StyleFaceUV: A 3D Face UV Map Generator for View-Consistent Face Image Synthesis

Wei-Chieh Chung*[1]
r08922002@ntu.edu.tw
Jian-Kai Zhu*[1]
r10922033@csie.ntu.edu.tw
I-Chao Shen[2]
ichao.shen@ui.is.s.u-tokyo.ac.jp
Yu-Ting Wu[3]
yutingwu@mail.ntpu.edu.tw
Yung-Yu Chuang[1]
cyy@csie.ntu.edu.tw

[1] National Taiwan University, Taipei, Taiwan
[2] The University of Tokyo, Tokyo, Japan
[3] National Taipei University, New Taipei City, Taiwan

Abstract

Recent deep image generation models, such as StyleGAN2, face challenges to produce high-quality 2D face images with multi-view consistency. We address this issue by proposing an approach for generating detailed 3D faces using a pre-trained StyleGAN2 model. Our method estimates the 3D Morphable Model (3DMM) coefficients directly from the StyleGAN2's stylecode. To add more details to the produced 3D face models, we train a generator to produce two UV maps: a diffuse map to give the model a more faithful appearance and a generalized displacement map to add geometric details to the model. To achieve multi-view consistency, we also add a symmetric view image to recover information regarding the invisible side of a single image. The generated detailed 3D face models allow for consistent changes in viewing angles, expressions, and lighting conditions. Experimental results indicate that our method outperforms previous approaches both qualitatively and quantitatively.

1 Introduction

Generative adversarial networks (GANs) have made significant advances in synthesizing high-quality realistic face images. Furthermore, some GANs offer flexibility in manipulating face attributes in images, since their latent spaces are highly correlated with various styles, including pose, gender, and age. However, despite the progress made, it remains challenging to maintain multi-view consistency of the face while manipulating it.

This paper attempts to convert StyleGAN2 [13] into a 3D face generator in order to provide direct and consistent control over faces across multiple views. We use the popular 3D Morphable Model as the base model. In the first step, we train a model for estimating 3DMM coefficients directly from a StyleGAN2 stylecode. This will allow our model to be

* These authors contributed equally to this work.

compatible with StyleGAN2 and reduce training efforts later. However, the 3DMM model cannot provide specific facial details. Therefore, the reconstructed image differs from the StyleGAN2 image in some respects. In order to improve the quality of the reconstructed image, we train a model that takes the stylecode as input and generates two UV maps that augment the details in order to make the reconstructed image resemble the detail in the StyleGAN2 image. The two maps are the diffuse map, which represents the face's appearance, and the generalized displacement map, which provides geometric details. Moreover, since a single image can only cover one side of the face and cannot guarantee the face quality on the opposite side, our method generates an opposite view image of the face using StyleGAN2 to add additional constraints. By integrating information from both views, a consistent reconstruction of the face can be achieved.

The result is a 3D face generator that provides detailed textures and shapes. In addition, it is compatible with StyleGAN2. In other words, it generates a 3D model with the same identity as the image created by StyleGAN2 using the same stylecode. Our experiments demonstrate that our method outperforms other methods and is more controllable. Moreover, since our method generates 3D face models, it allows for applications requiring a number of 3D faces with different identities, such as passers-by in video games or visual effects.

2 Related work

Disentanglement of 2D GANs. GANs have demonstrated their ability to generate images of high quality. Despite their ability to produce high-quality images, many previous studies have focused on providing semantic control over images by disentangling the latent space of GANs. Jahanian *et al*. [10] found both linear and non-linear optimal paths in the latent space of GANs by minimizing an objective function. By identifying the principal components of the latent space, GANSpace [8] identifies important factors of variation. Voynov *et al*. [26] learned the meaningful direction by jointly training a set of orthogonal directions and a model to distinguish the corresponding image transformations. InterFaceGAN [21, 22] obtains the semantically meaningful directions in latent space using the normal vector of the latent codes classification hyperplane.

Parameterized face models. Parameterized 3D face models have become an active research field since Blanz and Vetter proposed the 3D Morphable Model [2] based on the parameterization of the scanned 3D face data. Tran *et al*. [25] utilized two decoders to map from the shape and texture parameters to the vertex positions and the texture coordinates to achieve better representations than the linear models. In order to achieve accurate and fast face reconstruction, Deng *et al*. [6] leveraged image-level and perception-level losses. Tewari *et al*. [24] proposed to control the synthesized face via rig-like controllers. Abdal *et al*. [1] proposed StyleFlow, which uses conditional continuous normalizing flow (CNF) to create an invertible mapping from a latent code and the tunable face attribute variables in the StyleGAN1/2 latent space.

