Supplementary Material for GETAvatar: Generative Textured Meshes for Animatable Human Avatars

Xuanmeng Zhang[1,2*] Jianfeng Zhang[2,3*] Rohan Chacko[2]
Hongyi Xu[2] Guoxian Song[2] Yi Yang[4] Jiashi Feng[2]
[1]ReLER, AAII, University of Technology Sydney [2]ByteDance
[3] National University of Singapore [4]ReLER, CCAI, Zhejiang University

Abstract

In the supplementary document, we first present the implementation details in Sec. A. Second, we provide the experimental details in Sec. B. Finally, we show more visualization results in Sec. C. Please also refer to the project page for video results. Our project page: `https://getavatar.github.io/`.

A. Implementation Details

A.1. Network Architectures

Triplane. Following EG3D [3] and GET3D [5], we adopt the StyleGAN2 [8] generator to generate the triplane representation. Specifically, the backbone (StyleGAN2 [8] generator) produces two triplanes: the texture triplane and the geometry triplane. We employ two conditional layers for each style block to generate geometry features and texture features separately [11, 5]. For each triplane, the backbone outputs a 96-channel output feature map that is then reshaped into three axis-aligned feature planes, each of shape $256 \times 256 \times 32$.

Mapping Network. Both the geometry and texture mapping network are 8-layer MLPs network with leakyReLU as the activation function, and the dimension of the hidden layers is 512. We sample the input latent code $z_{geo} \in \mathbb{R}^{512}$ and $z_{tex} \in \mathbb{R}^{512}$ from a 512-dimensional standard Gaussian distribution.

Discriminator. We use 3 StyleGAN2-based [8] discriminators to perform adversarial training on RGB images, 2D masks, and normal maps, respectively. Following EG3D [3], we condition all the discriminators on the camera parameters by modulating the blocks of the discriminator via a mapping network.

SMPL-guided Deformation. SMPL defines a deformable mesh $\mathcal{M}(\beta, \theta) = (\mathcal{V}, \mathcal{S})$, where θ denotes the pose parameter, β represents the shape parameter, $\mathcal{V}$ is the set of $N_v = 6890$ vertices, and $\mathcal{S}$ is the set of linear blend skinning weights assigned for each vertex. The template mesh of SMPL can be deformed by linear blend skinning [10] with θ and β. Specifically, the linear blend skinning process can transform a vertex from the canonical pose to the target pose by the weighted sum of skinning weights that represent the influence of each bone and transformation matrices. In this work, we generalize the linear blend skinning process [10] of the SMPL model from the coarse naked body to our generated clothed human. The core idea is to associate each point with its closest vertex on the deformed SMPL mesh $\mathcal{M}(\theta, \beta)$, assuming they undergo the same kinematic changes between the deformed and canonical spaces. Specifically, for a point $\mathbf{x_d}$ in the deformed space, we first find its nearest vertex v^* in the SMPL mesh. Then we use the skinning weights of v^* to un-warp $\mathbf{x_d}$ to $\mathbf{x_c}$ in the canonical space:

$$\mathbf{x_c} = \left(\sum_{i=1}^{N_j} s_i^* \cdot B_i(\theta, \beta) \right)^{-1} \cdot \mathbf{x_d}, \tag{1}$$

where $N_j = 24$ is the number of joints, s_i^* is the skinning weight of vertex v^* w.r.t. the i-th joint, $B_i(\theta, \beta)$ is the bone transformation matrix of join i. Therefore, for any point $\mathbf{x_d}$ in the deformed space, we can determine the SDF value $d(\mathbf{x_d})$, color $c(\mathbf{x_d})$, and normal $n(\mathbf{x_d})$ as:

$$\begin{aligned} d(\mathbf{x_d}) &= d(\mathbf{x_c}), \ \ c(\mathbf{x_d}) = c(\mathbf{x_c}), \\ n(\mathbf{x_d}) &= \left(\sum_{i=1}^{N_j} s_i^* \cdot R_i(\theta, \beta) \right) \cdot n(\mathbf{x_c}), \end{aligned} \tag{2}$$

where $d(\mathbf{x_c})$, $c(\mathbf{x_c})$, $n(\mathbf{x_d})$ are the SDF value, color, and normal at the point $\mathbf{x_c}$ in the canonical space and $R_i(\theta, \beta)$ is the rotation component of $B_i(\theta, \beta)$.

