SurMo: Surface-based 4D Motion Modeling for Dynamic Human Rendering (Supplementary Material)

Tao Hu, Fangzhou Hong, Ziwei Liu
S-Lab, Nanyang Technological University, Singapore

A. Implementation

A.1. Network Architectures

Motion Encoder and Decoder. The motion encoder is based on the Pix2PixHD [19] architecture with 3 Encoder blocks of [Conv2d, Batch- Norm, ReLU], ResNet [5] blocks, and 3 Decoder blocks of [ReLU, ConvTranspose2d, BatchNorm]. The motion decoder has 2 Decoder blocks.

Volume Renderer. We use a 5-layer MLP with a skip connection from the input to the 3th layer as in DeepSDF [13]. From the 4th layer, the network branches out two heads, one to predict density with one fully-connected layer and the other one to predict color features with two fully-connected layers.

Super-Resolution. To super-resolve low-resolution volumetric features to low-resolution images, we first bilinearly upsample the features by $2\times$ and then feed the upsampled features into two convolutional layers with a kernel size of 3 to upsample the images by a factor of 2.

Surface-based Triplane. The size of the triplane is $256 \times 256 \times 48$.

Discriminator. We adopt the discriminator architecture of PatchGAN [7] for adversarial training. Note that different from EG3D [2] that applies the image discriminator at both resolutions, we only supervise the final rendered images with adversarial training and supervise the volumetric features with reconstruction loss.

A.2. Optimization

SurMo is trained end-to-end to optimize $\mathcal{E}_{\mathcal{M}}$, $\mathcal{D}_{\mathcal{M}}$, and renderers $\mathcal{G}_1$, $\mathcal{G}_1$ with 2D image loss. Given a ground truth image I_{gt}, we predict a target RGB image $\mathbf{I}^{+}_{\mathbf{RGB}}$ with the following loss:

Pixel Loss. We enforce an ℓ_1 loss between the generated image and ground truth as $L_{pix} = \|I_{gt} - \mathbf{I}^{+}_{\mathbf{RGB}}\|_1$.

Perceptual Loss. Pixel loss is sensitive to image misalignment due to pose estimation errors, and we further use a perceptual loss [8] to measure the differences between the activations on different layers of the pre-trained VGG network [16] of the generated image $\mathbf{I}^{+}_{\mathbf{RGB}}$ and ground truth image I_{gt},

$$L_{vgg} = \sum \frac{1}{N^j} \left\| g^j\left(I_{gt}\right) - g^j\left(\mathbf{I}^{+}_{\mathbf{RGB}}\right)\right\|_2, \tag{1}$$

where g^j is the activation and N^j the number of elements of the j-th layer in the pretrained VGG network.

Adversarial Loss. We leverage a multi-scale discriminator D [19] as an adversarial loss L_{adv} to enforce the realism of rendering, especially for the cases where estimated human poses are not well aligned with the ground truth images.

Face Identity Loss. We use a pre-trained network to ensure that the renderers preserve the face identity on the cropped face of the generated and ground truth image,

$$L_{face} = \|N_{face}\left(I_{gt}\right) - N_{face}\left(\mathbf{I}^{+}_{\mathbf{RGB}}\right)\|_2, \tag{2}$$

where N_{face} is the pretrained SphereFaceNet [12].

Velocity Loss. We employ a velocity loss (temporal motion derivates) for the motion dececoding supervision,

$$L_{velocity} = \|V^{uv}_{gt(t+1)} - \mathbf{V^{uv}_{t+1}}\|_2, \tag{3}$$

where $V^{uv}_{gt(t+1)}$ is the ground truth velocity at timestep $t+1$, and $\mathbf{V^{uv}_{t+1}}$ is the predicted velocity by $\mathcal{D}_{\mathcal{M}}$ at timestep $t+1$.

