FFHQ-UV: Normalized Facial UV-Texture Dataset for 3D Face Reconstruction

Haoran Bai[1*] Di Kang[2] Haoxian Zhang[2] Jinshan Pan[1†] Linchao Bao[2]
[1]Nanjing University of Science and Technology [2]Tencent AI Lab

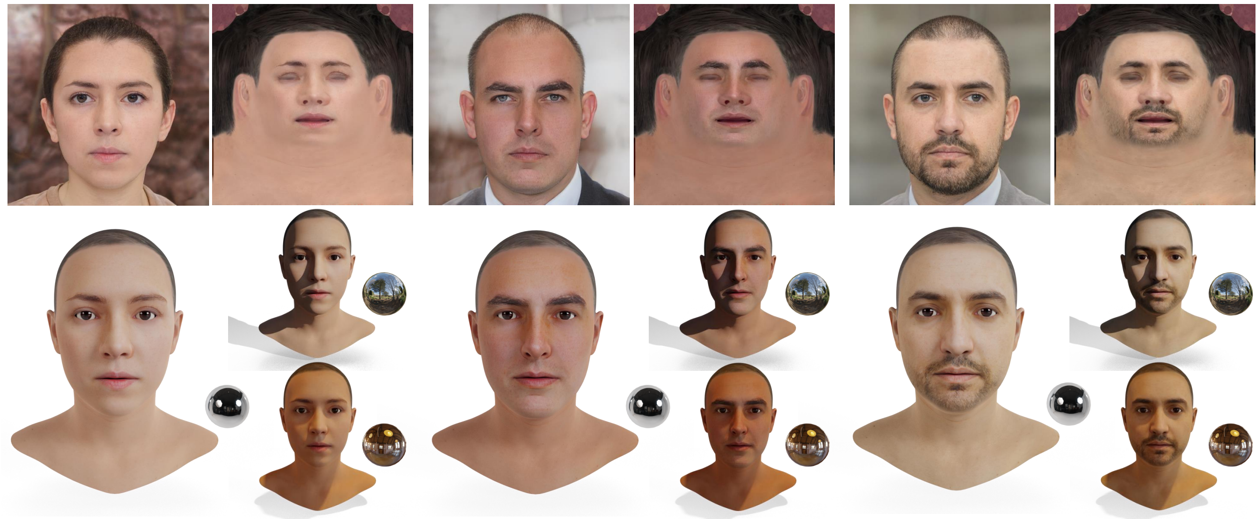

Figure 1. **Examples of the proposed FFHQ-UV dataset.** From left-to-right and top-to-bottom are the normalized face images after editing, the produced texture UV-maps, and the rendered images under different lighting conditions. The proposed dataset is derived from FFHQ and preserves the most variations in FFHQ. The facial texture UV-maps in the proposed dataset are with even illuminations, neutral expressions, and cleaned facial regions (e.g. no eyeglasses and hair), which are ready for realistic renderings.

Abstract

We present a large-scale facial UV-texture dataset that contains over 50,000 high-quality texture UV-maps with even illuminations, neutral expressions, and cleaned facial regions, which are desired characteristics for rendering realistic 3D face models under different lighting conditions. The dataset is derived from a large-scale face image dataset namely FFHQ, with the help of our fully automatic and robust UV-texture production pipeline. Our pipeline utilizes the recent advances in StyleGAN-based facial image editing approaches to generate multi-view normalized face images from single-image inputs. An elaborated UV-texture extraction, correction, and completion procedure is then applied to produce high-quality UV-maps from the normalized face images. Compared with existing UV-texture datasets, our dataset has more diverse and higher-quality texture maps. We further train a GAN-based texture decoder as the nonlinear texture basis for parametric fitting based 3D face reconstruction. Experiments show that our method improves the reconstruction accuracy over state-of-the-art approaches, and more importantly, produces high-quality texture maps that are ready for realistic renderings. The dataset, code, and pre-trained texture decoder are publicly available at https://github.com/csbhr/FFHQ-UV.

1. Introduction

Reconstructing the 3D shape and texture of a face from single or multiple images is an important and challenging task in both computer vision and graphics communities. Since the seminal work by Blanz and Vetter [3] showed that the reconstruction can be effectively achieved by parametric fitting with a linear statistical model, namely 3D Morphable Model (3DMM), it has received active research efforts in the past two decades [14]. While most 3DMM-based reconstruction approaches focused on improving the shape estimation accuracy, only a few works addressed the problem on texture UV-map recovery [2, 15, 23, 24, 26, 35, 41].

There are two key aspects that deserve attention in the texture map recovery problem, which are the *fidelity* and the *quality* of the acquired texture maps. In order to recover a

[*] Work done during an internship at Tencent AI Lab.
[†] Corresponding author.

high-fidelity texture map that better preserves the face identity of the input image, the texture basis in a 3DMM needs to have larger expressive capacities. On the other hand, a higher-quality texture map requires the face region to be evenly illuminated and without undesired hairs or accessories, such that the texture map can be used as facial assets for rendering under different lighting conditions.

The method GANFIT [15] trains a generative adversarial network (GAN) [19] from 10,000 UV-maps as a texture decoder to replace the linear texture basis in 3DMM to increase the expressiveness. However, their UV-maps in the training dataset are extracted from unevenly illuminated face images, and the resulting texture maps contain obvious shadows and are not suitable for differently lighted renderings. The same problem exists in another work [24] based on UV-GAN [11]. The work AvatarMe [23] combines a linear texture basis fitting with a super-resolution network trained from high-quality texture maps of 200 individuals under controlled conditions. HiFi3DFace [2] improves the expressive capacity of linear texture basis by introducing a regional fitting approach and a detail refinement network, which is also trained from 200 texture maps. The Normalized Avatar work [26] trains a texture decoder from a larger texture map dataset with over 5,000 subjects, consisting of high-quality scan data and synthetic data. Although the quality of the resulting texture maps of these methods is pretty high, the reconstruction fidelity is largely limited by the number of subjects in the training dataset. Besides, all these texture map datasets are not publicly available. A recent high-quality, publicly accessible texture map dataset is in the Facescape dataset [42], obtained in a controlled environment. However, the dataset only has 847 identities.

In this paper, we intend to contribute a large-scale, publicly available facial UV-texture dataset consisting of high-quality texture maps extracted from different subjects. To build such a large-scale dataset, we need a fully automatic and robust pipeline that can produce high-quality texture UV-maps from large-scale "in-the-wild" face image datasets. For the produced texture map, we expect it to have even illumination, neutral expression, and complete facial texture without occlusions such as hair or accessories. This is not a trivial task, and there exist several challenges: 1) The uncontrolled conditions of the in-the-wild face images cannot provide high-quality normalized textures; 2) From a single-view face image, the complete facial texture cannot be extracted; 3) Imperfect alignment between the face image and the estimated 3D shape would cause unsatisfactory artifacts in the unwrapped texture UV-maps.