Differentiable Marching Tetrahedra. To explicitly model the body surface, we extract a triangular mesh of the generated human from the tetrahedral grid via the differentiable

[*] Equal contribution.

marching tetrahedra algorithm [17]. For the tetrahedra grid, the marching tetrahedra algorithm [17] finds the surface boundary based on the sign of the signed distance value for vertices within each tetrahedron. If two vertices i and j in the edge of a tetrahedron have opposite signs for the signed distance value ($sign(d_i) \neq sign(d_j)$), we can determine the mesh face vertice by a linear interpolation between vertices i and j.

A.2. Training Protocol

Hyperparameters. We use Adam optimizer [9] with $\beta_1 = 0$, $\beta_2 = 0.99$, and the batch size of 32 for optimization. The learning rate is set to 0.002 for both the generator and the discriminator. Following StyleGAN2 [8], we use lazy regularization to stabilize the training process by applying R1 regularization to discriminators every 16 training steps. Here we set the regularization weight to 10 for THUman2.0 [18] and 20 for RenderPeople [1]. For the loss function, we set $\lambda_{eik} = 0.001$ for the eikonal loss, and $\lambda_{ce} = 0.01$ for the cross-entropy loss of SDF regularization.

Runtime Analysis. At training time, for images at 512^2 resolution, we train the model on 8 NVIDIA Tesla V100 GPUs using a batch size of 32 for 1 day. For images at 1024^2 resolution, the models are trained on 8 NVIDIA A100 GPUs for 1 day, with a batch size of 32. At test time, we evaluate the rendering speed in frames per second (FPS) at different resolutions. In particular, our model runs at 512^2 resolution with 17FPS and 1024^2 resolution with 14FPS on a single NVIDIA Tesla V100 GPU.

B. Experimental Details

B.1. Datasets

We conduct experiments on two high-quality 3D human scan datasets: THUman2.0 [18] and RenderPeople [1]. For every scan on these datasets, we render 100 RGB images, 2D silhouette masks, and normal maps with randomly-sampled camera poses. Specifically, we sample the pitch and yaw of the camera pose from a uniform distribution with the horizontal standard deviation of 2π radians and the vertical standard deviation of 0.1 radians. Besides, we use a fixed radius of 2.3 and the fov angle of $49.13°$ for all camera poses. For the SMPL parameters, we adopt the official provided SMPL fitting results for THUman2.0 [18], and the SMPL fitting results provided by AGORA dataset [15] for RenderPeople [1].

B.2. Evaluation Metrics

B.2.1 Texture Evaluation

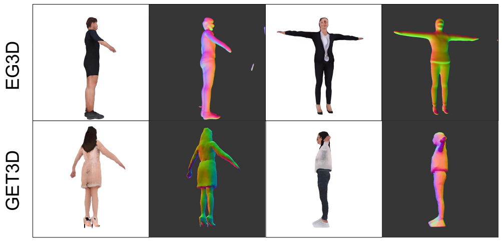
To evaluate the visual quality and diversity of the generated RGB images, we compute Frechet Inception Distance [6] between 50k generated RGB images and all real RGB images: FID$_{RGB}$. We adopt the FID implementation provided in the StyleGAN3 codebase.

