OSTeC: One-Shot Texture Completion

Baris Gecer, Jiankang Deng, and Stefanos Zafeiriou
Imperial College London, Huawei CBG
{b.gecer, j.deng16, s.zafeiriou}@imperial.ac.uk
{baris.gecer, jiankangdeng, stefanos.zafeiriou1}@huawei.com

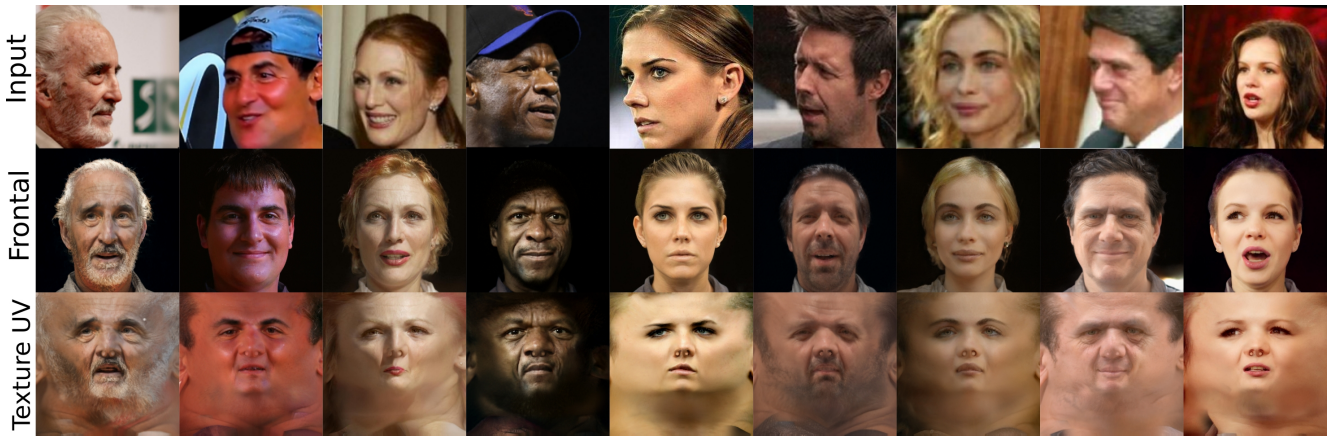


Figure 1: Face frontalization and UV texture completion by our approach. The first row is the input, the second row is the frontalization result, and the third row is the completed UV texture. The proposed method can produce photo-realistic and identity-preserved full UV textures even under extreme poses.

Abstract

The last few years have witnessed the great success of non-linear generative models in synthesizing high-quality photorealistic face images. Many recent 3D facial texture reconstruction and pose manipulation from a single image approaches still rely on large and clean face datasets to train image-to-image Generative Adversarial Networks (GANs). Yet the collection of such a large scale high-resolution 3D texture dataset is still very costly and difficult to maintain age/ethnicity balance. Moreover, regression-based approaches suffer from generalization to the in-the-wild conditions and are unable to fine-tune to a target-image. In this work, we propose an unsupervised approach for one-shot 3D facial texture completion that does not require large-scale texture datasets, but rather harnesses the knowledge stored in 2D face generators. The proposed approach rotates an input image in 3D and fill-in the unseen regions by reconstructing the rotated image in a 2D face generator, based on the visible parts. Finally, we stitch the most visible textures at different angles in the UV image-plane. Further, we frontalize the target image by projecting the completed texture into the generator. The qualitative and quantitative experiments demonstrate that the completed UV textures and frontalized images are of high quality, resembles the original identity, can be used to train a texture GAN model for 3DMM fitting and improve pose-invariant face recognition.[1]

1. Introduction

The problem of 3D face texture completion (as shown in Fig. 2) refers generally to the problem of recovering near ear-to-ear visible and non-visible colour from a single image [11] in a "canonical", deformation-free parameterization of the face surface (usually referred as UV-space). A very similar problem is that of producing arbitrary face rotations from a single image [51, 5]. Both of the above problems have important applications in many different domains of face analysis such as pose-invariant face recognition [11, 5], as well developing of 3D Morphable Model (3DMM) algorithms [6, 19] and creating complete head

[1]Project Page: https://github.com/barisgecer/OSTeC

avatars from single images [32]. That is why 3D face texture completion, as well as, producing face rotations has been very popular in the intersection of machine learning and computer vision, offering an important application domain to the advancements of machine learning in each era (from robust component analysis [38] to modern deep learning [11, 51]).

The problem of predicting the missing colour in the texture coordinated of the UV space or predicting a new view from a single image has been the application domain of many machine learning algorithms starting from simple nearest-neighbour interpolation, (*i.e.* Fig. 2c), regression techniques using linear-statistical priors (e.g., Robust Principal Component Analysis [7]) to modern deep learning regression techniques such as image-to-image translation models using conditional Generative Adversarial Networks (GANs) [27]. The problem has been modeled as fully supervised, *i.e.* the regression model was trained with pairs of missing and complete 3D facial texture [11], or recently using self-supervised methods and image rendering [51]. Nevertheless, fully-supervised or self-supervised, to the best of our knowledge, all current methods belong in the family of regression techniques.

Contrary to the above, we take a radically different line of work in this paper: We propose to re-think the 3D facial texture prediction and rotation generation as an optimisation problem and design our method as a one-shot texture completion approach. One of the key problems of regression-based approaches such as [51] is that they may lose the identity because the function they learn is quite generic. Contrary, our approach optimises, along-side many other functions, identity-related features. Our method produces visually stunning results in both 3D texture completion as well as frontalization (for some results please inspect Fig. 1). Another by-product of our method is a 3D texture model learned from in-the-wild images that, as we show, can be used for training state-of-the-art 3D face reconstruction algorithms such as GANFit [19] (which was trained with around 10K 3D faces captured in well-controlled conditions which are not released to the public).