Normal Loss. We also employ a surface normal loss (spatial motion derivates) for the motion dececoding supervision,

$$L_{normal} = \|N^{uv}_{gt(t)} - \mathbf{N^{uv}_{t}}\|_2, \tag{4}$$

where $N^{uv}_{gt(t)}$ is the ground truth normal at timestep t, and $\mathbf{N^{uv}_{t}}$ is the predicted normal by $\mathcal{D}_{\mathcal{M}}$ at timestep t. Note that in practical implementation, $\mathcal{D}_{\mathcal{M}}$ first predicts $\mathbf{N^{uv}_{t}}$, which is easier for the network than predicting $\mathbf{N^{uv}_{t+1}}$ directly, and $\mathbf{N^{uv}_{t+1}}$ can be drived and normalized from: $\mathbf{N^{uv}_{t+1}} = \frac{\partial \mathbf{P^{uv}_{t+1}}}{\partial \mathbf{x}} = \frac{\partial \{\mathbf{P^{uv}_{t}} + \mathbf{V^{uv}_{t+1}}\}}{\partial \mathbf{x}} = \mathbf{N^{uv}_{t}} + \frac{\partial \mathbf{V^{uv}_{t+1}}}{\partial \mathbf{x}}$. With the $\mathbf{V^{uv}_{t+1}}$ predicted for temporal motion supervision, the prediction of $\mathbf{N^{uv}_{t}}$ enforces a similar supervision with $\mathbf{N^{uv}_{t+1}}$ for the spatial motion learning.

Volume Rendering Loss. We supervise the training of volume rendering at low resolution, which is applied on the first three channels of $\mathbf{I_F}$, $L_{vol} = \|\mathbf{I_F}[:3] - I^D_{gt}\|_2$. I^D_{gt} is the downsampled reference image.

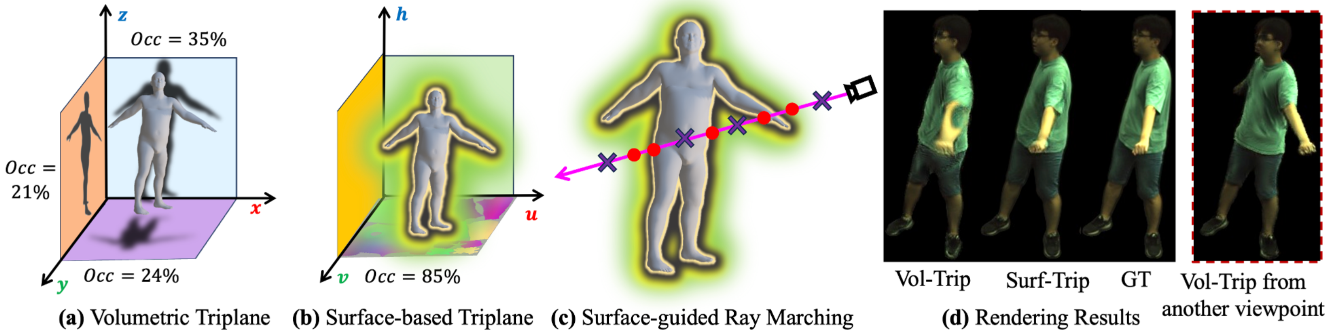

Figure 1. Illustration of Volumetric Triplane vs. Surface-based Triplane.

The networks were trained using the Adam optimizer [9]. The loss weights $\{\lambda_{pix}, \lambda_{vgg}, \lambda_{adv}, \lambda_{face}, \lambda_{velocity}, \lambda_{normal}, \lambda_{vol}\}$ are set empirically to $\{.5, 10, 1, 5, 1, 1, 15\}$. It takes about 12 hours to train a model from about 3000 images with 200 epochs on two NVIDIA V100 GPUs.

A.3. Training Data Processing.