B.2.2 Geometry Evaluation

We evaluate the geometry quality of generated human avatars from 3 aspects: the quality of surface details, the correctness of generated poses, and the plausibility of generated depth. First, to evaluate the quality of generated surface details, we measured Frechet Inception Distance [6] the normal maps: FID$_{normal}$, between 50k generated normal maps and all real normal maps. We adopt the widely-used implementation of FID with a pretrained Inception v3 feature extractor. Second, to measure the correctness of generated poses, we employ the Percentage of Correct Keypoints (PCK) metric, as used in previous animatable 3D human generation methods [13, 2, 7, 19]. To compute PCK, we first use a human pose estimation model to detect the human keypoints on both the generated and real images with the same camera and SMPL parameters. Then, we calculated the percentage of detected keypoints on the generated image within a distance threshold on the real image. Additionally, we evaluate the depth plausibility by comparing the generated depths with the pseudo-ground-truth depth estimated from the generated images by PIFuHD [16].

B.2.3 Baselines

When training the 3D-aware image synthesis [14, 5, 3] models, we follow the official implementations to train the model with only parameters. We also visualize the generated RGB images and normal maps in Fig. **??**.



https://github.com/ytrock/THuman2.0-Dataset
https://agora.is.tue.mpg.de/

https://github.com/NVlabs/stylegan3
https://github.com/NVlabs/stylegan3
https://api.ngc.nvidia.com/v2/models/nvidia/research/stylegan3/versions/1/files/metrics/inception-2015-12-05.pkl
https://github.com/nogu-atsu/ENARF-GAN
https://github.com/open-mmlab/mmpose
https://github.com/facebookresearch/pifuhd

C. Additional Results

We show more more generated images of the proposed method on THUman2.0 [18] (Fig. 1) and RenderPeople [1] (Fig. 2). We provide more transfer learning visualization results on in-the-wild datasets: DeepFashion [12] (Fig. 3) and SHHQ [4] (Fig. 4). In addition, we also make a comparison with the images generated by 2D GAN model StyleGAN2 [8] (Fig. 5). Please also refer to the supplementary video and our project page for more results.