In short, the contributions of our paper are as follows:

- We re-design the problem of 3D facial texture completion as a one-shot optimisation-based approach. We propose a well-engineered novel methodology and cost function suitable for the task.
- We capitalize on the power of 2D face generators to recover unseen part of 2D face by rotating it in 3D. So that, there would no need for 3D data collection.
- We show the effectiveness of the proposed approach in qualitative and quantitative experiments. Additionally, we apply the method to many in-the-wild images in order to train a large-scale prior of the 3D facial texture which we use to train state-of-the-art 3D face reconstruction algorithms.

2. Related Work

Face Generation, Manipulation & Rotation : In just a few years, the quality of face generations by GANs have improved incredibly [28, 29, 30]. The recently proposed StyleGANv2 [30] has shown high-quality 2D face generations up to 1024×1024 by eliminating artefacts that appear in the previous results. Many follow up works [41, 43, 3, 2, 22, 36] could successfully project real images over its latent space and perform semantic manipulation. This indicates that one can utilize StyleGAN generator as a 2D facial texture prior. In this study, we exploit this finding for image inpainting to recover the unseen part of a 2D face.

One of the commonly manipulated facial attributes is the pose, especially to a frontal view for its applications in face recognition and normalization. Unfortunately, above mentioned latent space manipulation methods are either struggling to disentangle other attributes from the latent parameters or having difficulty to project an in-the-wild image to this space. Even if it is possible to achieve excellent reconstruction by projecting to the extended latent space ($\mathbb{R}^{18\times512}$) of StyleGAN [3, 2], this enforcement exhaust its semantic meaning, therefore, become non-functional for frontalization. In fact, one can project a cat image to a StyleGAN trained on human faces by these approaches.

A large body of work addresses this problem by image-to-image translation GANs [45, 49, 26, 25, 46, 37]. Many of these approaches utilize paired datasets in a supervised setting which does not generalize well to in-the-wild settings. A recent work [51] proposed a self-supervised training approach which perturbs images by 3D rotation to generate training pairs automatically. Nevertheless, these regression-based methods suffer from generalization and fall behind the optimization-based approaches which can fine-tune for any target image.

3D Texture Completion : Modelling and synthesis of faces have been extensively studied in 3D as well [4, 15, 17, 16, 18, 40, 19]. Nevertheless, generations from these models have been far from being photorealistic. Therefore, there have been some works that proposed to complete a partially visible appearance of 2D images to a 3D appearance maps [34, 11]. The most recent one [11] trains an image-to-image translation network supervised by a set of controlled datasets, failing to generate high-quality images for in-the-wild settings. Although the proposed approach tackles the problem of texture completion, it brings a new perspective which is formulating texture completion as an optimization-based inpainting problem fortified by 2D StyleGAN and 3D geometry priors.

(a) Input image ($\mathbf{I}_0$) (b) 3D reconstruction($\mathbf{S}'$) (c) Interpolated UV map($\mathbf{T}_0$) (d) A different view ($\mathbf{I}_i$)

Figure 2: 3DMM Fitting and texture completion by nearest-neighbour interpolation. As can be seen in (d), interpolation method produces artefacts for different camera views.

Unsupervised 3D Face Model : There have been some studies to build 3d face model directly from 2D images such as [44] which learns a non-linear model from in-the-wild images and [42] learns a complete model from videos. As a side-product of this approach, we attempt to build a texture model from a set of complete texture UV-maps of 2D images and compare it to GANFit [19] model which is trained by $\sim$ 10,000 high-quality 3D textures.

3. Unsupervised UV Completion

The key insight of our work is to utilize 2D face generator networks and 3D geometry in a progressive one-shot optimization procedure for texture completion and frontalization. Basically, our approach rotates an input image in 3D and fill-in the unseen regions by reconstructing the rotated image in a 2D face generator, based on the visible parts. This 2D reconstruction is performed by an optimization in the latent space of the generator. Finally, textures acquired from these generations are collected progressively to build a coherent texture UV-map. In this section, we explain the details of our method.

3.1. 3DMM Fitting & Input Texture Acquisition

For a given 2D face image $\mathbf{I}_0$, our approach relies on a rough estimation of its dense landmarks by a 3D reconstruction method. Therefore, we begin by fitting an *off-the-shelf* 3DMM algorithm to estimate its geometry[2] $\mathbf{S} \in \mathbb{R}^{n\times 3}$ and camera parameters $\mathbf{c} = [f, r_x, r_y, r_z, t_x, t_y, t_z]$. Let us define a 2D projection operation by a pinhole camera model with the function $\mathcal{P}(\mathbf{S}, \mathbf{c}) : \mathbb{R}^{n\times 3}, \mathbb{R}^7 \rightarrow \mathbb{R}^{n\times 2}$, the geometry is then projected onto 2D image plane, *i.e.* dense landmarks, by $\mathbf{S}' = \mathcal{P}(\mathbf{S}, \mathbf{c})$.

Traditionally, high-quality 3D texture information can be stored in UV maps which assign 3D texture data into 2D planes with a universal per-pixel alignment for all textures. Each vertex of the geometry has a texture coordinate $t_{coord} \in \mathbb{R}^{n\times 2}$ in the UV image plane in which the texture information is stored. In our approach, starting from the texture available in the input image, we progressively complete the texture in the UV space.

Given a set of 2D vertex coordinates, a texture UV map $\mathbf{T} \in \mathbb{R}^{w\times h\times 3}$ and texture coordinates, one can render a textured geometry by performing rasterization with barycentric interpolation expressed as $\mathcal{R} : (\mathbb{R}^{n\times 2}, \mathbb{R}^{w\times h\times 3}, \mathbb{R}^{n\times 2}) \rightarrow \mathbb{R}^{w'\times h'\times 3}$.