StyleGANs for 3D face synthesis. StyleGANs [12] have been widely studied for their interpretability and disentanglement. In addition, their explainable latent spaces and high-quality generation also make them valuable tools for 3D synthesis. Zhang *et al*. [28] exploited StyleGAN as a multi-view image generator with pose-related latent codes to train the inverse graphic networks, but their work requires manual annotations for rough angles of view.

Pan *et al*. [13] first used a neural renderer to generate pseudo samples with various poses and lightings, then used these samples to guide the images generated by GANs toward the corresponding sampled poses and lighting conditions. Shi *et al*. [23] proposed LiftedGAN, a framework that maps a latent code of StyleGAN2 to various maps as representations of shape and appearance. However, using the depth map as the shape representation causes inevitable image distortion as the change of view angle grows. StyleUV [15] retrains the architecture of StyleGAN2 to produce texture maps. However, unlike us, they did not include a shape representation while generating faces. It does not obey the original latent space, either. Luo *et al*.'s method [16] synthesizes texture and shape maps by retraining StyleGAN2 but requires ground truth 3D geometries and albedo textures. Chan *et al*. [3, 4] used implicit radiance field as 3D-aware guidance for synthesizing realistic 3D-aware faces. However, their method lacks the controllability of the synthesized faces.

3 Method

In this section, we first give a brief introduction to our 3D face model and StyleGAN2. We then describe the individual modules and the overall training framework. Lastly, we discuss the loss functions.

3.1 Preliminary: 3D face model

In this paper, we employ the classic Basel Face Model (BFM) [19] and the expression basis of Guo *et al*. [7] as our 3D Morphable Model (3DMM). We use 177 values as our 3DMM coefficients $\phi = (P_{id}, P_{exp}, \alpha, \gamma, \delta)$, where $P_{id} \in \mathbb{R}^{80}$ and $P_{exp} \in \mathbb{R}^{64}$ are the coefficients of 3DMM basis; $\alpha \in \mathbb{R}^3$ is the rotation of the face; $\gamma \in \mathbb{R}^{27}$ contains the illumination parameters; and $\delta \in \mathbb{R}^3$ represents the translation. The shape S_{3DMM} is expressed as follows:

$$S_{3DMM}(P_{id}, P_{exp}) = \overline{S} + P_{id}B_{id} + P_{exp}B_{exp}, \tag{1}$$

where $\overline{S}$ denotes the mean shape; and B_{id} and B_{exp} are the 3DMM basis.

3.2 Stylecode to 3DMM and multi-view coefficients

We describe in this section how we construct the modules that predict the 3DMM coefficients ϕ and the multi-view coefficients (d, s_{yaw}) given the stylecode w.

3.2.1 Stylecode to 3DMM coefficients

Our goal is to determine the mapping between the stylecode w and the 3DMM coefficients ϕ. First, we randomly select 36,000 stylecodes and feed them to StyleGAN2 to generate corresponding face images. For the i-th image generated by w_i, we utilize a 3DMM coefficient fitting tool[1] for obtaining its 3DMM coefficients ϕ_i. Using 36,000 pairs of (w_i, ϕ_i), we train a multi-layer perceptron (MLP) network, $\phi = \Phi_{3D}(w)$, to predict the 3DMM coefficients ϕ from the input stylecode w. $\Phi_{3D}(w)$ comprises two fully-connected multi-layer perceptrons, both containing three hidden layers with $9 \cdot 512$, $6 \cdot 512$, $3 \cdot 512$ hidden units from the first

[1] The code can be obtained from https://github.com/ascust/3DMM-Fitting-Pytorch.

Figure 1: Results of the stylecode-to-3D-Coefficient mapping. Despite resembling the StyleGAN2 images in general, the generated 3D faces lack details such as eyeglasses or wrinkles.

hidden layer to the last one. The first MLP uses Tanh as its inter-connected neurons' activation function, while the second MLP uses ELU with $\alpha = 1.0$ instead. Concatenating the mapping results of these two parts forms the full 3DMM coefficients.