To address these issues, we first utilize StyleGAN-based image editing approaches [1, 21, 37] to generate multi-view normalized faces from a single in-the-wild image. Then a UV-texture extraction, correction, and completion process is developed to fix unsatisfactory artifacts caused by imperfect 3D shape estimation during texture unwrapping, so that high-quality texture UV-maps can be produced stably. With the proposed pipeline, we construct a large-scale normalized facial UV-texture dataset, namely FFHQ-UV, based on the FFHQ dataset [20]. The FFHQ-UV dataset inherits the data diversity of FFHQ, and consists of high-quality texture UV-maps that can directly serve as facial assets for realistic digital human rendering (see Fig. 1 for a few examples). We further train a GAN-based texture decoder using the proposed dataset, and demonstrate that both the fidelity and the quality of the reconstructed 3D faces with our texture decoder get largely improved.

In summary, our main contributions are:

- The first large-scale, publicly available normalized facial UV-texture dataset, namely FFHQ-UV, which contains over 50,000 high-quality, evenly illuminated facial texture UV-maps that can be directly used as facial assets for rendering realistic digital humans.
- A fully automatic and robust pipeline for producing the proposed UV-texture dataset from a large-scale, in-the-wild face image dataset, which consists of StyleGAN-based facial image editing, elaborated UV-texture extraction, correction, and completion procedure.
- A 3D face reconstruction algorithm that outperforms state-of-the-art approaches in terms of both fidelity and quality, based on the GAN-based texture decoder trained with the proposed dataset.

2. Related Work

3D Face Reconstruction with 3DMM. The 3D Face Morphable Model (3DMM) introduced by Blanz and Vetter [3] represents a 3D face model with a linear combination of shape and texture basis, which is derived using Principal Component Analysis (PCA) from topologically aligned 3D face models. The task of reconstructing 3D face models from images is typically tackled by estimating the 3DMM parameters using either optimization-based fitting or learning-based regression approaches [14, 46]. Beyond the PCA-based linear basis, various nonlinear bases emerged to enlarge the 3DMM representation capacity [5, 15, 18, 29, 33, 39, 40]. With a neural network-based nonlinear 3DMM basis, the 3D face reconstruction turns into a task of finding the best latent codes of a mesh decoder [5] or a texture decoder [15], which we still term as "3DMM fitting". For a thorough review of 3DMM and related reconstruction methods, please refer to the recent survey [14].

Facial UV-Texture Recovery. While the texture basis in the original 3DMM [3] is represented by vertex colors of a mesh, recent methods [2, 9, 15, 23–26, 35, 41] start to employ UV-map texture representation in order to fulfil high-resolution renderings. These methods can be categorized into two lines: image translation-based approaches [2, 23, 35, 41] or texture decoder-based approaches [15, 24, 26].

Table 1. Comparisons with existing UV-texture datasets, where $*$ denotes the dataset which is captured under controlled conditions.

Datasets	# images	Resolution	Even illum.	Public avail.
LSFM* [4]	10,000	512 × 512	×	×
AvatarMe* [23]	200	6144 × 4096	✓	×
HiFi3DFace* [2]	200	2048 × 2048	✓	×
Facescape* [42]	847	4096 × 4096	✓	✓
WildUV [11]	5,638	377 × 595	×	×
NormAvatar [26]	5,601	256 × 256	✓	×
FFHQ-UV (Ours)	54,165	1024 × 1024	✓	✓

The former approaches usually obtain a low-quality texture map and then perform an image translation to convert the low-quality texture map to a high-quality one [2,23,35,41]. The latter approaches typically train a texture decoder as the nonlinear 3DMM texture basis and then employ a 3DMM fitting algorithm to find the best latent code for a reconstruction [15, 24, 26]. Both the image translation operation and texture decoder need a high-quality UV-texture dataset for training. Unfortunately, to the best of our knowledge, there is no publicly available, high-quality facial UV-texture dataset in such a large scale that the data has enough diversity for practical applications (see Tab. 1 for a summary of the sizes of the datasets used in literature).

Facial Image Normalization. Normalizing a face image refers to the task of editing the face image such that the resulting face is evenly illuminated and in neutral expression and pose [10, 30]. In our goal of extracting high-quality, "normalized" texture maps from face images, we intend to obtain three face images in frontal/left/right views from a single image, such that the obtained images have even illumination, neutral expression, and no facial occlusions by forehead hairs or eyeglasses. To achieve this, we utilize recent advances in StyleGAN-based image editing approaches [1, 17, 31, 37]. In these approaches, an image is first projected to a latent code of an FFHQ-pretrained StyleGAN [20, 21] model through GAN inversion methods [34, 38], and then the editing can be performed in the latent space, where the editing directions can be discovered through the guidance of image attributes or labels.

3. FFHQ-UV: Normalized UV-Texture Dataset

In this section, we first describe the full pipeline for producing our normalized UV-texture dataset from in-the-wild face images (Sec. 3.1). Then we present extensive studies to analyze the diversity and quality of the dataset (Sec. 3.2).

3.1. Dataset Creation

The dataset creation pipeline, shown in Fig. 2, consists of three steps: StyleGAN-based facial image editing (Sec. 3.1.1), facial UV-texture extraction (Sec. 3.1.2), and UV-texture correction & completion (Sec. 3.1.3).

3.1.1 StyleGAN-Based Facial Image Editing

To extract high-quality texture maps from in-the-wild face images, we first derive multi-view, normalized face images from single-view images, where the resulting face images have even illumination, neutral expression, and no occlusions (e.g., eyeglasses, hairs). Specifically, we employ StyleFlow [1] and InterFaceGAN [37] to automatically edit the image attributes in $\mathcal{W}+$ latent space of StyleGAN2 [21]. For each in-the-wild face image I, we first obtain its latent code w in $\mathcal{W}+$ space using the GAN inversion method e4e [38], and then detect the attribute values of the inverted image $I_{inv} = G(w)$ from StyleGAN generator G, so that we can normalize these properties in the following semantic editing. The attributes we intend to normalize include lighting, eyeglasses, hair, head pose, and facial expression. The lighting condition, represented as spherical harmonic (SH) coefficients, are predicted using DPR model [44], and the other attributes are detected using Microsoft Face API [28].