In order to acquire the visible part of the texture from the input image ($\mathbf{I}_0$), we perform a similar rendering by swapping vertex coordinates with texture coordinates and the texture UV map with the input image (*i.e.*, image-to-UV rendering). In other words, the dense landmarks ($\mathbf{S}'$) from 3DMM fitting replace texture coordinates where the texture is actually the original image ($\mathbf{I}_0$). So, we unfold the input image into the UV space by giving the actual t_{coord} of our topology as the vertex coordinates to be rendered. Consequently, the rendering is performed by the following:

$$\mathbf{T}_0 = \mathcal{R}'(t_{coord}, \mathbf{I}_0, \mathbf{S}') \tag{1}$$

in which image-to-UV rendering ($\mathcal{R}'$) is essentially same operation as UV-to-image rendering ($\mathcal{R}$), however, we denote them differently to avoid confusion.

An obvious motivation of this work can be seen in the illustration of this operation in Fig. 2. After acquisition of the visible texture from the input image, we can see huge artefacts at invisible and narrow-angled parts of the geometry. Therefore, we explain how to detect and inpaint these regions by slowly building on top of the visible texture from the input image.

3.2. Re-Rendering of the Mesh

In order to fill-in the less-visible parts of the texture acquired from the original image, we rotate and render the fitted mesh by certain angles. We take the textured geometry as described in Sec. 3.1 and render it with a set of predefined camera parameters. The perspectives of these novel views are defined to maintain best visibility of every part of the face with a near-perpendicular view.

Given $\mathbf{c}_i$ ($i > 0$) as the ith novel camera parameters, we project geometry to the image plane and render texture geometry under this new perspective by the followings[3]:

$$\mathbf{S}'_i = \mathcal{P}(\mathbf{S}, \mathbf{c}_i) \tag{2}$$

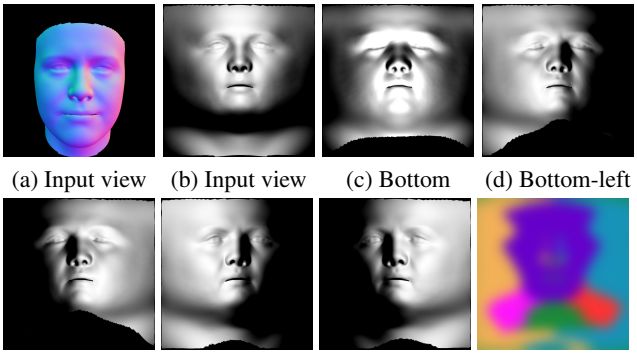
$$\mathbf{I}_i = \mathcal{R}(\mathbf{S}'_i, \overline{\mathbf{T}}_{i-1}, t_{coord}) \tag{3}$$

3.2.1 Building a Visibility Index

Each of the novel perspectives dominates certain part of the texture map in terms of clarity and visibility, *i.e.* bottom view is best for under-chin and side views are for cheeks.

[2]Please note that no texture reconstruction from the 3DMM fitting algorithm is passed to the next stages.

[3]The term $\overline{\mathbf{T}}_{i-1}$ refer to progressive texture of the previos iteration. It is explained in Sec. 3.4



(a) Input view (b) Input view (c) Bottom (d) Bottom-left

(e) Bottom-right (f) Left (g) Right (h) Visibility Ind.

Figure 3: Visibility scores are to measure optimal camera angles with respect to facial surface in UV-map.(a) $\mathbf{V}_0$. of input image (b-g) $\mathbf{V}_i$ of different views. (h) Visibility index ($\overline{\mathbf{V}}_i$): an index of optimal angles for texture acquisition.

This visibility score can be defined in terms of the angle between the normal of each triangle and its vector pointing towards the camera. Meaning that, the acquired texture would have higher resolution and less artefact with lower angles between the two vectors, *i.e.* for triangles that are facing towards the camera. For each perspective ($\mathbf{c}_i$), we extract a visibility UV map $\mathbf{V}_i$, ranging between $(-1, 1)$ where 1 indicates that the triangles around the vertex are facing towards the camera in average and -1 is facing the opposite direction. This process can be formulated by applying camera $\mathbf{c}_i$ to the geometry $\mathbf{S}$, and taking a dot product between vertex coordinates with respect to the camera and vertex normals.

$$\mathbf{V}_i = \text{diag}\left(\frac{[\mathbf{S}'_i, \mathbf{h}]}{|||[\mathbf{S}'_i, \mathbf{h}]||_2} \cdot \mathcal{N}(\mathbf{S}_i)^T\right) \tag{4}$$

where $\mathbf{h} \in \mathbb{R}^{n\times 1}$ stands for a vector of ones to make $\mathbf{S}'_i$ homogeneous. And $\mathcal{N}$ denotes the calculation the normals of the vertices. Some visibility score UV maps can be seen in Fig. 3 for different camera settings. Fig. 3h illustrates dominance map of all visibility scores, which we call *visibility index* and use it for stitching texture maps that are generated from the optimization of different views. Based on the previous equations, the binary masks of visibility index can be formulated as the following:

$$\overline{\mathbf{V}}_i = \bigcap_{i\neq j} (\mathbf{V}_i > \mathbf{V}_j) \tag{5}$$

3.3. Inpainting by Projection

The main assumption of this work is that we can utilize a generator network trained by 2D images as a prior appearance model for inpainting. Since we extracted a part of texture from the original image in Sec. 3.1, we can now use it for conditional projection to styleGAN model to generate high quality and consistent faces for the invisible part.

3.3.1 Masking

In order to separate visible and invisible regions, for each novel view, binary masks are generated from the visibility scores ($\mathbf{V}_i$). We empirically found that intersection of two masks gives the best results: 1) regions where the visibility score of the original camera is higher than a threshold ($\mathbf{V}_0 > t_1$), and 2) regions where the visibility score of the original camera perspective is higher than the target camera[4], as formulated below:

$$\mathbf{M}_i^{UV} = \left((\mathbf{V}_0 > t_1) \cap (2\mathbf{V}_0 > \mathbf{V}_i)\right) \cup \bigcup_{i>j} \overline{\mathbf{V}}_j \tag{6}$$

where $\cup_{i>j}\overline{\mathbf{V}}_j$ denotes progressive mask enlargement by the dominant regions of all previously processed camera views, which is explained in Sec. 3.4.