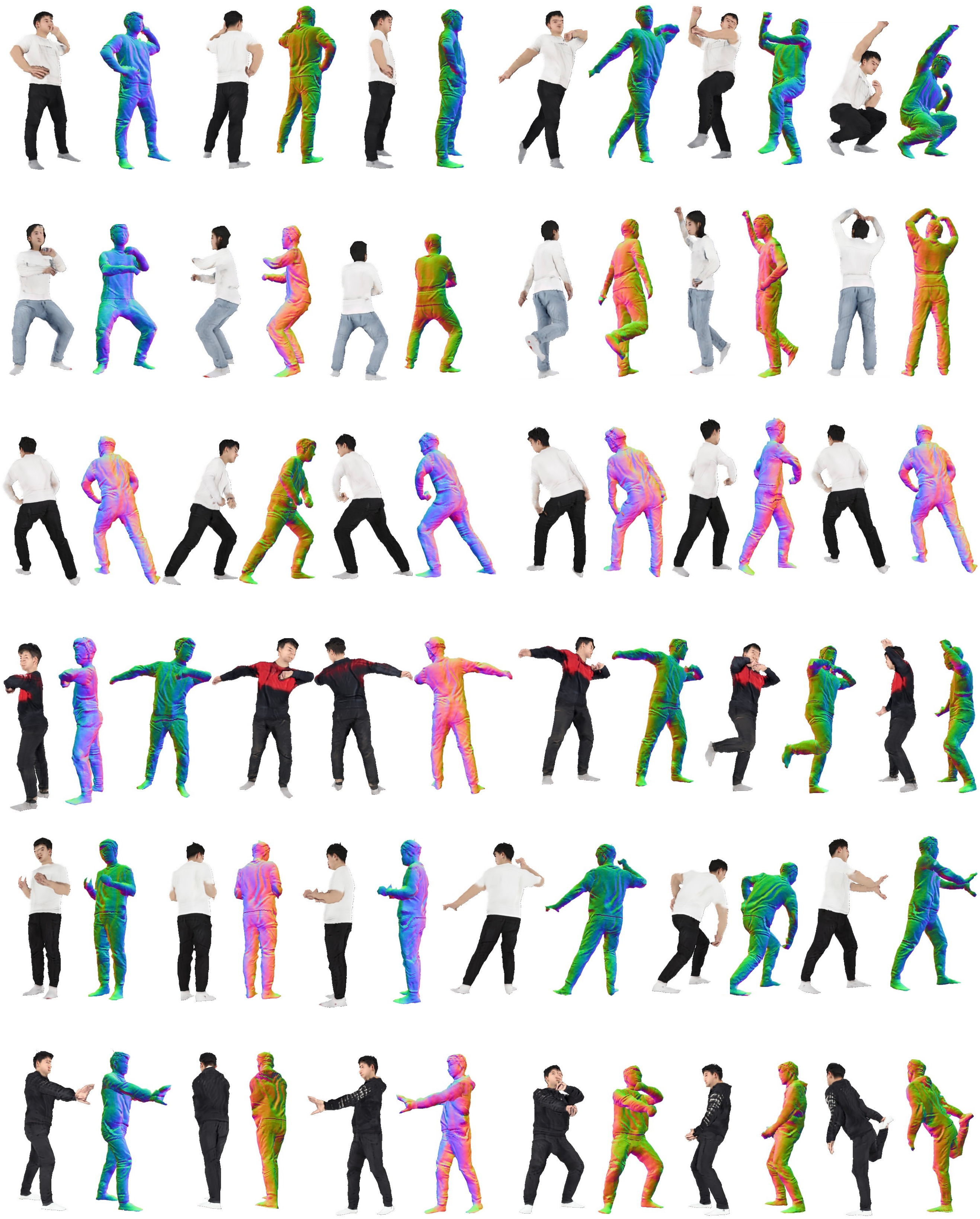


Figure 1: Images synthesized by GETAvatar on the THUman2.0 [18] dataset.

