Next3D: Generative Neural Texture Rasterization for 3D-Aware Head Avatars

Jingxiang Sun[1] Xuan Wang[2] Lizhen Wang[14] Xiaoyu Li[3] Yong Zhang[3]
Hongwen Zhang[1] Yebin Liu[1]
[1]Tsinghua University [2]Ant Group [3]Tencent AI Lab [4]NNKosmos

In this supplement, we first provide additional experiments (Sec. 1). Furthermore, we discuss the implementation details of our model including the network architecture, training strategies, and hyperparameters (Sec. 2). We also report the experiment details, such as one-shot facial avatars and 3D-aware stylization (Sec. 3). Lastly, we provide additional visual results as a supplement of the main paper (Sec. 4). Please refer to our accompanying supplemental video for more results.

1. Additional experiments

1.1. Deformation-aware discriminator

We propose a deformation-aware discriminator which additionally takes the synthetic renderings as input. Furthermore, we also take experiments on the parameter conditioning method proposed in GNARF [1]. Specifically, we first train our model without either synthetic renderings or FLAME parameters conditioning for about two days. Then, we test two methods based on the same checkpoint and report the changing trend of FID scores for two methods in Fig. 1. The discriminator with synthetic rendering input converges to a better FID score, while the one conditioned on FLAME parameters incurs divergency. Note that we have added random noise to the FLAME parameters for better convergency following GNARF.

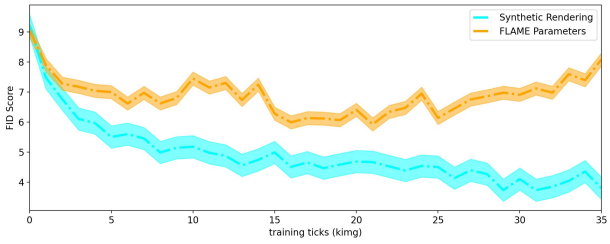

Figure 1. Training convergency with the discriminator designs.

Figure 2. Ablation study on the training strategies of 3D-aware stylization.

1.2. Training strategy of 3D-aware stylization

We conduct an ablation study on two strategies for freezing layers of the generator during 3D-aware stylization. The first one is the default setting following StyleGAN-NADA [5] that freezes all toRGB layers in the synthesis network. Though it works in 2D space, we found it leads to degraded image quality and dissymmetry. To this end, we adopt another strategy which optimizes the last toRGB layer for each synthesis network. In our case, there are three StyleGAN-based synthesis network including a neural texture generator G_{uv}, a static tri-plane generator G_{static}, and a teeth completing module G_{teeth} so we add the last toRGB layers of these three synthesis networks into optimization. As can be seen in Fig. 2, the second strategy improves the synthesis quality.

2. Implementation details

We implemented our 3D GAN framework on top of the official PyTorch implementation of EG3D [2] [1]. We adopt several hyperparameters and training strategies of EG3D including blurred real images at the beginning, pose-conditioned generator, density regularization, learning rates of the generator and discriminator. Due the limitation of computing material, we drop the two-stage training strategy

[1]https://github.com/NVlabs/eg3d

Figure 3. Detect the landmarks related to eyes.

and fix the neural rendering resolution to 64 and the final resolution to 512 instead.

Our teeth completing module G_{teeth} receives the cropped teeth features as input. To crop different-sized mouths into a unified size while varying expressions, we obtain the 2D mouth landmarks and FLAME parameters for each driving image during training and inference. Next, we crop the mouth features on the tri-planes into squares by setting the side e length and centering location based on these mouth landmarks. Note that these cropped squared feature maps are of different sizes due to different expressions and we resize them to 64x64 for later synthesis networks. Finally, the output mouth features are resized and transformed inversely.

2.1. Data preprocessing

We use FLAME template model to drive the facial deformation and use DECA [4] to extract FLAME parameters. Since there is no suitable model to accurately extract eye poses, we optimize eye poses with an off-the-shelf landmark detector [2]. Specifically, the detector extracts five landmarks around the eyes, as shown in Fig. 3. Accordingly, we select five vertices on the template mesh and the optimizable variables of eye poses are yaw and pitch. To optimize eye poses of a given portrait image, we minimize the re-projection errors of the vertices and detected landmarks by the PyTorch-implemented gradient descent. Since the FLAME template mesh has a different scale to the pretrained EG3D model, we initially rescale the template by 2.5 for a coarse visual alignment and fine-tune the translation and scale during training.