We evaluate the novel view synthesis on three datasets: ZJU-MoCap [14] (including sequences of S313, S315, S377, S386, S387, S394) at resolution 1024×1024, MPII-RDDC [17] at resolution 1285×940, and AIST++ [10] at 1920×1080. Note that sequences of ZJU-MoCap used in Neural Body are generally short, *e.g.*, only 60 frames for S313. Instead, to evaluate the time-varying effects, we extend the original training frames of S313, S315, S387, S394 to 400, 700, 600, 600 frames respectively depending on the pose variance of each sequence, whereas S377 and S386 remain the same 300 frames as the setup of Neural Body [14]. 4 cameras are used for training, and the others are used in testing for ZJU-MoCap. 6 cameras are used in training, 3 for testing in AIST++, 18 cameras for training and 9 cameras for testing in MPI-RDDC.

B. Additional Experimental Results

B.1. Comparisons with SOTA Methods

Comparisons with 3D pose- and image-driven approaches. In contrast to pose-driven methods (*e.g.*, Neural Body [14], Instant-NVR [3], HumanNeRF [20]), DVA [15] and HVTR++ [6] propose to utilize both the pose and driving view features in rendering. They model both the pose and texture features in UV space, whereas ours is distinguished by modeling motions in a surface-based triplane, and we jointly learn physical motions and rendering in a unified network for faithful rendering.

Tab. 1 summarizes the quantitative results for novel view synthesis on the two sequences (S386 and S387) mentioned in DVA, which suggest that our method significantly out-

Table 1. Quantitative comparisons against the 3D pose- and image-driven approach DVA [15] and HVTR++ [6] on ZJU-MoCap datasets (averaged on all test views and poses) for novel view synthesis. To reduce the influence of the background, all scores are calculated from images cropped to 2D bounding boxes as used in [6]. Note that the training and test are conducted at the image resolution of 1024×1024 by following the setup in DVA [15]. For reference, we report the quantitative results of HVTR++ and DVA from the HVTR++ paper.

S386	LPIPS↓	FID↓	SSIM↑	PSNR↑
DVA [15]	.146	117.80	.791	26.209
HVTR++ [6]	.131	84.291	.797	26.517
Ours	**.108**	**72.556**	**.807**	**27.164**
S387	**LPIPS↓**	**FID↓**	**SSIM↑**	**PSNR↑**
DVA [15]	.166	142.67	.791	22.474
HVTR++ [6]	.136	101.03	.786	22.515
Ours	**.112**	**76.097**	**.808**	**23.581**

performs DVA and HVTRPP in terms of both per-pixel and perception metrics. Qualitative comparisons are provided in Fig. 2, which shows that our method produces sharper reconstructions with faithful wrinkles than both DVA and HVTR++. In contrast to the image resolution of 512×512 used in Neural Body [14], HumanNeRF [20] and Instant-NVR [3], DVA and HVTR++ were trained and evaluated at the resolution of 1024×1024 in [6, 15]. We follow the same protocol used in [6, 15] for fair comparisons.

Comparisons with PoseVocap [11]. PoseVocap [11] proposes joint-structured pose embeddings for better temporal consistency in rendering. Qualitative comparisons on novel view synthesis are shown in Fig. 3, which suggest that our method is capable of generating higher-quality wrinkles than PoseVocap [11]. Note that PoseVocap only provides qualitative results on ZJU-MoCap, and the test results of PoseVocap are reported in the paper [11].