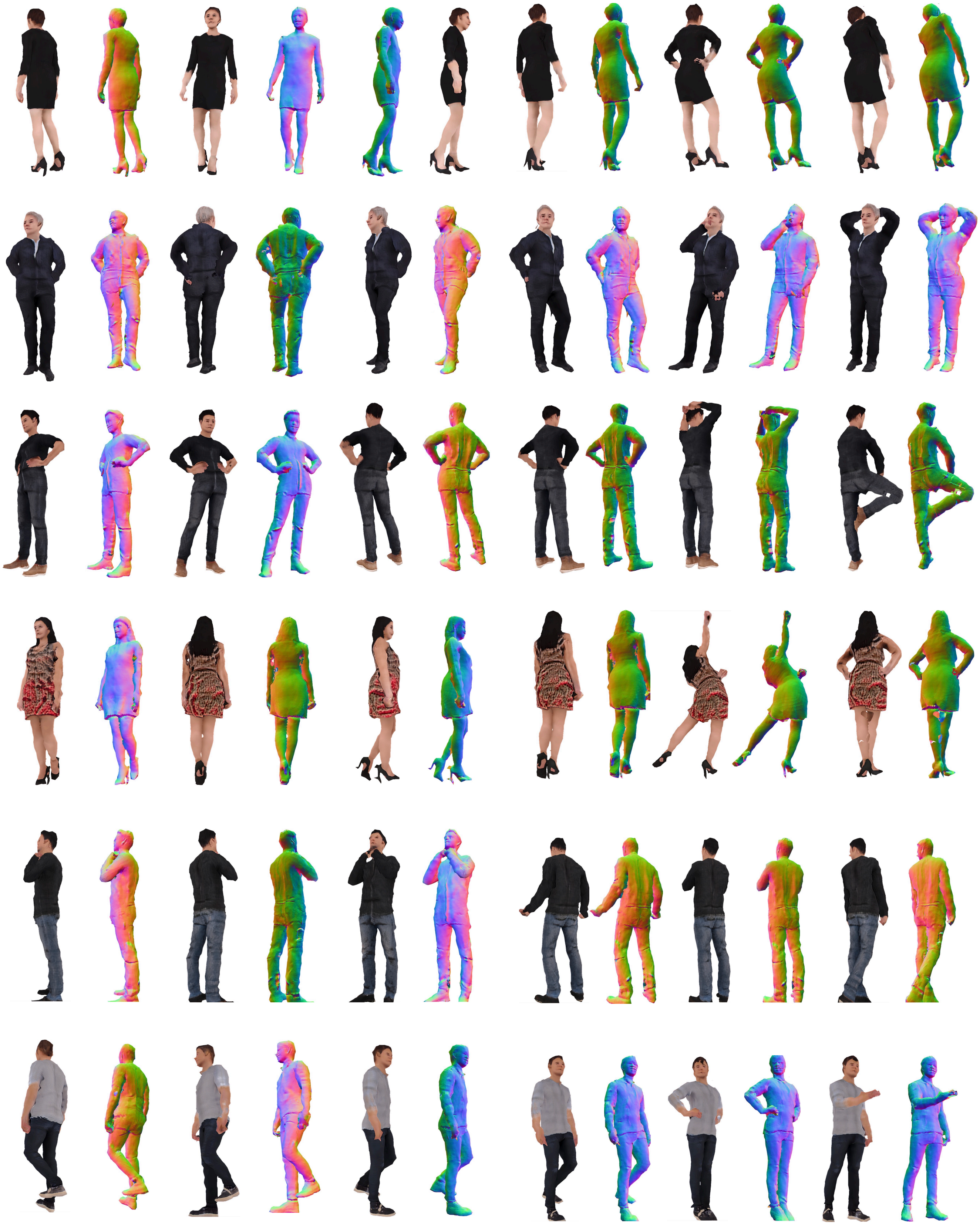


Figure 2: Images synthesized by GETAvatar on RenderPeople [1].