To acquire pairs of stylecodes and their corresponding 3DMM coefficients for training the network, we first synthesize face images corresponding to their stylecodes using the original StyleGAN2. We then employ an open source implementation[2] to fit the 3DMM coefficients from the face images. The network is trained by minimizing the L1 distance between the predicted 3DMM coefficients and the fitted 3DMM coefficients. We use Adam optimizer with learning rate decay from 1×10^{-5} to 1×10^{-8}. The training epochs and batch sizes are 22 and 16, respectively. Figure 1 provides some results. The mapping module faithfully predicts the 3DMM coefficients that match the corresponding StyleGAN2 image generated using a specific stylecode.

3.2.2 Stylecode to multi-view coefficients

Inspired by InterFaceGAN [21, 22], we aim to find a direction d in the StyleGAN2 latent space so that moving a stylecode along d generates the image of a face from different viewing angles. To determine d, we randomly sample 200K stylecodes. Then, we estimate their yaw angles using Φ_{3D}. We pick the stylecodes with the highest 2% yaw angles as one class, and the stylecodes with the lowest 2% yaw angles as the other class. We solve this problem as a binary classification task using linear Support Vector Machine (SVM) [5]. And we obtain d for yaw angles by taking the normal to the decision hyperplane. With the obtained d, we can rotate a face by $w' = w + s_{\text{yaw}} * d$, where s_{yaw} denotes the step size we take along the yaw axis. For generating the face image at the yaw angle α using w, we train a "stylecode-to-yaw-angle" mapping to determine the corresponding step size: $s_{\text{yaw}} = \Phi_{yaw}(w, \alpha)$. We can then obtain the stylecode for the same person with the desired angle α using:

$$w' = w + \Phi_{yaw}(w, \alpha) * d \tag{2}$$

3.3 StyleGAN2 as a 3D generator

As shown in Figure 1, we can generate a textured 3D face model corresponding to the StyleGAN2-generated image using the same stylecode through Φ_{3D}. However, the resulting faces directly generated from the mapped 3D coefficient lack many details and components compared to those generated by StyleGAN2. As an example, the eyeglasses are absent, as are the wrinkles and nasal lines (Figure 1). To recover details in an image generated by

[2] https://github.com/ascust/3DMM-Fitting-Pytorch

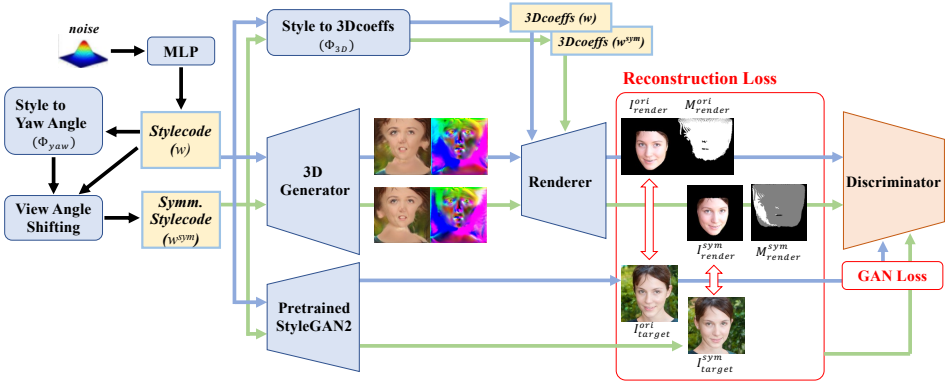

Figure 2: The training procedure of our 3D face UV map generator.