The mask as explained above would give a mask in UV space which is then rendered to the image space by the current camera parameters $\mathbf{c}_i$ (*i.e.* similar to Eq. 3):

$$\mathbf{M}_i = \mathcal{R}(\mathbf{S'}_i, \mathbf{M}_i^{UV}, t_{coord}) \tag{7}$$

3.3.2 Face Generation

The proposed approach requires a good quality generator that can synthesize face images from an arbitrary noise vector. Therefore, we borrow one of the *state-of-the-art* GAN network: StyleGANv2 [30] for this task. The StyleGAN or StyleGANv2 generators are particularly practical for this task as they consist of a mapping network that adds flexibility for manipulation and better projection. The mapping network ($\mathcal{G}_M : \mathbb{R}^{1\times 512} \rightarrow \mathbb{R}^{18\times 512}$) inputs a noise vector $\mathbf{z} \in \mathbb{R}^{1\times 512}$ and generates an extended latent parameters $\mathbf{W} \in \mathbb{R}^{18\times 512}$. The generator network can synthesize face images from this extended latent parameters fed into its different layers, *i.e.* $\mathcal{G} : \mathbb{R}^{18\times 512} \rightarrow \mathbb{R}^{h'\times w'\times 3}$. In this work, we optimize only based on $\mathbf{W}$ which we call latent parameters and ignore the mapping network.

During the forward pass of the optimization, we generate an image $\mathbf{G}_i^*$ by the generator network $\mathcal{G}(\mathbf{W}_i^*)$ and extract a set of features for the energy terms that we explain below. As explained in Sec. 3.3.4, the loss is backpropagated to find a good generation by updating the latent parameters $\mathbf{W}$.

3.3.3 Energy Functions

Photometric Loss : Obviously, one of the simplest form of supervision is photometric loss which encourages low-level similarity at the visible part of the image. Although

[4]Other cameras are handicapped by a factor of 2 to enlarge texture from the original image

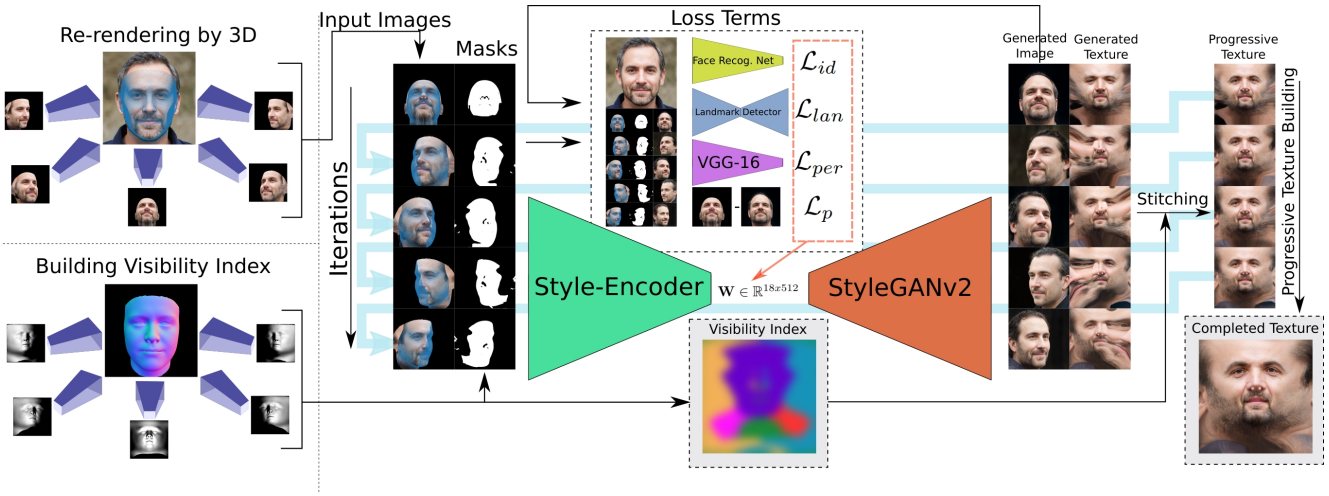


Figure 4: Overview of the method. The proposed approach iteratively optimizes the texture UV-maps for different re-rendered images with their masks. At the end of each optimization, generated images are used to acquire partial UV images by dense landmarks. Finally, the completed UV images are fed to the next iteration for progressive texture building.

simpler form of photometric loss can be defined as pixel-wise mean absolute difference between two images, we empirically find that log-cosh loss provides smoother convergence. Log-cosh loss can be defined as the following:

$$\mathcal{L}_p = \frac{1}{w' \times h' \times 3} \sum^{w' \times h' \times 3} \log\Big(\cosh\big(\mathbf{M}_i \odot (\mathbf{I}_i - \mathbf{G}_i)\big)\Big) \tag{8}$$

where $\odot$ stands for element-wise multiplication.

Identity Loss : Since photometric loss is only concerned by the low-level similarity, it struggles to achieve smooth convergence. Following [19, 10, 20, 17], we exploit identity features from a pretrained face recognition network [12] in order to capture good identity resemblance with the original image. Given a network $\mathcal{F} : \mathbb{R}^{h' \times w' \times c} \rightarrow \mathbb{R}^{512}$, we calculate the cosine distance between the identity features of the generated image and the input image as following:

$$\mathcal{L}_{id} = 1 - \frac{\mathcal{F}(\mathbf{I}_0) \cdot \mathcal{F}(\mathbf{G}_i))}{||\mathcal{F}(\mathbf{I}_0)||_2 ||\mathcal{F}(\mathbf{G}_i))||_2} \tag{9}$$

Perceptual Loss : Following the previous studies [19, 3], we exploit high-level similarity features, known as perceptual loss, to regularize convergence. We empirically choose 9th layer of a VGG-16 network that is pretrained as an ImageNet classifier as below:

$$\mathcal{L}_{per} = \sum \log\Big(\cosh\big(\mathbf{M}_i \odot (\mathrm{VGG}(\mathbf{I}_i) - \mathrm{VGG}(\mathbf{G}_i))\big)\Big) \tag{10}$$

