hair-level details of the eyebrows. Fig. 21 shows two examples of our results with attached hair models, which are retrieved from our hair model database by performing hairstyle classification on the selfie photos. More results are in the supplementary video.

8.7 Limitations

Our approach does not take into account cross-ethnicity generalization. Since our 3DMM is constructed from East-Asian subjects, our evaluations are designed on the same ethnical population. We tried to directly apply our approach on some people from other ethnicities and Fig. 23 shows two examples. Although the results roughly resemble the subjects being captured, some ethnicity-specific facial features are not recovered. Besides, our approach cannot model facial hairs like moustache and beard. Children or aged people beyond the age scope of our dataset are also not considered in our approach.

9 APPLICATIONS

The avatars created with our method are animation-ready. Animations can be retargeted from existing characters [Bouaziz and Pauly 2014], or interactively keyframe-posed [Ichim et al. 2015],

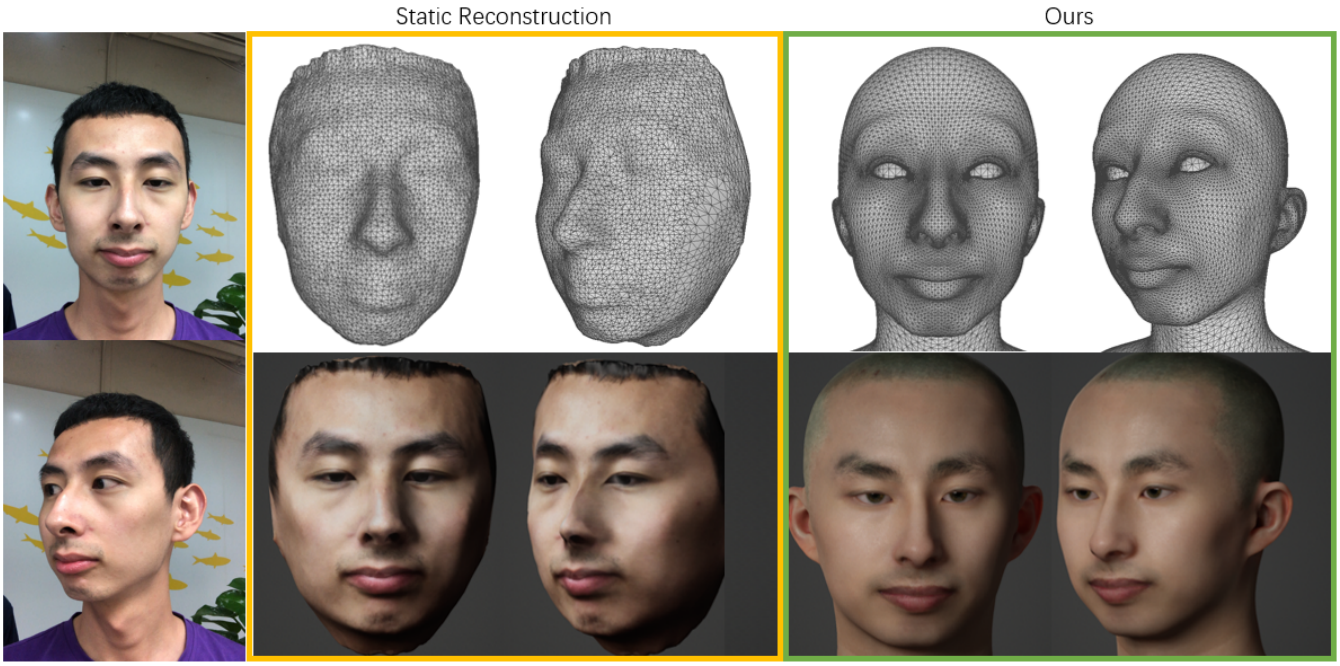


Fig. 20. Comparison to model-free reconstruction (e.g., Bellus3D [2020]). The model-free reconstruction are not topologically consistent and are prone to flaws, which cause difficulties when being animated. The extracted textures contain undesired shadows and highlights that would result unnatural renderings. Our results are topological consistent and ready for animation. The high-quality albedo/normal maps make our rendering very realistic.

Fig. 21. Rendering results with hair models.