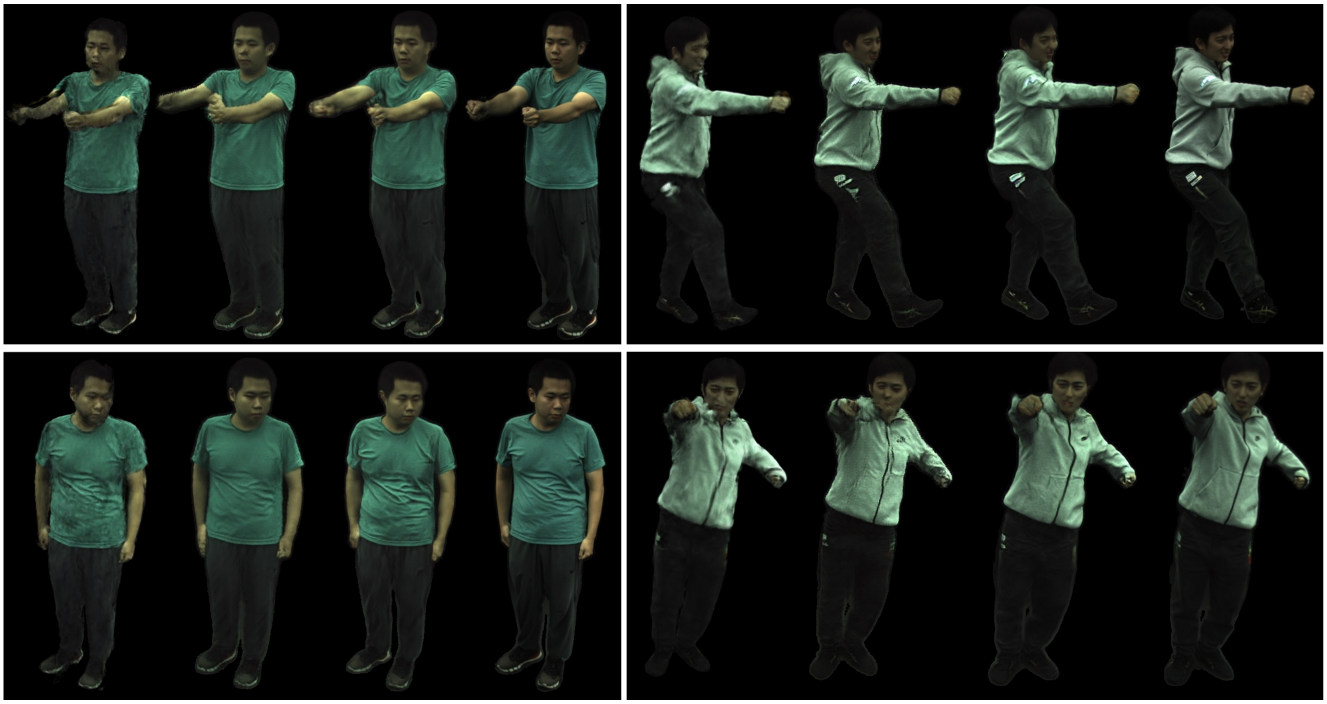

Figure 2. Qualitative comparisons against the 3D pose- and image-driven approach DVA [15] and HVTR++ [6] for novel view synthesis of training poses on ZJU-MoCap. For each example, from left to right: DVA, HVTR++, Ours, Ground Truth. Rendering results of DVA and HVTR++ are provided by the authors.

Figure 3. Qualitative comparisons against PoseVocap [11] for novel view synthesis of training poses on ZJU-MoCap.

Pose Generalization. Our method is focused on generating free-viewpoint video of dynamic humans, whereas we evaluate the pose generalization capability on ZJU-MoCap and it is observed that our method is not overfitted to the training poses, as suggested in Fig. 4 and Tab. 2.

Compared to ARAH [18] (a forward-skinning-based approach), the state-of-the-art method in pose generalization tasks, we generate better quantitative results in terms of novel view synthesis on training poses or novel poses as summarized in Tab. 2. The qualitative comparisons in Fig. 4 suggest that our method is capable of synthesizing higher-

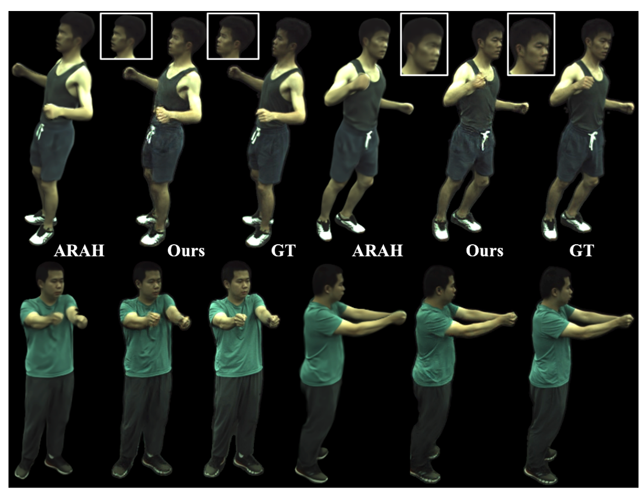

Figure 4. Qualitative comparisons against ARAH [18] for novel view synthesis of novel poses on ZJU-MoCap.