For the lighting normalization, we set the target lighting SH coefficients to keep only the first dimension and reset the rest dimensions to zeros, and then use StyleFlow [1] to get evenly illuminated face images. As in the SH representation, only the first dimension of the SH coefficients represents uniform lighting from all directions, while the other dimensions represent lighting from certain directions which are undesired. After the lighting normalization, we normalize the eyeglasses, head pose, and hair attributes by setting their target values to 0, and obtain the edited latent code w'. For the facial expression attribute, similar to InterFaceGAN [37], we find a direction β of editing facial expression using SVM, and achieve the normalized latent code $\hat{w}$ by walking along the direction β starting from w'. To avoid over-editing, we further introduce an expression classifier to decide the stop condition for walking. Here, we obtain the normalized face image $I_n = G(\hat{w})$. Finally, two side-view face images I_n^l and I_n^r are generated using StyleFlow [1] by modifying the head pose attribute.

3.1.2 Facial UV-Texture Extraction

The process of extracting UV-texture from a face image, also termed "unwrapping", requires a single-image 3D face shape estimator. We train a Deep3D model [13] with the recent 3DMM basis HiFi3D++ [8] to regress the 3D shape in 3DMM coefficients, as well as the head pose parameters, from each normalized face image. Then the facial UV-texture is unwrapped by projecting the input image onto the 3D face model. In addition, we employ a face parsing model [45] to predict the parsing mask for the facial region, so that non-facial regions can be excluded from the unwrapped texture UV-map. In this way, for each face we obtain three texture maps, T_f, T_l, and T_r from frontal, left, and right views, respectively. To fuse them together, we first perform a color matching between them to avoid color jumps. The color matching is computed from T_l and T_r to

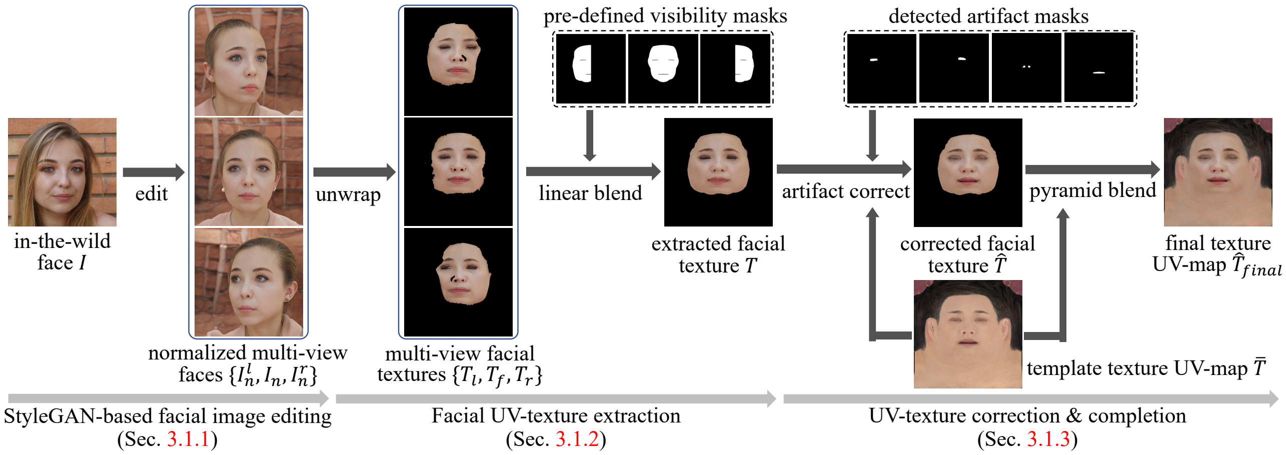
pre-defined visibility masks
detected artifact masks
in-the-wild face I
edit
unwrap
linear blend
artifact correct
pyramid blend
extracted facial texture T
corrected facial texture $\hat{T}$
final texture UV-map $\hat{T}_{final}$
template texture UV-map $\bar{T}$
normalized multi-view faces $\{I_n^l, I_n, I_n^r\}$
multi-view facial textures $\{T_l, T_f, T_r\}$
StyleGAN-based facial image editing (Sec. 3.1.1)
Facial UV-texture extraction (Sec. 3.1.2)
UV-texture correction & completion (Sec. 3.1.3)

Figure 2. The proposed pipeline for producing normalized texture UV-map from a single in-the-wild face image, which mainly contains three modules: StyleGAN-based facial image editing, facial UV-texture extraction, and UV-texture correction & completion.

T_f for each channel in YUV color space:

$$T_a' = \frac{T_a - \mu(T_a)}{\sigma(T_a) \times \omega} \times \sigma(T_b) + \mu(T_b), \qquad (1)$$

where T_b, T_a, and T_a' are the target texture, source texture, and color-matched texture, respectively; μ and σ denote the mean and standard deviation; and ω is a hyper-parameter (set to 1.5 empirically) used to control the contrast of the output texture. Finally, the three color matched texture maps are linearly blended together using pre-defined visibility masks (see Fig. 2) to get a complete texture map T.

3.1.3 UV-Texture Correction & Completion

The obtained texture map T usually contains artifacts near eyes, mouth, and nose regions, due to imperfect 3D shape estimation. For example, eyeball and mouth interior textures* that are undesired would appear in the texture map if the eyelids and lips between the estimated 3D shape and image are not well aligned. While performing local mesh deformation according to image contents [44] could fix the artifacts in some cases, we find it would still fail for many images when processing large-scale dataset. To handle these issues, we simply extend and exclude error-prone regions and fill the missing regions with a template texture UV-map $\bar{T}$ (see Fig. 2). Specifically, we extract eyeball and mouth interior regions predicted from the face parsing result, and then unwrap them to the UV coordinate system after a dilation operation to obtain the masks of these artifacts M_{leye}, M_{reye}, and M_{mouth} on its texture UV-map. As for the nose region, we extract the nostril regions by brightness thresholding around the nose region to obtain a mask $M_{nostril}$, since nostril regions are usually dark. Then the regions in these masks are filled with textures from template $\bar{T}$ using Poisson editing [32] to get a texture map $\hat{T}$.

Finally, to obtain a complete texture map beyond facial regions (e.g., ear, neck, hair, etc.), we fill the rest regions using template $\bar{T}$, with a color matching using Eq. (1) followed by Laplacian pyramid blending [7]. The final obtained texture UV-map is denoted as $\hat{T}_{final}$.