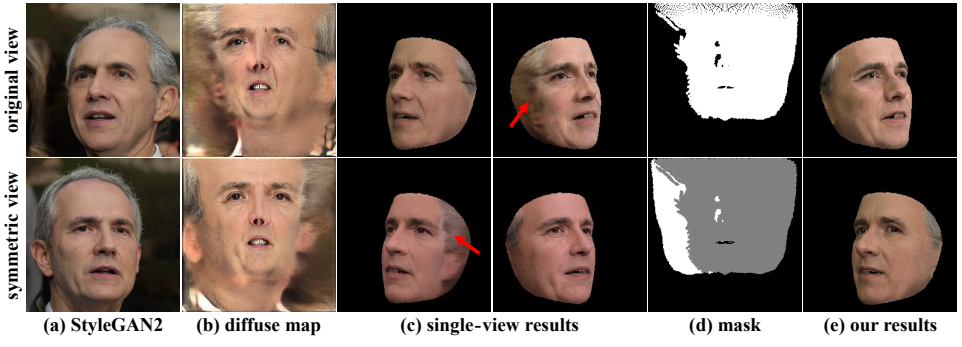

Figure 3: When only one view is used to generate a face model, (c) the result will usually include artifacts. (e) Our method generates a high-quality model of the entire face by integrating of information from the original and symmetric views.

StyleGAN2, we augment the 3DMM model with a diffuse map and a generalized displacement map; both are UV maps. The former captures the necessary color variations, whereas the latter captures geometric details of the face image generated by StyleGAN2.

Figure 2 illustrates the training process. To obtain the diffuse map and the generalized displacement map, we train a 3D generator. The 3D generator shares the architecture with the generator of StyleGAN2, but outputs the two required maps. During training, the stylecode w is fed to a pre-trained Φ_{3D} (Section 3.2.1) as well as a pre-trained StyleGAN2 module in order to obtain the corresponding 3DMM parameters ϕ_w and the target images I_{target}. With the base face model, the 3DMM coefficients, and the two UV maps, we render the output image I_{render} using a differentiable renderer. We train the network with a reconstruction loss between I_{target} and I_{render} in order to encourage the diffuse map and displacement map to capture the image details generated by StyleGAN2. To improve the fidelity of the generated maps, we include a pre-trained 2D discriminator from StyleGAN2 as well as a GAN loss.

As shown in Figure 3(c), using a single image to reconstruct the 3D model usually produces artifacts, especially on the other side of the face. In order to achieve multi-view consistency and to ensure the quality of the entire face, our method automatically generates a symmetric face image using StyleGAN2 and uses it as an additional constraint when train-

ing the 3D generator. As introduced in Section 3.2.2, given a stylecode w, we first obtain its corresponding step size s_{yaw} using the "stylecode-to-yaw-angle" mapping. Following this, we apply Eq. 2 to obtain a symmetric stylecode w^{sym} that corresponds to the original face under a symmetric viewing angle (abbreviated as "view angle shifting"). We feed the symmetric stylecode w^{sym} into the 3D generator and compute the reconstruction loss between the rendered image $I^{\text{sym}}_{\text{render}}$ and the symmetric StyleGAN2 image I^{sym}_{target} as well. Because the face rendered in the original perspective and its symmetric view provides reliable information in different regions, our approach utilizes the rendering gradients from the differentiable renderer to obtain a pixel-wise mask that weighs the effects of each view. Figure 3(d) gives examples of the masks. For both masks, the texels' values are set to 1.0 if they are accessed by the renderer and to 0.0 otherwise. We override the values to 0.5 empirically for the symmetric view if the texels are also assessed by the original view. Through the integration of information from the two pairs of views and masks, our method is able to achieve geometric and texture consistency across the entire face. A detailed description of the mask generation can be found in the supplemental material.

3.4 Loss functions

We aim to create high-quality 3D face models with latent codes that are compatible with StyleGAN2. As a result, we design our loss functions on top of the original StyleGAN2 loss function with reconstruction loss. In order to synthesize the view-consistent diffuse map and displacement map of the face, we also consider multi-view consistency loss to utilize information from both views while training our generator and discriminator.

3.4.1 Reconstruction loss with weighted masks

Our reconstruction loss consists of photometric and perceptual losses:

$$\mathcal{L}_{rec}(I_{\text{target}}, I_{\text{render}}, M_{\text{render}}) = \mathcal{L}_{photo}(I_{\text{target}}, I_{\text{render}}, M_{\text{render}}) \tag{3}$$

$$+ \lambda_{percept}\mathcal{L}_{percept}(I_{\text{target}}, I_{\text{render}}, M_{\text{render}}), \tag{4}$$

where I_{target}, I_{render}, and M_{render} denote the target image, the rendered image, and the weighted mask of rendered face. $\lambda_{percept}$ denotes the weight to combine these two losses. The photometric loss encourages the generator to generate a proper diffuse map and displacement map so that the rendered image resembles the target image:

$$\mathcal{L}_{photo}(I_{\text{target}}, I_{\text{render}}, M_{\text{render}}) = \frac{\|I_{\text{render}} - I_{\text{target}}\|_2^2 \odot M_{\text{render}}}{|M_{\text{render}}|}, \tag{5}$$

where $|M_{\text{render}}|$ denotes the number of valid pixels inside the M_{render}. We compute the perceptual loss [11, 27] using a pre-trained VGG-16 network.