quality faces and cloth wrinkles than ARAH. Note that our method is not targeted at animation, and since the pose variance of ZJU-MoCap is not big enough, the experiments do not illustrate that our method achieves the SOTA results in animation tasks. However, the experimental results suggest

Table 2. Quantitative comparisons against ARAH [18] for novel view synthesis of training poses and novel poses on ZJU-MoCap datasets (averaged on all test views and poses) for novel view synthesis. To reduce the influence of the background, all scores are calculated from images cropped to 2D bounding boxes.

S377-Train	LPIPS↓	FID↓	SSIM↑	PSNR↑
ARAH [18]	.096	83.900	**.870**	25.176
Ours	**.069**	**63.008**	.866	**25.306**
S386-Train	LPIPS↓	FID↓	SSIM↑	PSNR↑
ARAH [18]	.112	99.614	**.808**	27.008
Ours	**.080**	**85.811**	.801	**27.069**
S377-Novel	LPIPS↓	FID↓	SSIM↑	PSNR↑
ARAH [18]	.116	106.46	**.821**	23.355
Ours	**.088**	**78.961**	.819	**23.594**
S386-Novel	LPIPS↓	FID↓	SSIM↑	PSNR↑
ARAH [18]	.150	114.24	**.742**	**25.031**
Ours	**.123**	**104.45**	.728	24.821

Table 3. Quantitative comparisons on MPII-RDDC datasets [4]. To reduce the influence of the background, all scores are calculated from images cropped to 2D bounding boxes.

Methods	LPIPS↓	FID↓	SSIM↑	PSNR↑
HumanNeRF [20]	.175	116.53	.615	17.443
Ours	**.153**	**107.79**	**.627**	**18.048**

Table 4. Quantitative comparisons on S13 and S21 sequences from AIST++ datasets [10]. To reduce the influence of the background, all scores are calculated from images cropped to 2D bounding boxes.

S13	LPIPS↓	FID↓	SSIM↑	PSNR↑
Neural Body [14]	.266	276.70	.732	**17.649**
Ours	**.183**	**161.68**	**.751**	17.488
S21	LPIPS↓	FID↓	SSIM↑	PSNR↑
Neural Body [14]	.296	333.03	.731	17.137
Ours	**.205**	**177.36**	**.757**	**17.334**

that our method is not completely overfitted to the training poses. We use the publicly released test results of ARAH for comparisons.

B.2. Quantitative Comparisons on MPII-RDDC and AIST++ Datasets.

The quantitative comparisons on MPII-RDDC [4] are summarized in Tab. 3, which suggests that our method outperforms HumanNeRF in the lighting-conditioned scenario. The quantitative comparisons on AIST++ [10] are summarized in Tab. 4, which confirms the effectiveness of our method in rendering fast motions.

B.3. Ablation study

Surface-based Triplane vs. Volumetric Triplane. We compare the volumetric triplane (Vol-Trip) [1] and our proposed surface-based triplane (Surf-Trip) for human modeling as shown in Fig. 1. It is observed that the volumetric triplane is a sparse representation for human body modeling, *i.e.*, only 21-35% features are utilized to render the human under the specific pose, and hence the Vol-Trip fails to handle the self-occlusions effectively as shown in Fig. 1 (d), though Vol-Trip generates plausible results from another viewpoint without sever self-occlusions. In contrast, about 85% surface-based triplane features are utilized in rendering. In addition, with surface-guided ray marching, our method is more efficient by filtering out invalid points that are far from the body surface.