3.1.4 FFHQ-UV Dataset

We apply the above pipeline to all the images in FFHQ dataset [20], which includes 70,000 high-quality face images with high variation in terms of age and ethnicity. Images with facial occlusions that cannot be normalized are excluded using automatic filters based on face parsing results. The final obtained UV-maps are manually inspected and filtered, leaving 54,165 high-quality UV-maps at 1024 × 1024 resolution, which we name as FFHQ-UV dataset. Tab. 1 shows the statistics of the proposed dataset compared to other UV-map datasets.

3.2. Ablation Studies

We conduct ablation studies to demonstrate the effectiveness of the three major steps (i.e., StyleGAN-based image editing, UV-texture extraction, and UV-texture correction & completion) of our dataset creation pipeline. We compare our method to three baseline variants: 1) "w/o editing", which extracts facial textures directly from in-the-wild images without StyleGAN-based image editing; 2) "w/o multi-view", which uses only a single frontal-view face image for texture extraction; and 3) "naive blending", which replaces our UV-texture correction & completion step with a naive blending to the template UV-map.

3.2.1 Qualitative Evaluation

Fig. 3 shows an example of the UV-maps obtained with different baseline methods. The baseline "w/o editing" (see Fig. 3(a)) yields a UV-map that has significant shadows and occlusions of hair. The UV-map generated by the baseline "w/o multi-view" (see Fig. 3(b)) contains smaller area of actual texture and heavily relies on template texture for completion. For the baseline method "naive blending", there

[*] In our texture UV-map, only eyelids and lips are needed since the eyeballs and mouth interiors are independent mesh models similar to the mesh topology used in HiFi3D++ [8].

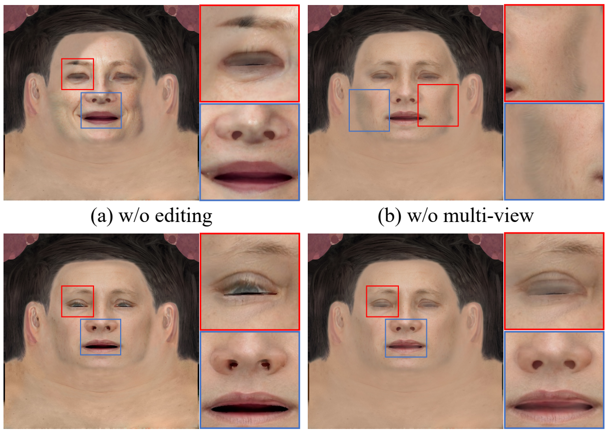
(a) w/o editing
(b) w/o multi-view
(c) naive blending
(d) Ours

Figure 3. Extracted UV-maps with different variants of our pipeline. More results in the supplementary materials.

Table 2. Quantitative evaluation on the diversity of the proposed dataset in terms of identity feature standard deviation, where all the values are divided by the value of FFHQ. $*$ indicates that ID features are extracted from rendered face images.

Datasets	FFHQ	FFHQ -Inv	FFHQ -Norm	FFHQ -UV*	Facescape*
ID std.	100.0%	91.86%	90.06%	90.01%	84.24%

Table 3. Average identity similarity score between images in FFHQ-Norm and rendered images using FFHQ-UV. The score is averaged over the whole dataset. The "negative samples" is computed between the rendered image and an index-shifted real image. Note that the rendered images are in different views (with random head poses) from real images.

Methods	negative samples	w/o multi-view	naive blending	Ours
Similarity	0.0648	0.7195	0.7712	0.7818

are obvious artifacts near eyes, mouth, and nose regions (Fig. 3(c)). In contrast, our pipeline is able to produce high-quality UV-maps with complete facial textures (Fig. 3(d)).

3.2.2 Data Diversity

We expect FFHQ-UV would inherit the data diversity of FFHQ dataset [20]. To verify this, we compute the identity vector using Arcface [12] for each face, and then calculate the standard deviation of these identity vectors to measure the identity variations of the dataset. Tab. 2 shows the identity standard deviation of the original dataset (FFHQ), the dataset inverted to the latent space (FFHQ-Inv), the normalized dataset using our StyleGAN-based facial image editing (FFHQ-Norm), the rendered face images using our facial UV-texture dataset (FFHQ-UV), where FFHQ-UV preserves the most identity variations in FFHQ (over 90%). Furthermore, our dataset has a higher identity standard deviation value compared to Facescape dataset [42], indicating that FFHQ-UV is more diverse. To analyze the identity preservation from FFHQ-Norm to FFHQ-UV, we compute the identity similarity between each image in FFHQ-Norm and the rendered image using the corresponding UV-map in FFHQ-UV with a random head pose. Tab. 3 shows that the average identity similarity score over the whole dataset achieves 0.78, which is pretty satisfactory considering that the rendered images are in different views (with random head poses) from real images. The table also shows that our result is superior to the results obtained by the baseline variants of our UV-map creation pipeline.

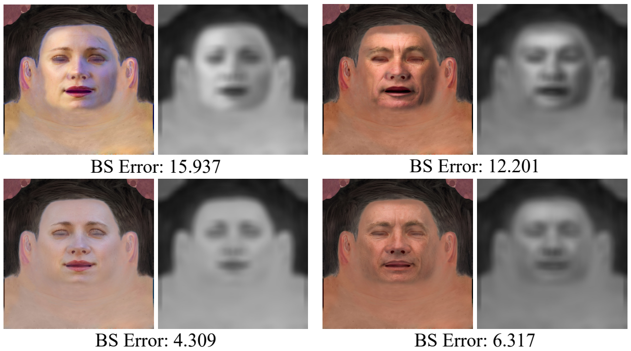

Figure 4. Examples of the computed BS Error on UV-maps and their intermediate results $\mathcal{B}_\alpha(T^Y)$. The first row are the texture UV-maps produced by the baseline method "w/o editing", and the second row are those produced by the proposed method.

Table 4. Quantitative evaluation on the illumination of the proposed UV-texture dataset in terms of BS Error, where $*$ denotes the dataset which is captured under controlled conditions.

Methods	Facescape*	w/o editing	Ours
BS Error	6.984	11.385	7.293

3.2.3 Quality of Even Illumination

Having even illumination is one important aspect for measuring the quality of a UV-map [26]. To quantitatively evaluate this aspect, we present a new metric, namely Brightness Symmetry Error (BS Error) as follows

$$BS\_Error(T) = \left\|\mathcal{B}_\alpha(T^Y) - \mathcal{F}_h(\mathcal{B}_\alpha(T^Y))\right\|_1 , \qquad (2)$$

where T^Y denotes the Y channel of T in YUV space; $\mathcal{B}_\alpha(\cdot)$ denotes the Gaussian blurring operation with the kernel size of α (set to 55 empirically); $\mathcal{F}_h(\cdot)$ denotes the horizontal flip operation. The metric is based on the observation that an unevenly illuminated texture map usually has shadows on the face which makes the brightness of UV-map asymmetrical. Fig. 4 shows two examples of the computed BS Error on UV-maps. Tab. 4 shows the average BS Error computed over the whole dataset, which demonstrates that the StyleGAN-based editing step in our pipeline effectively improves the quality in terms of more even illumination. In addition, the BS Error of our dataset is competitive with that of Facescape [42], which is captured under controlled conditions with even illumination, indicating that our dataset is indeed evenly illuminated.