3.4.2 GAN loss

Our GAN loss follows StyleGAN2's configuration, which also uses a non-saturating loss with R1 regularization [17]. The generator loss comes with a non-saturating loss without the regularization of path length. By employing the non-saturating loss, we are able to provide the generator with valid gradients at the early stage of training, which allows us to train our

model more easily in the beginning. With the reconstruction loss ($\mathcal{L}_{rec}$), our full generator loss is

$$\mathcal{L}_G(I_{\text{target}}, I_{\text{render}}, M_{\text{render}}) = -E[\log D(I_{\text{render}})] + \lambda_{rec}\mathcal{L}_{rec}(I_{\text{target}}, I_{\text{render}}, M_{\text{render}}), \quad (6)$$

where D is the discriminator. Our discriminator loss consists of logistic loss term and R1 regularization term:

$$\mathcal{L}_D(I_{\text{target}}, I_{\text{render}}) = -E[\log D(I_{\text{target}})] - E[1 - \log D(I_{render})] + \frac{\gamma}{2}E_x[\|\nabla D_\psi(x)\|_2^2], \quad (7)$$

where x is the sampled image from StyleGAN2 output; γ is the hyperparameter; and ψ is the weight of the discriminator.

3.4.3 Multi-view consistency loss

To enforce multi-view consistency, we include the multi-view generator loss on both the original and symmetric views with a weight λ_{mv} to combine the information from both views:

$$\mathcal{L}_{\text{G}}^{\text{mv}} = \mathcal{L}_G(I_{\text{target}}^{\text{ori}}, I_{\text{render}}^{\text{ori}}, M_{\text{render}}^{\text{ori}}) + \lambda_{mv}\mathcal{L}_G(I_{\text{target}}^{\text{sym}}, I_{\text{render}}^{\text{sym}}, M_{\text{render}}^{\text{sym}}), \quad (8)$$

where ($I_{\text{target}}^{\text{ori}}$,$I_{\text{target}}^{\text{sym}}$) denotes the target original and symmetric images; ($I_{\text{render}}^{\text{ori}}$,$I_{\text{render}}^{\text{sym}}$) denotes the rendered original and symmetric images; and ($M_{\text{render}}^{\text{ori}}$,$M_{\text{render}}^{\text{sym}}$) denotes the weighted masks of original and symmetric rendered faces. The multi-view discriminator loss is:

$$\mathcal{L}_D^{\text{mv}} = \mathcal{L}_D(I_{\text{target}}^{\text{ori}}, I_{\text{render}}^{\text{ori}}) + \lambda_{mv}\mathcal{L}_D(I_{\text{target}}^{\text{sym}}, I_{\text{render}}^{\text{sym}}). \quad (9)$$

4 Experiments

4.1 Implementation detail

We use the StyleGAN2 [13] pre-trained on the FFHQ dataset [12] with Pytorch reimplementation[3] as our backbone. The training images along with their corresponding stylecodes are all generated by the pre-trained StyleGAN2, the resolution is 256×256, and the number of images is $35,820$. We use Adam [14] with learning rate set to 0.001 as the optimizers of both generator and discriminator, the number of training epochs and batch size are set to 15 and 12 respectively. The training hyperparameters λ_{rec}, $\lambda_{percept}$, λ_{mv}, γ are set to 10.0, 0.2, 0.75, 10.0, respectively. The ratio of training iterations between the generator and discriminator is set at $5:1$. The differential rendering pipeline is implemented using Pytorch3D [20].

4.2 Qualitative results

Figure 4 compares our method with other controllable and 3D-aware face synthesis methods based on StyleGAN2, including LiftedGAN [23], InterfaceGAN [22], StyleFlow [1], and EG3D [4]. We exclude StyleRig [24] from the comparison because its implementation is not available. Among all the compared methods, only our method and InterfaceGAN are compatible with StyleGAN2 and share the same latent space. As a result, it is not possible to compare the same person with other methods. We encourage readers to evaluate each

[3] Source code is available at https://github.com/rosinality/stylegan2-pytorch.