Table 5. Ablation study of motion prediction and training views.

S313	LPIPS ↓	FID ↓	SSIM↑	PSNR↑
$w/o\ Pred$	.085	73.674	.834	24.908
$Pred_t$	.073	60.942	.848	25.537
$Pred_{t+1}$	**.060**	**50.170**	**.869**	**26.654**
$Pred_{t+1}(1\ view)$	.126	112.19	.788	22.830
S387	LPIPS ↓	FID ↓	SSIM↑	PSNR↑
$w/o\ Pred$	.115	93.688	.761	22.152
$Pred_t$	.096	83.825	.790	23.083
$Pred_{t+1}$	**.084**	**71.216**	**.810**	**23.735**
$Pred_{t+1}(1\ view)$	.151	128.18	.729	21.093

Motion Prediction. Predicting the next frame based on the status of the current frame is a one-to-many mapping problem. However, we take as input additional dynamics, and trajectory features to infer the motion of the next frame, which alleviates the one-to-many mapping issue. The paper is not focused on motion prediction/generation. Instead, we use the motion prediction to force a meaningful embedding of the feature space, which improves the rendering quality. Predicting the next motion frame $Pred_{t+1}$ offers higher-quality rendering than predicting the current motion frame $Pred_t$, *i.e.* $\mathbf{V^{uv}_{t+1}}$ vs. $\mathbf{V^{uv}_{t}}$, as listed in Tab. 5. We conduct experiments on the S313 and S387 sequences of the ZJU-MoCap dataset in Tab. 5.

Training Views. Tab. 5 suggests that the performances of novel view synthesis degrade with fewer training views, *i.e.*, from 4 training views $Pred_{t+1}$ to 1 view $Pred_{t+1}(1\ view)$. Even with 1 view, our performance is still comparable with Instant-NVR (Tab. 8).

Table 6. Ablation study of dynamics conditioning.

Methods	LPIPS↓	FID↓	SSIM↑	PSNR↑
Norm. + Velo. [21]	.093	81.900	.825	24.113
Ours w/ D_{cond}	.085	73.674	.834	24.908
Ours w/ V_{pred}, N_{pred}	.060	50.170	.869	26.654

Table 7. Ablation study of super-resolution module under different image resolutions and upsampling factors.

Methods	LPIPS↓	FID↓	SSIM↑	PSNR↑
512^2, $\times 2$	.060	49.714	.870	26.678
512^2, $\times 4$	.070	56.456	.854	26.166
1024^2, $\times 2$	.076	54.563	.862	26.063

Dynamics Conditioning. We compare the methods of conditioning dynamics in the rendering network between [21] and ours. [21] takes as input the velocities of the past 10 consecutive poses and normal maps of the current pose, whereas we take as input the positional map of the current pose and aggregated trajectory of the past 5 frames as input. Tab. 6 suggests that our method enables better quantitative results, and we improve the performances by further learning motions, *e.g.*, surface velocity and normal prediction.
Super-resolution. Our method utilizes a super-resolution module to synthesize high-quality images. The quantitative results are summarized in Tab. 7. It is observed that the performances are improved when the upsampling factor is increased from 4 to 2, which indicates more geometric features are utilized by increasing the resolution of volumetric rendering.

B.4. Efficiency

At test time, our method runs at 3.2 FPS on one NVIDIA V100 GPU to render 512×512 resolution images, about 39× faster than Neural Body [14], 17× faster than HumanNeRF [20], and 9× faster than Instant-NVR [3].

B.5. Failure Cases

Our method fails to generate high-quality wrinkles for complicated textures of AIST++ [10], as shown in Fig. 5. This is because we cannot learn to infer dynamic wrinkles from the complicated appearances.

Figure 5. Failure cases.