Landmark Loss : All previous objectives are segmented by the visibility mask that covers the face partially. Therefore, invisible parts become totally relaxed, which leads to ill-aligned generations with the rendered dense landmarks ($\mathbf{S}'_i$). To this end, we propose to minimize the landmark distance between $\mathbf{I}_i$ and $\mathbf{G}_i$. As we rotate 3D mesh with a fixed topology, sparse landmark locations of the rendered images can be easily obtained from the mesh with pre-defined landmark indices ($l \in \mathbb{N}^{68}, l < n$), *i.e.* ($\mathbf{S}'_i(l)$). In order to extract landmarks of the generated image ($\mathbf{G}_i$) during the optimization [5], we employ a differentiable landmark estimator [14] defined as $\mathcal{K} : \mathbb{R}^{w' \times h' \times 3} \rightarrow \mathbb{R}^{68 \times 2}$. And the loss is expressed as:

$$\mathcal{L}_{lan} = \frac{1}{68} \sum^{68} ||\mathcal{K}(\mathbf{G}_i) - \mathbf{S}'_i(l)||_2 \tag{11}$$

3.3.4 Projection

Initialization by Regression : Following [3], we train an encoder CNN network $\mathcal{E} : \mathbb{R}^{h' \times w' \times 3} \rightarrow \mathbb{R}^{18 \times 512}$ from random styleGAN generated images ($\mathcal{G}(\mathcal{G}_M(\mathbf{z})), \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) to predict their latent parameters ($\mathbf{W} = \mathcal{G}_M(\mathbf{z})$). We initialize $\mathbf{W}$ by the regression of this network for the rendered images, *i.e.* $\mathbf{W}^* = \mathcal{E}(\mathbf{I}_i)$. Initializing the latent parameters with this regression not only accelerate the convergence but also assist optimizer to avoid local minimas.

Optimization : Given a rendered image $\mathbf{I}_i$, its respective mask $\mathbf{M}_i$, and dense landmarks $\mathbf{S}'_i$, our goal is to find the

[5] In order to flow the gradient from landmark loss, landmarks need to be computed by differentiable connections. To the best of our knowledge, this is the first attempt of such point-based supervision to a 2D image.

best latent parameters ($\mathbf{W}_i$) to reconstruct $\mathbf{I}_i$ by a pretrained StyleGANv2 generator $\mathcal{G}$. To this end, we first align $\mathbf{I}_i$, $\mathbf{M}_i$, and $\mathbf{S}'_i$ to the alignment template of StyleGANv2. And, we perform gradient descent optimization by ADAM optimizer [31] with a weighted sum of loss functions defined above:

$$\min_{\mathbf{W}_i} \mathcal{L}_{total}(\mathbf{W}_i) = \lambda_p \mathcal{L}_p + \lambda_{id} \mathcal{L}_{id} + \lambda_{per} \mathcal{L}_{per} + \lambda_{lan} \mathcal{L}_{lan} \tag{12}$$

After convergence, we synthesize a face image with the novel view $\mathbf{c}_i$ by $\mathbf{G}_i = \mathcal{G}(\mathbf{W}_i)$. Finally, we can acquire partial texture in the same way as input texture acquisition in Sec. 3.1 by $\mathbf{T}_i = \mathcal{R}'(t_{coord}, \mathbf{G}_i, \mathbf{S}'_i)$.

3.4. Progressive Texture Building for Consistency

In order to generate globally consistent texture maps, we run the optimization for each of the camera views *iteratively* to progressively improve the texture UV map. After every iteration, we blend the generated UV map ($\mathbf{T}_i$) into the current UV map at the dominated pixels ($\overline{\mathbf{V}}_i$) by that particular camera settings $\mathbf{c}_i$.

$$\overline{\mathbf{T}}_i = \overline{\mathbf{V}}_i \odot \mathbf{T}_i + (1 - \overline{\mathbf{V}}_i) \odot \overline{\mathbf{T}}_{i-1} \tag{13}$$

Blending: Texture UV-maps are stitched by alpha blending for smooth shift between different UV maps. Also, they are RGB normalized by Gaussian statistics at the intersection of visibility indices $\overline{\mathbf{V}}_0$ and $\overline{\mathbf{V}}_i$ [6].

Face Frontalization: Finally, with the complete UV map $\overline{\mathbf{T}}$, we render it once more by a frontal camera and perform a final optimization as in Eq. 12 to generate the frontal image of the input image.

4. Experiments

We implement the proposed approach in Tensorflow framework [1] and it takes around 5 minutes to UV-complete and frontalize an input image. Unfortunately, some of the preprocessing steps are CPU-intensive, therefore is a room for further efficiency. We have used geometry fitting pipeline of GANFit [19] as a preprocessing step. Other than pretrained networks for the loss function, the method itself does not require any additional training data. In the following, we illustrate some qualitative and quantitative results of our method.

4.1. Unsupervised Texture Model: UTEM

Many 3D texture reconstruction approaches rely on large-scale high-quality 3D appearance data which is costly to collect, difficult to maintain diversity (*e.g.* ethnicity, age) and often kept private due licensing issues. On the other hand, large-scale high-quality 2D face datasets are widely available [33, 28] for all. As a by-product of our approach, we build a 3D texture model by completing texture UV-maps for ∼1,500 images from CelebA-HQ [33] (as can be seen in Fig. 7 a), without any 3D data collection. After the completion, we train a GAN [28] as a GAN-based texture model and perform 3DMM fitting similar to [19]. We call this model UTEM and show some generated samples in Fig. 7 b. The 3DMM fitting results by the original GANFit [19] and the one with UTEM texture model can be seen in the last two rows of Fig. 8. The reconstructed textures show similar identity recovery and quality as GANFit textures, and it will be available for all.