[2]https://mediapipe.dev/

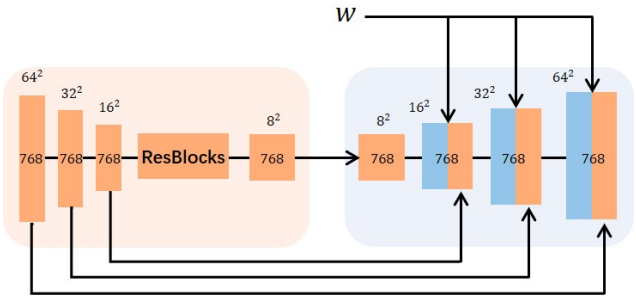

Figure 4. The detailed architecture of G_{teeth}.

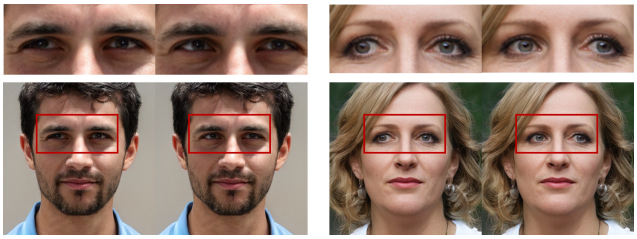

Figure 5. Gaze control while fixing identity and other expression parameters.

2.2. Generator

Our generator introduces a style-unet-based teeth completing module G_{teeth} whose architecture is illustrated in Fig. 4. The left part encodes the concatenated tri-plane teeth textures with dimensions of 768 (256×3) into multi-scale feature maps ranging from 64^2 to 8^2. Then the feature map with a resolution of 8^2 is processed into the residual blocks and fed into the right generator as the input feature map. Finally, the generator outputs a $64 \times 64 \times 768$ feature map.

3. Experiment details

Inversion-based one-shot facial avatars. We use an off-the-shelf face detector [3] to extract camera poses and crop the portraits in the wild to be consistent with the trainingset. We further extract the FLAME parameters and obtain the template mesh for each image by DECA [4]. Following Pivotal Tuning Inversion (PTI) [6], we first optimize the latent code for 450 iterations and then fine-tune the generator weights for an additional 500 iterations.

3D-aware stylization. Following StyleGAN-NADA [5], We optimize partial generator weights with others fixed. In practice, we fixed all toRGB layers of the synthesis blocks except for the last ones for the texture generator and static generator. We also fix the NeRF decoders for preventing the 3D consistency from degeneration.

4. Additional visual results

In this section, we provide additional visual results as a supplement to the main paper. Fig. 5 provides examples of gaze animation. Fig. 6 provides selected examples of four certain expressions and poses, highlighting the image quality, expression controllability (e.g. gaze animation), and the diversity of outputs produced by our method. Fig. 7 provides a qualitative comparison against baselines on facial animation.

Fig. 8 provides more results of animated virtual avatars with high-quality shapes. Note that the motions of eyelids can be reflected on the extracted meshes. Furthermore, the eyes are modeled as convex, suggesting that "hollow face illusion" is alleviatived. This is because while the gaze directions are highly pose-related, the rotated eyeballs in the template mesh provide an explicit gaze direction signal and thus helps to model such pose-related attribute and decouple them during inference.

Finally, we show additional results of the applications of our methods including one-shot avatars for real portraits and 3D-aware stylization in Fig. 9. We encourage readers to view the accompanying supplemental video for the dynamic results.

Figure 6. Generated examples with selected expressions and poses.

Figure 7. Qualitative comparison against baselines.

Figure 8. Animated virtual avatars with high-quality shapes.

Figure 9. Visual results of one-shot avatars for real portraits and 3D-aware stylization.

References

[1] Alexander W. Bergman, Petr Kellnhofer, Wang Yifan, Eric R. Chan, David B. Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. In *NeurIPS*, 2022. 1

[2] 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 *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2021. 1

[3] 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. 2

[4] Yao Feng, Haiwen Feng, Michael J Black, and Timo Bolkart. Learning an animatable detailed 3d face model from in-the-wild images. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 2

[5] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. 1, 2

[6] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 2