Table 5. Quantitative comparison of 3D face reconstruction on the RELAY benchmark [8]. $*$ denotes the results reported from [8]. “FS” stands for Facescape dataset [42].

Methods	nose	mouth	forehead	cheek	all
3DDFA-v2* [16]	1.903	1.597	2.477	1.757	1.926
GANFit* [15]	1.928	1.812	2.402	1.329	1.868
MGCNet* [36]	1.771	1.417	2.268	1.639	1.774
Deep3D* [13]	1.719	1.368	2.015	1.528	1.657
$\mathcal{N}_{enc}$	**1.557**	1.661	1.940	1.014	1.543
PCA tex basis	1.904	1.419	1.773	0.982	1.520
w/o multi-view	1.780	1.419	1.711	0.980	1.473
w/ FS (scratch)	1.731	1.653	1.711	1.207	1.576
w/ FS (finetune)	1.570	1.576	**1.581**	1.074	1.450
Ours	1.681	**1.339**	1.631	**0.943**	**1.399**

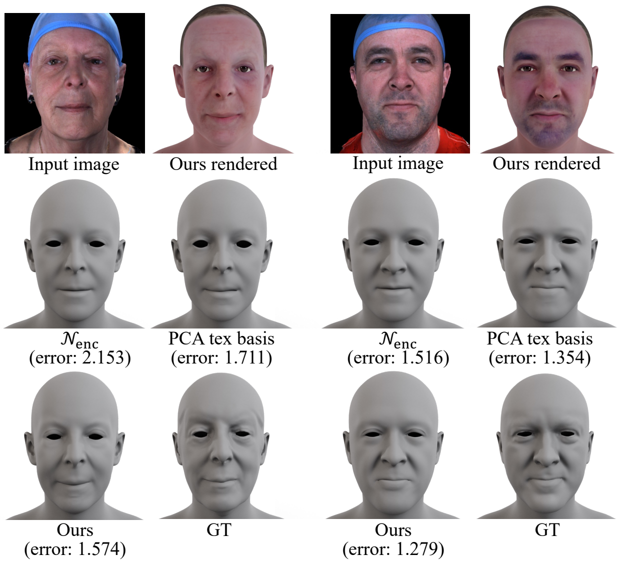
Input image
Ours rendered
Input image
Ours rendered
$\mathcal{N}_{\text{enc}}$ (error: 2.153)
PCA tex basis (error: 1.711)
$\mathcal{N}_{\text{enc}}$ (error: 1.516)
PCA tex basis (error: 1.354)
Ours (error: 1.574)
GT
Ours (error: 1.279)
GT

Figure 5. The shape reconstruction examples from REALY [8], where our method reconstructs more accurate shapes and the rendered faces well resemble the input faces.

4. 3D Face Reconstruction with FFHQ-UV

In this section, we apply the proposed FFHQ-UV dataset to the task of reconstructing a 3D face from a single image, to demonstrate that FFHQ-UV improves the reconstruction accuracy and produces higher-quality UV-maps compared to state-of-the-art methods.

4.1. GAN-Based Texture Decoder

We first train a GAN-based texture decoder on FFHQ-UV similar to GANFIT [15], using the network architecture of StyleGAN2 [21]. A texture UV-map $\tilde{T}$ is generated by

$$\tilde{T} = \mathcal{G}_{tex}(z), \tag{3}$$

where $z \in \mathcal{Z}$ denotes the latent code in the $\mathcal{Z}$ space and $\mathcal{G}_{tex}(\cdot)$ denotes the texture decoder. The goal of our UV-texture recovery during 3D face reconstruction is to find a latent code z^* that produces the best texture UV-map for an input image via the texture decoder $\mathcal{G}_{tex}(\cdot)$.

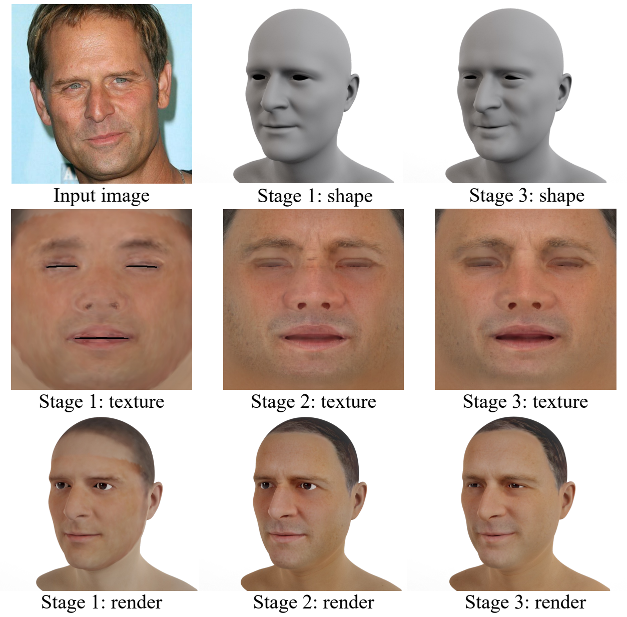

Figure 6. Intermediate reconstruction results of each stage in our algorithm. By comparing the rendered faces to the input face, it reveals that the final result better resembles the input face, especially the shape and texture around eyes, cheeks, and mouth regions.

4.2. Algorithm

Our 3D face reconstruction algorithm consists of three stages: linear 3DMM initialization, texture latent code z optimization, and joint parameter optimization. We use the recent PCA-based shape basis HiFi3D++ [8] and the differentiable renderer-based optimization framework provided by HiFi3DFace [2]. The details are as follows.

Stage 1: linear 3DMM initialization. We use Deep3D [13] trained with shape basis HiFi3D++ [8] (the same as Sec. 3.1.2) to initialize the reconstruction. Given a single input face image I^{in}, the predicted parameters are $\{p_{id}, p_{exp}, p_{tex}, p_{pose}, p_{light}\}$, where p_{id} and p_{exp} are the coefficients of identity and expression shape basis of HiFi3D++, respectively; p_{tex} is the coefficient of the linear texture basis of HiFi3D++; p_{pose} denotes the head pose parameters; p_{light} denotes the SH lighting coefficients. We use $\mathcal{N}_{enc}$ to denote the initialization predictor in this stage.