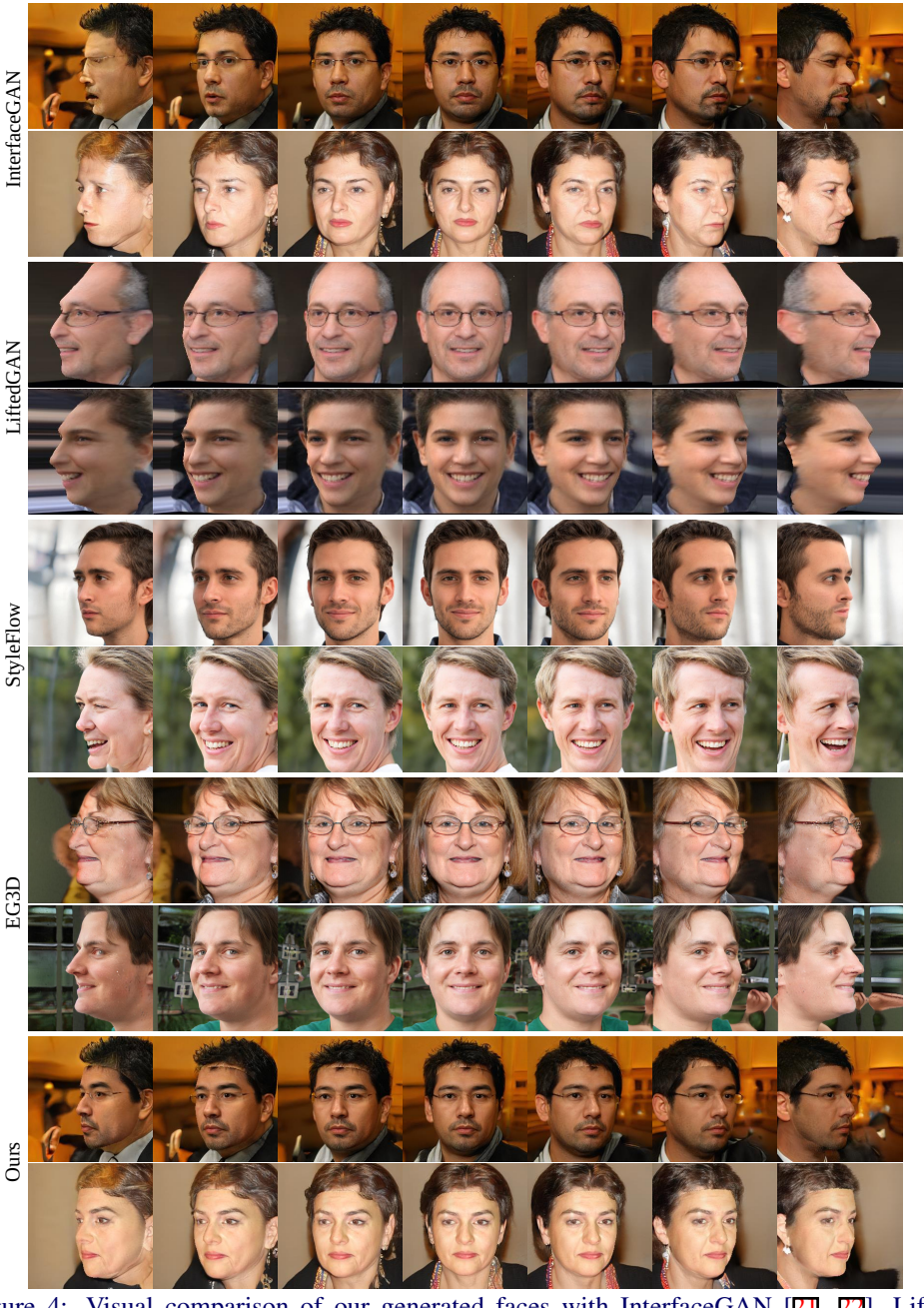

Figure 4: Visual comparison of our generated faces with InterfaceGAN [21, 22], LiftedGAN [23], StyleFlow [1], EG3D [4] in different yaw angles. From left to right, the yaw degrees are -60°, -30°, -15°, 0°, 15°, 30°, 60°. Please note that LiftedGAN, StyleFlow and EG3D use different latent spaces from the original StyleGAN2. Therefore, we are unable to compare the results using the same identity. Nevertheless, it is clear that LiftedGAN, StyleFlow, and EG3D all generate distorted results at large yaw angles. Our method generates more consistent results than others when varying the yaw angle.