Fig. 22. Snapshots of our lip-sync animation. See supplementary video.

Fig. 23. Results on other ethnicities different from our 3DMM data source. The results roughly resemble the subjects but lack ethnicity-specific features.

or even transferred from facial tracking applications [Weise et al. 2011]. We demonstrate an application of lip-sync animation in the supplementary video, where a real-time multimodal synthesis system [Yu et al. 2019] is adopted to simultaneously synthesize speech and expression blendshape weights given input texts. Fig. 22 shows several snapshots of the animation. The application enables users to conveniently create high-fidelity, realistic digital humans that can be interacted with in real time. We also include another lip-sync animation result driven by speech inputs [Huang et al. 2020] in the supplementary video.

10 CONCLUSION

We have introduced a fully automatic system that can produce high-fidelity 3D facial avatars with a commercial RGB-D selfie camera. The system is robust, efficient, and consumer-friendly. The total acquisition and processing for a user can be finished in less than 30 seconds. The generated geometry models and reflectance maps are in very high fidelity and quality. With a physically based renderer, the assets can be used to render highly realistic digital humans. Our system provides an excellent consumer-level solution for users to create high-fidelity digital humans.

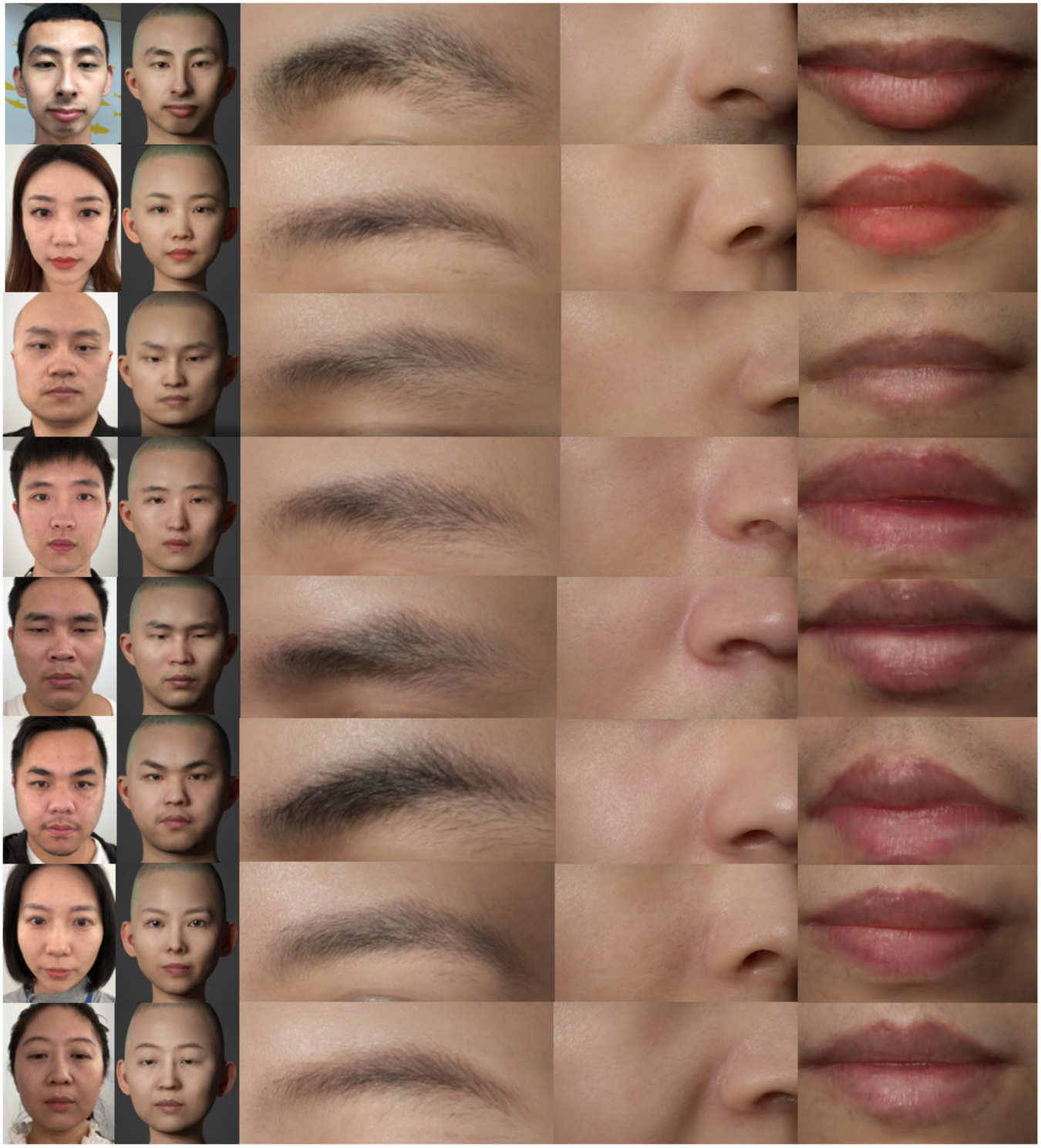


Fig. 24. Rendered results with UE4 rendering engine. Our method can recover faithful face models with high-quality, realistic hair/pore/wrinkle details. Note that the selfie photos are with perspective camera projection, while the rendered results are with orthogonal camera projection.

Future Work The animation with generic expression blendshapes are not satisfactory. We intend to extend our system to capture personalized expression blendshapes like Ichim et al. [2015]. Besides, the current system employs very simple approaches to handle accessories like hair, eyeballs, and teeth. We intend to incorporate more advanced methods to model accessories.

ACKNOWLEDGMENTS

We would like to thank Cheng Ge and other colleagues at Tencent NExT Studios for valuable discussions; Shaobing Zhang, Han Liu, Caisheng Ouyang, Yanfeng Zhang, and other colleagues at Tencent AI Lab for helping us with the videos; and all the subjects for allowing us to use their selfie data for testing.

REFERENCES

Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. 2009. The Digital Emily project: photoreal facial modeling and animation. In *ACM SIGGRAPH 2009 Courses*. ACM.

K Somani Arun, Thomas S Huang, and Steven D Blostein. 1987. Least-squares fitting of two 3-D point sets. *IEEE Transactions on pattern analysis and machine intelligence* 5 (1987), 698–700.

Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. 2010. High-quality single-shot capture of facial geometry. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29, 4 (2010), 1–9.

Bellus3D. 2020. *Bellus3D.* Retrieved Sept 18, 2020 from https://www.bellus3d.com/

Pascal Bérard, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. Lightweight eye capture using a parametric model. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35, 4 (2016), 1–12.

Pascal Bérard, Derek Bradley, Markus Gross, and Thabo Beeler. 2019. Practical Person-Specific Eye Rigging. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 441–454.

Volker Blanz and Thomas Vetter. 1999. A morphable model for the synthesis of 3D faces. In *Proc. SIGGRAPH*. ACM, 187–194.

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

James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. 2016. A 3d morphable model learnt from 10,000 faces. In *Proc. CVPR*. IEEE, 5543–5552.

Sofien Bouaziz and Mark Pauly. 2014. *Semi-supervised facial animation retargeting.* Technical Report.

Sofien Bouaziz, Andrea Tagliasacchi, Hao Li, and Mark Pauly. 2016. Modern techniques and applications for real-time non-rigid registration. In *ACM SIGGRAPH Asia 2016 Courses*. 1–25.

Sofien Bouaziz, Yangang Wang, and Mark Pauly. 2013. Online modeling for realtime facial animation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4 (2013), 1–10.

Peter J Burt and Edward H Adelson. 1983. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.* 2, 4 (1983), 217–236.

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

Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. 2016. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph. (Proc. SIGGRAPH)* 35, 4 (2016), 1–12.

Menglei Chai, Linjie Luo, Kalyan Sunkavalli, Nathan Carr, Sunil Hadap, and Kun Zhou. 2015. High-quality hair modeling from a single portrait photo. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 6 (2015), 1–10.

Yajing Chen, Fanzi Wu, Zeyu Wang, Yibing Song, Yonggen Ling, and Linchao Bao. 2020. Self-supervised Learning of Detailed 3D Face Reconstruction. *IEEE Transactions on Image Processing* (2020).