Stage 2: texture latent code z optimization. We use the parameters $\{p_{id}, p_{exp}, p_{pose}, p_{light}\}$ initialized in the last stage and fix these parameters to find a latent code $z \in \mathcal{Z}$ of the texture decoder that minimizes the following loss:

$$\mathcal{L}_{s2} = \lambda_{lpips}\mathcal{L}_{lpips} + \lambda_{pix}\mathcal{L}_{pix} + \lambda_{id}\mathcal{L}_{id} + \lambda^{z}_{reg}\mathcal{L}^{z}_{reg}, \tag{4}$$

where $\mathcal{L}_{lpips}$ is the LPIPS distance [43] between I^{in} and the rendered face I^{re}; $\mathcal{L}_{pix}$ is the per-pixel L_2 photometric error between I^{in} and I^{re} calculated on the face region predicted by a face parsing model [45]; $\mathcal{L}_{id}$ is the identity loss based on the final layer feature vector of Arcface [12]; $\mathcal{L}^{z}_{reg}$ is the regularization term for the latent code z. Sim-

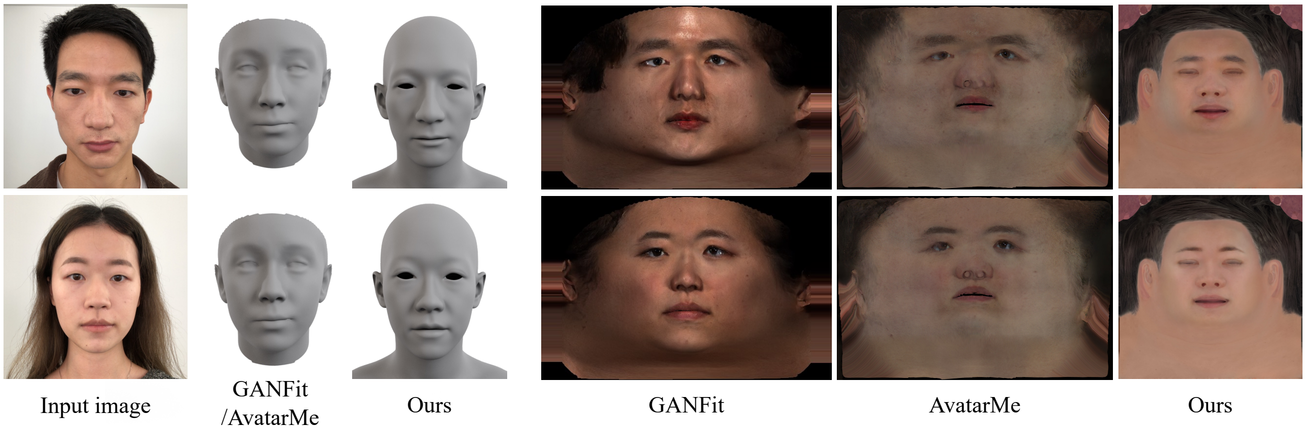

Figure 7. Visual comparison of the reconstruction results to state-of-the-art approaches GANFit [15] and AvatarMe [23]. Our reconstructed shapes are more faithful to input faces, and our recovered texture maps are more evenly illuminated and of higher quality.

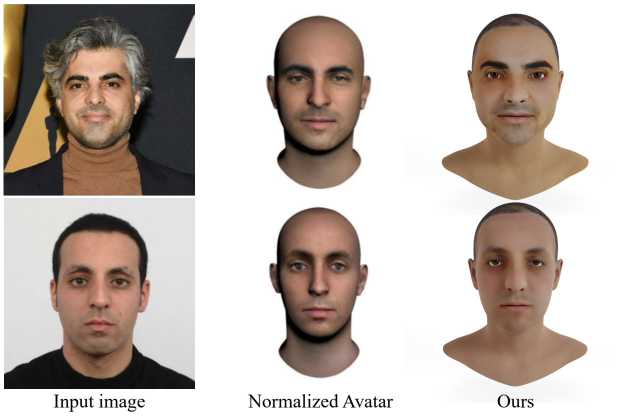

Figure 8. Visual comparison of reconstruction results to Normalized Avatar [26]. Our results better resemble the input faces.

ilar to [27], we constrain the latent code z on the hypersphere defined by $\mathcal{Z}' = \sqrt{d}S^{d-1}$ to encourage realistic texture maps, where S^{d-1} is the unit sphere in d dimensional Euclidean space.

Stage 3: joint parameter optimization. In this stage, we relax the hyperspherical constraint on the latent code z (to gain more expressive capacities) and jointly optimize all the parameters $\{z, p_{id}, p_{exp}, p_{pose}, p_{light}\}$ by minimizing the following loss function:

$$\begin{aligned}\mathcal{L}_{s3} =& \lambda_{pix}\mathcal{L}_{pix} + \lambda_{id}\mathcal{L}_{id} + \lambda^{z}_{reg}\mathcal{L}^{z}_{reg} \\ &+ \lambda_{lm}\mathcal{L}_{lm} + \lambda^{id}_{reg}\mathcal{L}^{id}_{reg} + \lambda^{exp}_{reg}\mathcal{L}^{exp}_{reg},\end{aligned} \quad (5)$$

where $\mathcal{L}_{lm}$ denotes the 2D landmark loss with a 68-points landmark detector [6]; $\mathcal{L}^{id}_{reg}$ and $\mathcal{L}^{exp}_{reg}$ are the regularization terms for the coefficients p_{id} and p_{exp}.

4.3. Evaluation

Implementation details. The GAN-based texture decoder $\mathcal{G}_{tex}(z)$ is trained using the same hyperparameters as "Config F" in StyleGAN2 [21] on 8 NVIDIA Tesla V100 GPUs, where the learning rate is set to $2e^{-3}$ and minibatch is set to 32. The loss weights $\{\lambda_{lpips}, \lambda_{pix}, \lambda_{id}, \lambda^{z}_{reg}\}$ in Eq. (4) of Stage 2 are set to $\{100, 10, 10, 0.05\}$. The loss weights $\{\lambda_{pix}, \lambda_{id}, \lambda^{z}_{reg}, \lambda_{lm}, \lambda^{id}_{reg}, \lambda^{exp}_{reg}\}$ in Eq. (5) of Stage 3 are set to $\{0.2, 1.6, 0.05, 2e^{-3}, 2e^{-4}, 1.6e^{-3}\}$. We use Adam optimizer [22] to optimize 100 steps with a learning rate of 0.1 in Stage 2, and 200 steps with a reduced learning rate of 0.01 in Stage 3. The total fitting time per image is around 60 seconds tested on an NVIDIA Tesla V100 GPU.