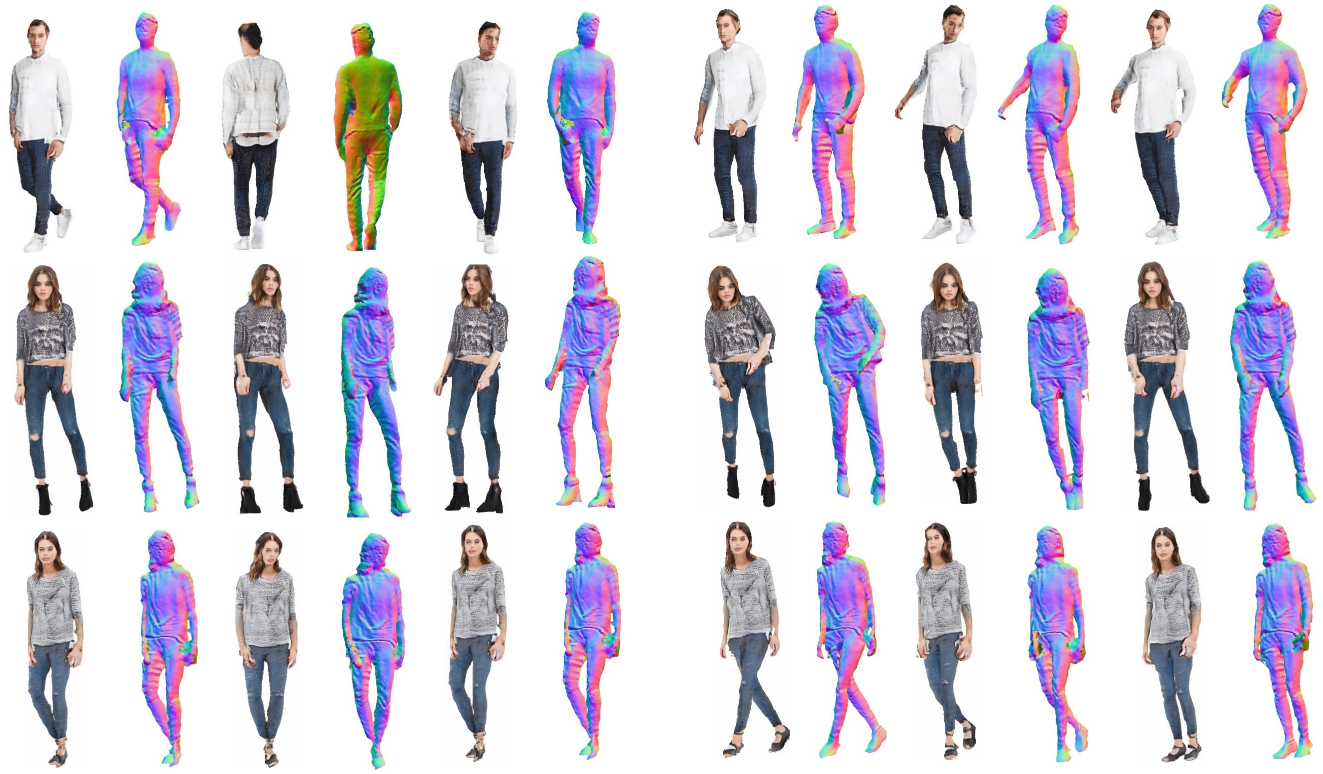


Figure 3: Images synthesized by GETAvatar on DeepFashion [12].