Yen-Lin Chen, Hsiang-Tao Wu, Fuhao Shi, Xin Tong, and Jinxiang Chai. 2013. Accurate and robust 3d facial capture using a single rgbd camera. In *Proc. ICCV*. IEEE, 3615–3622.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. 2000. Acquiring the reflectance field of a human face. In *Proc. of SIGGRAPH*. ACM, 145–156.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proc. CVPR*. IEEE.

Pengfei Dou and Ioannis A Kakadiaris. 2018. Multi-view 3D face reconstruction with deep recurrent neural networks. *Image and Vision Computing* 80 (2018), 80–91.

Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 2020. 3D Morphable Face Models–Past, Present and Future. *ACM Trans. Graph.* (2020).

EpicGames. 2020. *Rendering Digital Humans in Unreal Engine 4.* Retrieved May 20, 2020 from https://docs.unrealengine.com/en-US/Resources/Showcases/DigitalHumans/index.html

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

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of personalized 3D face rigs from monocular video. *ACM Trans. Graph.* 35, 3 (2016), 1–15.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proc. CVPR*. IEEE, 2414–2423.

Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. 2019. Ganfit: Generative adversarial network fitting for high fidelity 3d face reconstruction. In *Proc. CVPR*. IEEE, 1155–1164.

Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. 2018. Unsupervised Training for 3D Morphable Model Regression. In *Proc. CVPR*. IEEE, 8377–8386.

Yudong Guo, Juyong Zhang, Jianfei Cai, Boyi Jiang, and Jianmin Zheng. 2019. CNN-Based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 6 (2019), 1294–1307.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. 2015. Single-view hair modeling using a hairstyle database. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4 (2015), 1–9.

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

Huirong Huang, Zhiyong Wu, Shiyin Kang, Dongyang Dai, Jia Jia, Tianxiao Fu, Deyi Tuo, Guangzhi Lei, Peng Liu, Dan Su, Dong Yu, and Helen Meng. 2020. Speaker Independent and Multilingual/Mixlingual Speech-Driven Talking Head Generation Using Phonetic Posteriorgrams. *arXiv preprint arXiv:2006.11610* (2020).

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D avatar creation from hand-held video input. *ACM Trans. Graph. (Proc. SIGGRAPH)* 34, 4 (2015), 1–14.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*. IEEE, 1125–1134.

Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. 2017. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In *Proc. ICCV*. IEEE, 1031–1039.

Ira Kemelmacher-Shlizerman and Ronen Basri. 2011. 3D face reconstruction from a single image using a single reference face shape. *IEEE transactions on pattern analysis and machine intelligence* 33, 2 (2011), 394–405.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. 2020. AvatarMe: Realistically Renderable 3D Facial Reconstruction" in-the-wild". In *Proc. CVPR*. IEEE.

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision* 81, 2 (2009), 155.

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

Linjie Luo, Hao Li, and Szymon Rusinkiewicz. 2013. Structure-aware hair capture. *ACM Trans. Graph. (Proc. SIGGRAPH)* 32, 4 (2013), 1–12.

Marcel Lüthi, Thomas Gerig, Christoph Jud, and Thomas Vetter. 2017. Gaussian process morphable models. *IEEE transactions on pattern analysis and machine intelligence* 40, 8 (2017), 1860–1873.

Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. 2018. paGAN: real-time avatars using dynamic textures. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 1–12.

Thomas Neumann, Kiran Varanasi, Stephan Wenger, Markus Wacker, Marcus Magnor, and Christian Theobalt. 2013. Sparse localized deformation components. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 32, 6 (2013), 1–10.

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. ISMAR*. IEEE, 127–136.

Sylvain Paris and Frédo Durand. 2009. A fast approximation of the bilateral filter using a signal processing approach. *International journal of computer vision* 81, 1 (2009), 24–52.

Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition. In *Proc. BMVC*.

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

R3ds. 2020. *Wrap 3*. Retrieved May 20, 2020 from https://www.russian3dscanner.com/

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5 (2001), 34–41.

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

Sami Romdhani and Thomas Vetter. 2005. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proc. CVPR*, Vol. 2. IEEE, 986–993.