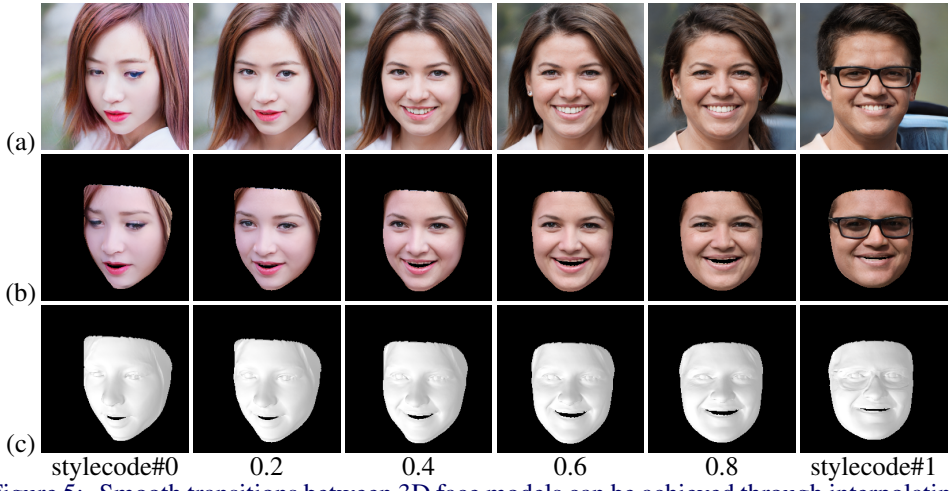

Figure 5: Smooth transitions between 3D face models can be achieved through interpolating style codes. (a) StyleGAN2 images. (b) Rendered images. (c) 3D shapes.

generated face individually based on the quality and preservation of identity under large angle changes. Our method, in contrast to other methods that directly synthesize 2D images, creates a 3D face that is accompanied by two UV maps and 3DMM parameters. To compare our results with other methods, we render the 3D face at the specified view and paste the rendered face into the StyleGAN2 image. Consequently, some of our results exhibit seams at the face boundaries. They could be removed using advanced image compositing techniques. InterFaceGAN[21, 22] uses the Support Vector Machine (SVM) to train a binary classifier and takes the normal vector of the decision boundary in the latent space as a meaningful direction. Considering the entire process was conducted in 2D space, InterFaceGAN suffers from the loss of image content as the yaw angle increases, making its results less consistent. LiftedGAN [23] utilizes the depth map as the shape representation and trains a couple of modules to render stably in various perspectives. Compared to InterFaceGAN [22], LiftedGAN maintains a more consistent image content with varying yaw angles. However, it suffers from severe image distortion as the yaw angle increases. StyleFlow achieves disentangled control over different style attributes, including gender, glasses, pose, lighting, and expression. Nevertheless, its results exhibit significant distortion at large yaw angles due to a limitation imposed by the latent space of StyleGAN. The recently proposed EG3D [4] can better preserve shape consistency than InterfaceGAN, LiftedGAN, and StyleFlow; however, it also exhibits distortions when the yaw angle becomes large, especially around the eyes. As demonstrated in Figure 4, our method achieves superior results in terms of view consistency while preserving the quality of the face.

4.3 Quantitative results

To validate the performance of our generative model, we utilize the Frechet Inception Distance(FID) [9] which quantifies the distance between two distributions of images. We use it to evaluate the generative power of our model by computing the distance between the distributions of the real-world data and our results. As our method is unable to synthesize the background, we paste our rendered image onto the original StyleGAN2 image for FID