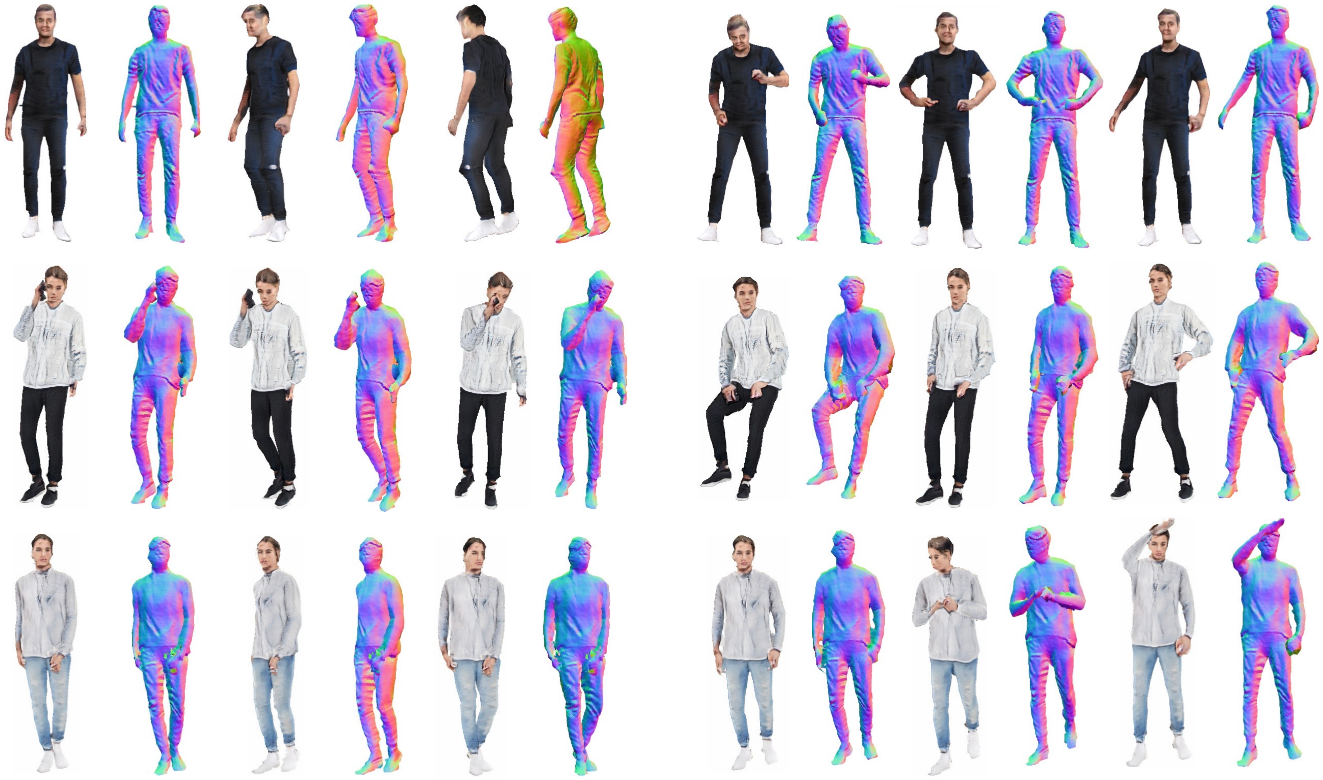
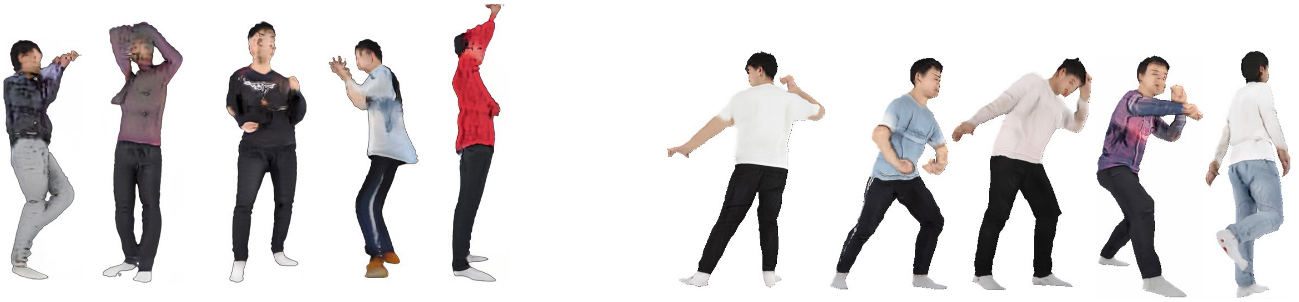


Figure 4: Images synthesized by GETAvatar on SHHQ [4].



StyleGAN2

Ours

Figure 5: Comparison of 2D StyleGAN2 [8] with our GETAvatar.

References

[1] Renderpeople, 2020. `https://renderpeople.com/`.

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

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022.

[4] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. Stylegan-human: A data-centric odyssey of human generation. In *ECCV*, 2022.

[5] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *NeurIPS*, 2022.

[6] 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. In *NeurIPS*, 2017.

[7] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. Eva3d: Compositional 3d human generation from 2d image collections. *arXiv preprint arXiv:2210.04888*, 2022.

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

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

[10] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

[11] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 2021.

[12] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[13] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Unsupervised learning of efficient geometry-aware neural articulated representations. In *ECCV*, 2022.

[14] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *CVPR*, 2022.

[15] Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black. Agora: Avatars in geography optimized for regression analysis. In *CVPR*, 2021.

[16] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020.

[17] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *NeurIPS*, 2021.

[18] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *CVPR*, 2021.

[19] Jianfeng Zhang, Zihang Jiang, Dingdong Yang, Hongyi Xu, Yichun Shi, Guoxian Song, Zhongcong Xu, Xinchao Wang, and Jiashi Feng. Avatargen: a 3d generative model for animatable human avatars. *arXiv preprint arXiv:2211.14589*, 2022.