Shunsuke Saito, Liwen Hu, Chongyang Ma, Hikaru Ibayashi, Linjie Luo, and Hao Li. 2018. 3D hair synthesis using volumetric variational autoencoders. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 1–12.

Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. 2017. Photorealistic facial texture inference using deep neural networks. In *Proc. CVPR*, Vol. 3. IEEE.

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

Fuhao Shi, Hsiang-Tao Wu, Xin Tong, and Jinxiang Chai. 2014. Automatic acquisition of high-fidelity facial performances using monocular videos. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 33, 6 (2014), 1–13.

J Rafael Tena, Fernando De la Torre, and Iain Matthews. 2011. Interactive region-based linear 3D face models. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4 (2011), 1–10.

Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. 2018. Self-Supervised Multi-Level Face Model Learning for Monocular Reconstruction at Over 250 Hz. In *Proc. CVPR*. IEEE.

Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. 2017. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. ICCV*, Vol. 2. IEEE, 5.

Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. 2015. Real-time expression transfer for facial reenactment. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 34, 6 (2015), 183–1.

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

Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. 2017. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *Proc. CVPR*. IEEE, 1493–1502.

Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard Medioni. 2018. Extreme 3D Face Reconstruction: Seeing Through Occlusions. In *Proc. CVPR*. IEEE.

Luan Tran and Xiaoming Liu. 2018. Nonlinear 3D Face Morphable Model. In *Proc. CVPR*. IEEE.

Zdravko Velinov, Marios Papas, Derek Bradley, Paulo Gotardo, Parsa Mirdehghan, Steve Marschner, Jan Novák, and Thabo Beeler. 2018. Appearance capture and modeling of human teeth. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37, 6 (2018), 1–13.

Javier von der Pahlen, Jorge Jimenez, Etienne Danvoye, Paul Debevec, Graham Fyffe, and Oleg Alexander. 2014. Digital ira and beyond: creating real-time photoreal digital actors. In *ACM SIGGRAPH 2014 Courses*. ACM.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proc. CVPR*. IEEE.

Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. 2005. Modeling hair from multiple views. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24, 3 (2005), 816–820.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime performance-based facial animation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 4 (2011), 1–10.

Chenglei Wu, Derek Bradley, Pablo Garrido, Michael Zollhöfer, Christian Theobalt, Markus H Gross, and Thabo Beeler. 2016. Model-based teeth reconstruction. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 35, 6 (2016), 220–1.

Fanzi Wu, Linchao Bao, Yajing Chen, Yonggen Ling, Yibing Song, Songnan Li, King Ngi Ngan, and Wei Liu. 2019. Mvf-net: Multi-view 3d face morphable model regression. In *Proc. CVPR*. IEEE, 959–968.

Shugo Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. 2018. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Trans. Graph. (Proc. SIGGRAPH)* 37, 4 (2018), 1–14.

Chengzhu Yu, Heng Lu, Na Hu, Meng Yu, Chao Weng, Kun Xu, Peng Liu, Deyi Tuo, Shiyin Kang, Guangzhi Lei, Dan Su, and Dong Yu. 2019. Durian: Duration informed attention network for multimodal synthesis. *arXiv preprint arXiv:1909.01700* (2019).

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Proc. ECCV*. 286–301.

Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. 2016. Face alignment across large poses: A 3d solution. In *Proc. CVPR*. IEEE, 146–155.

Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. 2015. High-fidelity pose and expression normalization for face recognition in the wild. In *Proc. CVPR*. IEEE, 787–796.

Michael Zollhöfer, Michael Martinek, Günther Greiner, Marc Stamminger, and Jochen Süßmuth. 2011. Automatic reconstruction of personalized avatars from 3D face scans. *Computer Animation and Virtual Worlds* 22, 2-3 (2011), 195–202.

Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. 2014. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4 (2014), 1–12.

Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. 2018. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 523–550.

Gaspard Zoss, Thabo Beeler, Markus Gross, and Derek Bradley. 2019. Accurate markerless jaw tracking for facial performance capture. *ACM Trans. Graph. (Proc. SIGGRAPH)* 38, 4 (2019), 1–8.

Gaspard Zoss, Derek Bradley, Pascal Bérard, and Thabo Beeler. 2018. An empirical rig for jaw animation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 37, 4 (2018), 1–12.