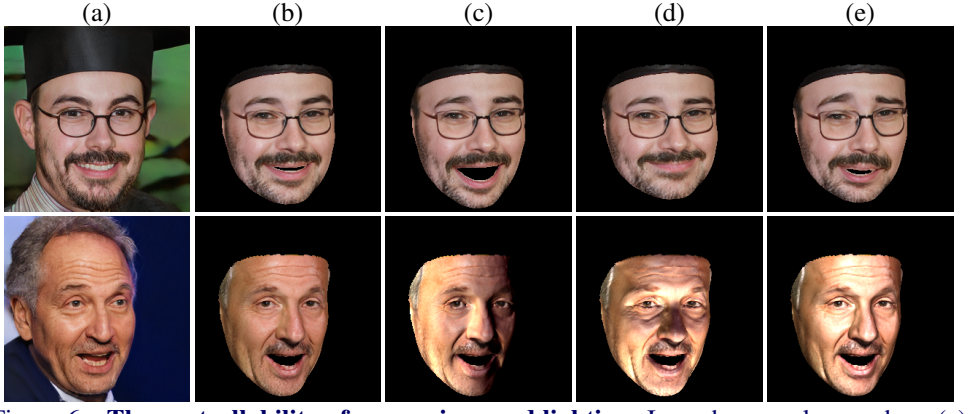

Figure 6: **The controllability of expressions and lighting**. In each example, we show (a) the StyleGAN2 image and (b) the rendered image using our method. From (c)-(e), we show the generated 3D face with manipulated expressions (top row) and lighting (bottom row).

evaluation. The FID scores of our method, LiftedGAN [23], and StyleGAN2 [13] are 11.68, 29.81, and 12.57 respectively. We also use Masked FID, a metric that measures the distance between the distributions of the foreground-masked real-world image and the rendered results. On masked FID, our method achieves 11.94, which outperforms StyleGAN2's 12.93.

4.4 Controllability

In light of the fact that our model has the same architecture as the StyleGAN2, the stylecodes are compatible with interpolating smoothly in the latent space. Consequently, smooth interpolation in the latent space allows smooth transitions between textured 3D face models. Figure 5 illustrates the smooth transition between two stylecodes. Moreover, we can also edit other 3D features such as expression and illumination by adjusting the 3DMM coefficients and rendering parameters. Figure 6 demonstrates some examples.

5 Conclusions

We propose StyleFaceUV, a framework for generating controllable 3D face models by augmenting a pre-trained StyleGAN2 model with 3DMM face models, diffuse maps, and generalized displacement maps. With the 3DMM model as our base face representation, we can control the appearance of the generated face. Our framework also predicts an additional diffuse map and a generalized displacement map to improve the visual quality of generated faces. To maintain the multi-view consistency of the predicted maps, we introduce a novel multi-view consistency loss using weighted masks. Our experiments demonstrate that our method maintains multi-view consistency across face images while avoiding image distortions under large viewing angles.

Acknowledgments. This work was supported in part by grants MOST 110-2221-E-002-124-MY3 and JSPS KAKENHI Grant Number JP21F20075. We thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources.

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.*, 40(3), May 2021. ISSN 0730-0301. doi: 10.1145/3447648. URL https://doi.org/10.1145/3447648.

[2] 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*, pages 187–194, 1999.

[3] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proc. CVPR*, 2021.

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *arXiv*, 2021.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20 (3):273–297, 1995.

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

[7] Yudong Guo, Jianfei Cai, Boyi Jiang, Jianmin Zheng, et al. Cnn-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE transactions on pattern analysis and machine intelligence*, 41(6):1294–1307, 2018.

[8] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[10] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[15] Myunggi Lee, Wonwoong Cho, Moonheum Kim, David Inouye, and Nojun Kwak. Styleuv: Diverse and high-fidelity uv map generative model. *arXiv preprint arXiv:2011.12893*, 2020.

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11662–11672, 2021.

[17] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? 2018. doi: 10.48550/ARXIV.1801.04406. URL https://arxiv.org/abs/1801.04406.

[18] Xingang Pan, Bo Dai, Ziwei Liu, Chen Change Loy, and Ping Luo. Do 2d gans know 3d shape? unsupervised 3d shape reconstruction from 2d image gans. *arXiv preprint arXiv:2011.00844*, 2020.

[19] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009.

[20] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020.

[21] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020.

[22] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[23] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2d stylegan for 3d-aware face generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6258–6266, 2021.

[24] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6142–6151, 2020.

[25] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7346–7355, 2018.

[26] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020.

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[28] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv preprint arXiv:2010.09125*, 2020.