[6] Normally, these two indices do not overlap, however we build $\overline{\mathbf{V}}_i$ without the handicap to find out true dominated regions. And $\overline{\mathbf{V}}_0$ is from the previous index in which it is given advantage by a factor of 2

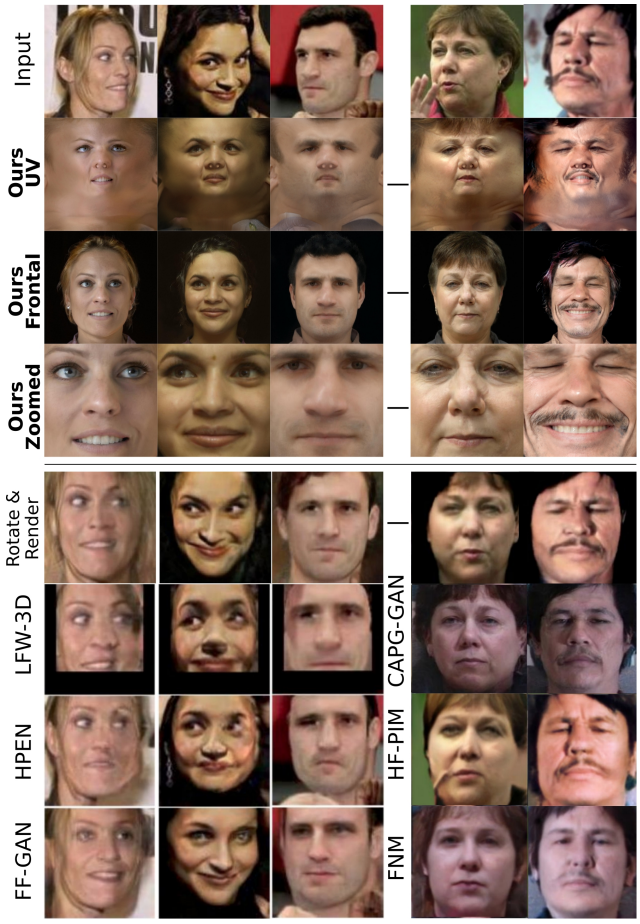


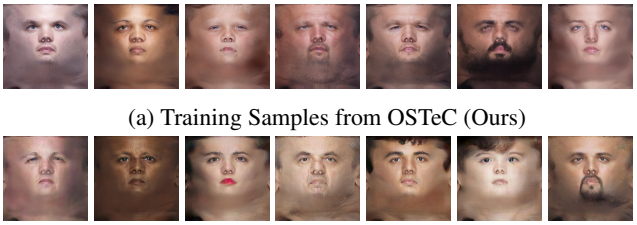
Figure 5: Comparison of our frontalization to others: Rotate&Render [51], FNM [37], CAPG-GAN [25], FF-GAN [49], HF-PIM [8], HPEN [52], LFW-3D [24]

4.2. Qualitative Results

We run our algorithm on some images in comparison with the recent state-of-the-art approaches, as shown in

(a) Input (b) $\mathcal{E}(\mathbf{I}_0)$ (c) $+\mathcal{L}_p$ (d) $+\mathcal{L}_{lan}$ (e) $+\mathcal{L}_{per}$ (f) $+\mathcal{L}_{id}$

Figure 6: Additive ablation study. '+' refers to addition of that loss term compared to the column on the left. (f) refers to all loss term used in this paper, *i.e.* **Ours**.



(a) Training Samples from OSTeC (Ours)

(b) Generated samples from Unsupervised Texture Model (Ours)

Figure 7: We build a texture model from completed textures from 2D image and train a GAN similar to GANFit [19] approach for high-quality texture modeling.

Fig. 8,5 and 1. Fig. 8 shows better quality and semantically meaningful UV-maps compared to UV-GAN [11] and GANFit [19]. Frontalization results in both Fig. 8 and 5 look superior to other previous methods in terms of identity-resemblance, artefacts and resolution.

4.3. Quantitative Results

UV Texture Completion. For the quantitative evaluation of UV texture completion, we employ the UVDB (Multi-PIE [21]) dataset released by [11]. Following [11], we skip the first 200 subjects, as there is no training, and test on the remaining 137 subjects. We employ two metrics namely peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM), which are computed between the predicted UV texture and the ground truth. In Tab. 1, the proposed method shows great priority over UV-GAN [11] and CE [35], especially for the profile faces.

Pose-invariant Face Matching. We evaluate the performance of frontalization of our work on pose-invariant face recognition in the wild. We choose the widely used dataset

Methods	Metric	0°	±30°	±60°	±90°
CE [35]	PSNR	23.03	21.93	20.27	19.63
	SSIM	0.9201	0.8920	0.8881	0.7179
UV-GAN [11]	PSNR	23.36	22.25	20.53	19.83
	SSIM	0.9241	0.8971	0.8919	0.7250
Ours	PSNR	23.95	22.54	21.04	20.44
	SSIM	0.9282	0.9018	0.8979	0.7462

Table 1: Quantitative evaluations of UV texture completion on the MultiPIE dataset [21] under view changes.

Method	Frontal-Frontal	Frontal-Profile
Human	96.24 ± 0.67	94.57 ± 1.10
DR-GAN [45]	97.84 ± 0.79	93.41 ± 1.17
DR-GAN+ [46]	98.36 ± 0.75	93.89 ± 1.39
PIM [50]	99.44±0.36	93.10 ± 1.01
HF-PIM [9]	-	94.71 ± 0.83
UVGAN [11]	98.83 ± 0.27	93.09 ± 1.72
+Profile2Frontal	-	93.55 ± 1.67
+Frontal2Profile	-	93.72 ± 1.59
+Set2set	-	94.05 ± 1.73
CASIA-R18-ArcFace	99.34 ± 0.49	93.69 ± 1.33
+Profile2Frontal	-	94.87 ± 0.96
+Frontal2Profile	-	95.68 ± 0.91
+Set2set	-	95.92 ± 0.87
MS1M-R18-ArcFace	99.68 ±0.29	96.14 ± 1.06
+Profile2Frontal	-	97.06 ± 0.74
+Frontal2Profile	-	97.43 ± 0.61
+Set2set	-	97.85 ± 0.57

Table 2: Verification accuracy(%) comparison on the CFP dataset [39].