References

[1] Eric Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, S. Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. *ArXiv*, abs/2112.07945, 2021. 4

[2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 1

[3] Chen Geng, Sida Peng, Zhen Xu, Hujun Bao, and Xiaowei Zhou. Learning neural volumetric representations of dynamic humans in minutes. In *CVPR*, 2023. 2, 5, 6

[4] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *ACM Transactions on Graphics (TOG)*, 40:1 – 16, 2021. 4

[5] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016. 1

[6] Tao Hu, Hongyi Xu, Linjie Luo, Tao Yu, Zerong Zheng, He Zhang, Yebin Liu, and Matthias Zwicker. Hvtr++: Image and pose driven human avatars using hybrid volumetric-textural rendering. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–15, 2023. 2, 3

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, pages 5967–5976, 2017. 1

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. volume 9906, pages 694–711, 10 2016. 1

[9] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[10] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation, 2021. 2, 4, 5

[11] Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. Posevocab: Learning joint-structured pose embeddings for human avatar modeling. In *ACM SIGGRAPH Conference Proceedings*, 2023. 2, 3

[12] Weiyang Liu, Y. Wen, Zhiding Yu, Ming Li, B. Raj, and Le Song. Sphereface: Deep hypersphere embedding for face

Table 8. Quantitative comparisons with Neural Body [14], Instant-NVR [3], HumanNeRF [20] on ZJU-MoCap. Instant-NVR* and Instant-NVR are trained with 100 and 30 epochs respectively, which generate better results than the official models that were trained with 6 epochs. Qualitative results can be found in the demo video.

	S313				S315				S377			
Models	LPIPS ↓	FID ↓	SSIM ↑	PSNR ↑	LPIPS	FID	SSIM	PSNR	LPIPS	FID	SSIM	PSNR
Neural Body	.152	149.43	.844	26.755	.108	112.57	.855	23.340	.119	132.16	.862	25.997
Instant-NVR	.199	153.46	.783	23.123	.230	175.68	.716	19.066	.173	123.24	.810	22.976
Instant-NVR*	.185	132.73	.783	23.029	.186	148.43	.704	18.592	.159	119.97	.806	22.884
HumanNeRF	.098	69.868	.822	24.870	.084	82.412	.830	21.314	.092	79.760	.804	24.651
Ours	.060	50.170	.869	26.654	.058	59.664	.868	23.125	.069	63.008	.866	25.306
	S386				S387				S394			
Neural Body	.148	133.74	.815	27.648	.215	173.33	.769	23.454	.217	169.12	.803	26.467
Instant-NVR	.171	137.29	.742	24.639	.237	161.94	.724	20.990	.251	159.11	.725	23.111
Instant-NVR*	.161	135.96	.736	24.591	.230	155.97	.724	21.070	.247	155.41	.727	23.244
HumanNeRF	.105	100.43	.763	26.590	.129	96.722	.762	22.452	.119	97.947	.766	24.643
Ours	.080	85.811	.801	27.069	.084	71.216	.810	23.735	.095	78.949	.787	25.237

recognition. *CVPR*, pages 6738–6746, 2017. 1

[13] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *CVPR*, 2019. 1

[14] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. *CVPR*, 2021. 2, 4, 5, 6

[15] Edoardo Remelli, Timur M. Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabián Prada, Jason M. Saragih, and Yaser Sheikh. Drivable volumetric avatars using texel-aligned features. *ACM SIGGRAPH*, 2022. 2, 3

[16] K. Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 1

[17] G. Varol, J. Romero, X. Martin, Naureen Mahmood, Michael J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. *CVPR*, pages 4627–4635, 2017. 2

[18] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *European Conference on Computer Vision*, 2022. 3, 4

[19] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 1

[20] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. *ArXiv*, abs/2201.04127, 2022. 2, 4, 5, 6

[21] Jae Shin Yoon, Duygu Ceylan, Tuanfeng Y. Wang, Jingwan Lu, Jimei Yang, Zhixin Shu, and Hyunjung Park. Learning motion-dependent appearance for high-fidelity rendering of dynamic humans from a single camera. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3397–3407, 2022. 5