Shape reconstruction accuracy. We first evaluate the shape reconstruction accuracy on REALY benchmark [8], which consists of 100 face scans and performs region-wise shape alignment to compute shape estimation errors. Tab. 5 shows the results, where our method outperforms state-of-the-art single-image reconstruction approaches including MGCNet [36], Deep3D [13], 3DDFA-v2 [16], and GANFIT [15]. The table also shows the comparison to the results produced by the linear 3DMM initializer in Stage 1 ($\mathcal{N}_{enc}$), parameters optimization with linear texture basis instead of Stage 2 & 3 ("PCA tex basis"), and the texture decoder trained using UV-map dataset created without generating multi-view images ("w/o multi-view"). The results demonstrate that our texture decoder effectively improves the reconstruction accuracy. Fig. 5 shows two examples of the reconstructed meshes for visual comparison. In addition, we adopt the texture UV-maps in Facescape [42], which are carefully aligned to our topology with manual adjustment, to train a texture decoder using the same setting as ours. Tab. 5 ("w/ FS (scratch)") shows that training from scratch using Facescape does not perform well. Using FFHQ-UV as pretraining and finetuning the decoder with Facescape brings substantial improvements (see "w/ FS (finetune)"), but still decreases the result compared to ours, due to lost diversity.

Texture quality. Fig. 6 shows an example of the obtained UV-map in each stage and the corresponding rendered image. The result obtained from Stage 3 better resembles the input image, and the UV-map is more flatten and of higher quality. Fig. 7 shows two examples compared

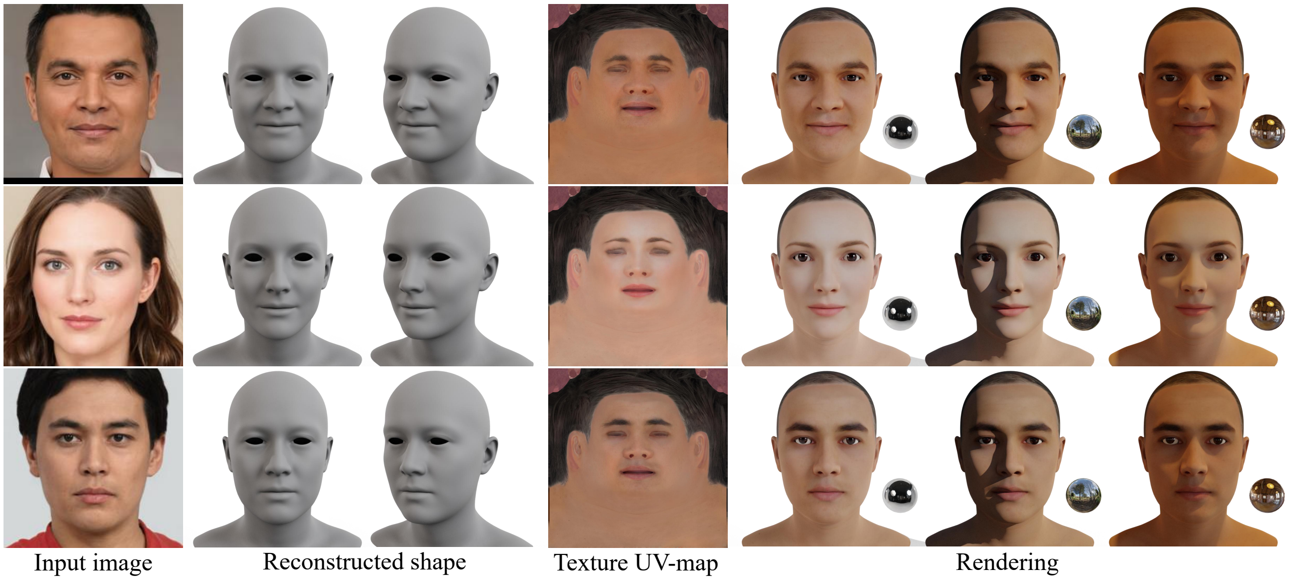

Figure 9. Examples of our reconstructed shapes, texture UV-maps, and renderings, where the produced textures are detailed and uniformly illuminated which can be rendered with different lighting conditions.

Table 6. Fitting errors on CelebA-UV-100 with different texture decoders trained on different amounts of data from FFHQ-UV. The linear basis is the PCA-based texture basis in Stage 1 of Sec. 4.2.

Tex decoder	Linear basis	5,000	20,000	54,165 (Ours)
LPIPS error	0.4581	0.2029	0.1853	**0.1487**

to GANFIT [15] and AvatarMe [23], where our obtained meshes and UV-maps are superior to other results in terms of both fidelity and asset quality. Note that there are undesired shadows and uneven shadings in the UV-maps obtained by GANFIT and AvatarMe, while our UV-maps are more evenly illuminated. Fig. 8 shows two examples of our results compared to Normalized Avatar [26], where our rendered results better resemble the input faces, thanks to the more powerful expressive texture decoder trained on our much larger dataset. In Fig. 9, we further show some examples of our reconstructed shapes, texture UV-maps, and renderings under different realistic lightings. More results are presented in the supplementary materials.

Expressive power of texture decoder. We further validate the advantage of the expressive power of the texture decoder trained on larger datasets through the following experiments. We create a small validation UV-map dataset by randomly selecting 100 images from CelebA-HQ [19] and then creating their UV-maps using our pipeline in Sec. 3.1. The validation dataset, namely CelebA-UV-100, consists of unseen data by texture decoders. We then use variants of texture decoders trained with different amounts of data to fit these UV-maps using GAN-inversion optimization [21]. Tab. 6 shows the fitting results of the final average LPIPS errors between the target UV-maps and the fitted UV-maps. The results show that the texture decoder trained on a larger dataset apparently has larger expressive capacities.

5. Conclusion and Future Work

We have introduced a new facial UV-texture dataset, namely FFHQ-UV, that contains over 50,000 high-quality facial texture UV-maps. The dataset is demonstrated to be of great diversity and high quality. The texture decoder trained on the dataset effectively improves the fidelity and quality of 3D face reconstruction. The dataset, code, and trained texture decoder will be made publicly available. We believe these open assets will largely advance the research in this direction, making 3D face reconstruction approaches more practical towards real-world applications.