CFP [39], which focuses on extreme pose face verification. We employ the ArcFace loss [13] to train the ResNet-18 networks [51] on CASIA-WebFace [48] and the refined version of MS1M [23, 12]. Note that the backbone of our embedding network is smaller than LightCNN-29 [47] used by HF-PIM [9] and ResNet-27 used by UV-GAN [11]. As shown in Tab. 2, synthesising frontal faces from profile faces improves the accuracy by 1.18% and 0.92% for the ArcFace models trained on CASIA and MS1M, respectively. Since face frontalization is a very challenging problem, we also synthesise profile faces from frontal faces following [11], which leads to even better results, 95.68% for the CASIA model and 97.43% for the MS1M model. In addition, we use a view interpolation of 15° to generate a set of images for each test face. Then, we use the generated set centres to conduct verification. The accuracy further improves to 95.92% for the CASIA model and 97.85% for the MS1M model, both surpassing recent state-of-art methods (*e.g.* HF-PIM [9] and UV-GAN [11]) by a large margin.

4.4. Ablation Study

We performed an ablation study to explore the contribution of each loss terms in Fig. 6. The study shows that

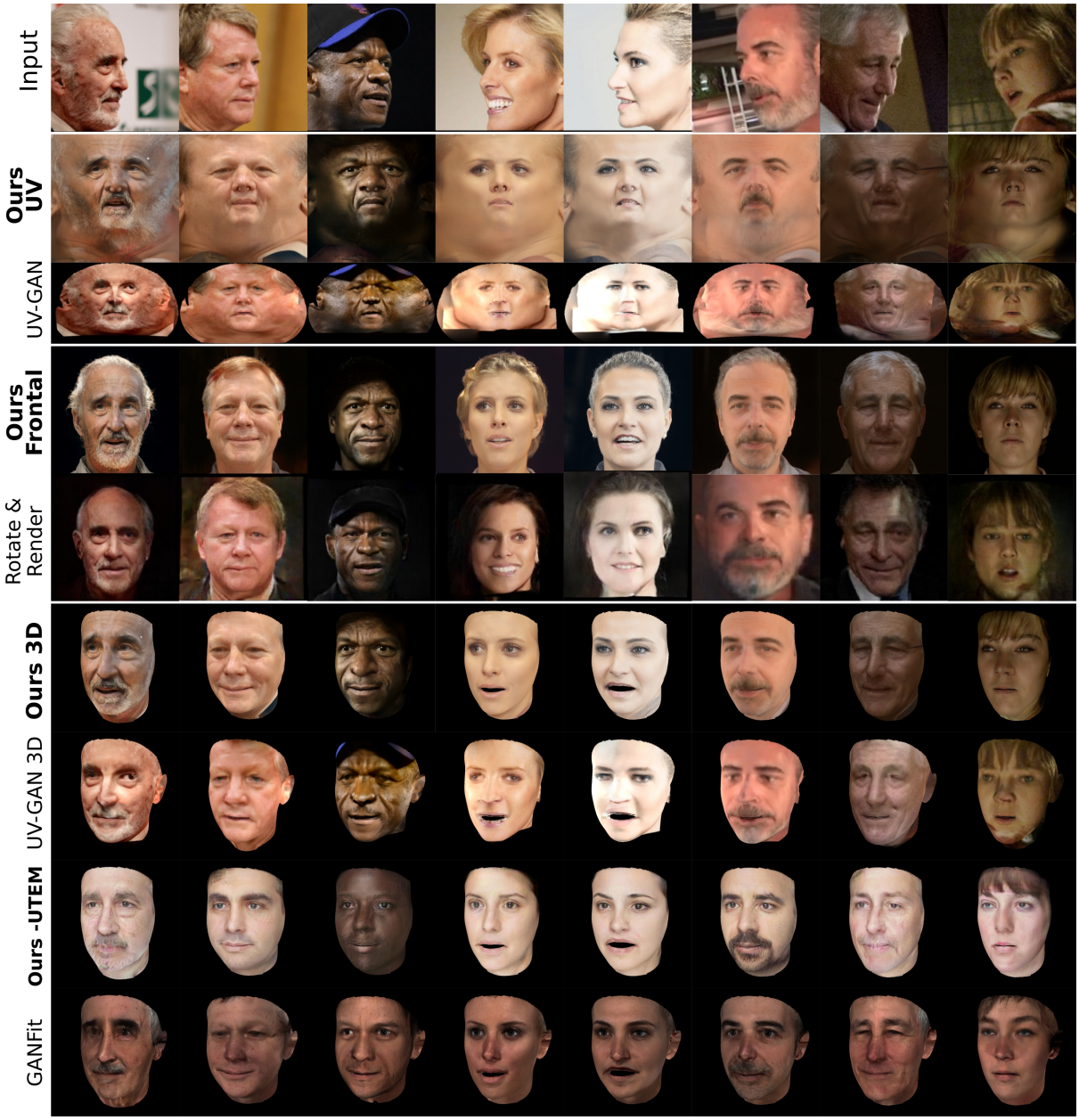


Figure 8: Qualitative results in comparison with other state-of-the-art methods (UV-GAN [11], Rotate&Render [51] and GANFit [19]). (From Top to down) First block shows input images, second block UV-completion, third block frontalization, and the fourth block texture completion/reconstruction results.

encoder $\mathcal{E}$ starts with a good initialization. $\mathcal{L}_p$ helps to match some low-level features. $\mathcal{L}_{lan}$ aligns generated images to the input geometry, *e.g.* background leakage around the neck. $\mathcal{L}_{per}$ matches mid-level features and finally $\mathcal{L}_{per}$ shows the biggest contribution by precise identity recovery.

5. Conclusion

In this paper, we propose an optimization-based one-shot 3D texture completion and frontalization approach by exploiting pretrained 2D image generation networks. Our approach can generate visually remarkable, accurate and identity-resembling complete texture maps and frontalized faces. The experiments show its superiority over other methods by accuracy and face matching at extreme poses.

Acknowledgment: The work of Stefanos Zafeiriou was funded by the EPSRC Fellowship DEFORM: Large Scale Shape Analysis of Deformable Models of Humans (EP/S010203/1).

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. 6

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to Edit the Embedded Images? *arXiv*, 2019. 2

[3] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4431–4440, 2019. 2, 5

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999. 2

[5] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3D morphable model. *TPAMI*, 25(9):1063–1074, 2003. 1

[6] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3D face morphable models "In-the-Wild". In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, pages 5464–5473, 2017. 1

[7] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011. 2

[8] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. In *NeurIPS*, 2018. 6

[9] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Towards high fidelity face frontalization in the wild. *IJCV*, 2019. 7

[10] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T. Freeman. Synthesizing normalized faces from facial identity features. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, pages 3386–3395, 2017. 5

[11] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. UV-GAN: Adversarial facial UV map completion for pose-invariant face recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 7093–7102, 2018. 1, 2, 7, 8

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:4685–4694, 2019. 5, 7

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 7

[14] Jiankang Deng, Yuxiang Zhou, Shiyang Cheng, and Stefanos Zaferiou. Cascade multi-view hourglass model for robust 3D face alignment. In *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, pages 399–403. IEEE, 2018. 5

[15] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3D Morphable Face Models – Past, Present, and Future. *ACM Transactions on Graphics*, 39(5):157:1–157:38, June 2020. 2

[16] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, and Jamie Shotton. High Resolution Zero-Shot Domain Adaptation of Synthetically Rendered Face Images. *arXiv:2006.15031 [cs]*, June 2020. 2

[17] Baris Gecer, Binod Bhattarai, Josef Kittler, and Tae-Kyun Kim. Semi-supervised Adversarial Learning to Generate Photorealistic Face Images of New Identities from 3D Morphable Model. In *European Conference on Computer Vision (ECCV)*, volume 11215, pages 230–248. Springer International Publishing, 2018. 2, 5

[18] Baris Gecer, Alexander Lattas, Stylianos Ploumpis, Jiankang Deng, Athanasios Papaioannou, Stylianos Moschoglou, and Stefanos Zafeiriou. Synthesizing Coupled 3D Face Modalities by Trunk-Branch Generative Adversarial Networks. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[19] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: Generative Adversarial Network Fitting for High Fidelity 3D Face Reconstruction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1164, Long Beach, CA, USA, June 2019. IEEE. 1, 2, 3, 5, 6, 7, 8

[20] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3D morphable model regression. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018. 5

[21] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*. 7

[22] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image Processing Using Multi-Code GAN Prior. *arXiv:1912.07116 [cs]*, Mar. 2020. 2

[23] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016. 7

[24] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *Proceedings of the IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition, volume 07-12-June-2015, pages 4295–4304, 2015. 6

[25] Yibo Hu, Xiang Wu, Bing Yu, Ran He, and Zhenan Sun. Pose-guided photorealistic face rotation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 8398–8406, 2018. 2, 6

[26] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2439–2448, 2017. 2

[27] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:5967–5976, 2017. 2

[28] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018. 2, 6

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:4396–4405, 2019. 2

[30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. *arXiv:1912.04958 [cs, eess, stat]*, Mar. 2020. 2, 4

[31] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. 6

[32] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 760–769, 2020. 2

[33] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6

[34] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3911–3919, 2017. 2

[35] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 7

[36] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a face: Towards arbitrary high fidelity face manipulation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10033–10042, 2019. 2

[37] Yichen Qian, Weihong Deng, and Jiani Hu. Unsupervised face normalization with extreme pose and expression in the wild. In *CVPR*, 2019. 2, 6

[38] Christos Sagonas, Yannis Panagakis, Stefanos Zafeiriou, and Maja Pantic. Robust statistical frontalization of human and animal faces. *International journal of computer vision*, 122(2):270–291, 2017. 2

[39] Soumyadip Sengupta, Jun Cheng Chen, Carlos Castillo, Vishal M. Patel, Rama Chellappa, and David W. Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016. 7

[40] Gil Shamai, Ron Slossberg, and Ron Kimmel. Synthesizing facial photometries and corresponding geometries using generative adversarial networks. *ACM Transactions on Multimedia Computing Communications and Applications*, 15(3s), Oct. 2019. 2

[41] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the Latent Space of GANs for Semantic Face Editing. *arXiv*, 2019. 2

[42] Ayush Tewari, Florian Bernard, Pablo Garrido, Gaurav Bharaj, Mohamed Elgharib, Hans-Peter Seidel, Patrick Perez, Michael Zollhofer, and Christian Theobalt. FML: Face Model Learning From Videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10804–10814, Long Beach, CA, USA, June 2019. IEEE. 3

[43] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. StyleRig: Rigging StyleGAN for 3D Control over Portrait Images. *arXiv:2004.00121 [cs]*, Mar. 2020. 2

[44] Luan Tran and Xiaoming Liu. On learning 3D face morphable model from in-the-wild images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 3

[45] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, pages 1283–1292, 2017. 2, 7

[46] Luan Tran, Xi Yin, and Xiaoming Liu. Representation learning by rotating your faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(12):3007–3021, 2019. 2, 7

[47] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 2018. 7

[48] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 7

[49] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, pages 4010–4019, 2017. 2, 6

[50] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen,

Junliang Xing, et al. Towards pose invariant face recognition in the wild. In *CVPR*, 2018. 7

[51] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 6, 7, 8

[52] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z. Li. High-fidelity Pose and Expression Normalization for face recognition in the wild. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 787–796, 2015. 6