Limitations and future work. The proposed dataset FFHQ-UV is derived from FFHQ dataset [20], thus might inherit the data biases of FFHQ. While one may consider further extending the dataset with other face image datasets like CelebA-HQ [19], it may not help much because our dataset creation pipeline relies on StyleGAN-based face normalization, where the resulting images are projected to the space of a StyleGAN decoder, which is still trained from the FFHQ dataset. Besides, the evaluation of the texture recovery result lacks effective metrics that can reflect the quality of a texture map. It still requires visual inspections to judge which results are better in terms of whether the illuminations are even, whether facial details are preserved, whether there exist artifacts, whether the rendered results well resemble the input images, etc. We intend to further investigate these problems in the future.

Acknowledgements. This work has been partly supported by the National Key R&D Program of China (No. 2018AAA0102001), the National Natural Science Foundation of China (Nos. U22B2049, 61922043, 61872421, 62272230), and the Fundamental Research Funds for the Central Universities (No. 30920041109).

References

[1] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 2021. 2, 3

[2] Linchao Bao, Xiangkai Lin, Yajing Chen, Haoxian Zhang, Sheng Wang, Xuefei Zhe, Di Kang, Haozhi Huang, Xinwei Jiang, Jue Wang, Dong Yu, and Zhengyou Zhang. High-fidelity 3D digital human head creation from RGB-D selfies. *ACM Trans. Graph.*, 2021. 1, 2, 3, 6

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. *ACM Trans. Graph.*, 1999. 1, 2

[4] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3D morphable model learnt from 10,000 faces. In *CVPR*, 2016. 3

[5] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *ICCV*, 2019. 2

[6] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *ICCV*, pages 1021–1030, 2017. 7

[7] Peter J Burt and Edward H Adelson. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 1983. 4

[8] Zenghao Chai, Haoxian Zhang, Jing Ren, Di Kang, Zhengzhuo Xu, Xuefei Zhe, Chun Yuan, and Linchao Bao. REALY: Rethinking the evaluation of 3D face reconstruction. In *ECCV*, 2022. 3, 4, 6, 7

[9] Wei-Chieh Chung, Jian-Kai Zhu, I-Chao Shen, Yu-Ting Wu, and Yung-Yu Chuang. Stylefaceuv: A 3d face uv map generator for view-consistent face image synthesis. pages 89–99, 2022. 2

[10] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. Synthesizing normalized faces from facial identity features. In *CVPR*, 2017. 3

[11] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. UV-GAN: Adversarial facial UV map completion for pose-invariant face recognition. In *CVPR*, 2018. 2, 3

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 5, 6

[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 *CVPRW*, 2019. 3, 6, 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.*, 2020. 1, 2

[15] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFit: Generative adversarial network fitting for high fidelity 3D face reconstruction. In *CVPR*, 2019. 1, 2, 3, 6, 7, 8

[16] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *ECCV*, pages 152–168, 2020. 6, 7

[17] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. 2020. 3

[18] Zi-Hang Jiang, Qianyi Wu, Keyu Chen, and Juyong Zhang. Disentangled representation learning for 3d face shape. In *CVPR*, pages 11957–11966, 2019. 2

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 2, 8

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2, 3, 4, 5, 8

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 2, 3, 6, 7, 8

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[23] Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou. AvatarMe: Realistically renderable 3D facial reconstruction in-the-wild. In *CVPR*, 2020. 1, 2, 3, 7, 8

[24] Gun-Hee Lee and Seong-Whan Lee. Uncertainty-aware mesh decoder for high fidelity 3D face reconstruction. In *CVPR*, 2020. 1, 2, 3

[25] Zhiqian Lin, Jiangke Lin, Lincheng Li, Yi Yuan, and Zhengxia Zou. High-quality 3d face reconstruction with affine convolutional networks. In *ACM MM*, pages 2495–2503, 2022. 2

[26] Huiwen Luo, Koki Nagano, Han-Wei Kung, Qingguo Xu, Zejian Wang, Lingyu Wei, Liwen Hu, and Hao Li. Normalized avatar synthesis using StyleGAN and perceptual refinement. In *CVPR*, 2021. 1, 2, 3, 5, 7, 8

[27] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *CVPR*, pages 2437–2445, 2020. 7

[28] Microsoft. Azure face, 2020. https://azure.microsoft.com/en-in/services/cognitive-services/face/. 3

[29] Stylianos Moschoglou, Stylianos Ploumpis, Mihalis A Nicolaou, Athanasios Papaioannou, and Stefanos Zafeiriou. 3dfacegan: adversarial nets for 3d face representation, generation, and translation. *Int. J. Comput. Vis.*, 2020. 2

[30] Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. Deep face normalization. *ACM Trans. Graph.*, 2019. 3

[31] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 3

[32] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 2003. 4

[33] Zesong Qiu, Yuwei Li, Dongming He, Qixuan Zhang, Longwen Zhang, Yinghao Zhang, Jingya Wang, Lan Xu, Xudong Wang, Yuyao Zhang, and Jingyi Yu. Sculptor: Skeleton-consistent face creation using a learned parametric generator. *ACM Trans. Graph.*, 2022. 2

[34] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*, 2021. 3

[35] Shunsuke Saito, Lingyu Wei, Liwen Hu, Koki Nagano, and Hao Li. Photorealistic facial texture inference using deep neural networks. In *CVPR*, 2017. 1, 2, 3

[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 multi-view geometry consistency. In *ECCV*, pages 53–70, 2020. 6, 7

[37] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *CVPR*, 2020. 2, 3

[38] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM Trans. Graph.*, 2021. 3

[39] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *CVPR*, 2018. 2

[40] Lizhen Wang, Zhiyuan Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. Faceverse: A fine-grained and detail-controllable 3D face morphable model from a hybrid dataset. In *CVPR*, 2022. 2

[41] Shugo Yamaguchi, Shunsuke Saito, Koki Nagano, Yajie Zhao, Weikai Chen, Kyle Olszewski, Shigeo Morishima, and Hao Li. High-fidelity facial reflectance and geometry inference from an unconstrained image. *ACM Trans. Graph.*, 2018. 1, 2, 3

[42] 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 *CVPR*, pages 601–610, 2020. 2, 3, 5, 6, 7

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6

[44] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W Jacobs. Deep single-image portrait relighting. In *ICCV*, 2019. 3, 4

[45] zllrunning. face-parsing.pytorch, 2018. https://github.com/zllrunning/face-parsing.PyTorch. 3, 6

[46] M. Zollhöfer, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Pérez, M. Stamminger, M. Nießner, and C. Theobalt. State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Comput. Graph. Forum*, 37(2), 2018. 2