Holoported Characters: Real-time Free-viewpoint Rendering of Humans from Sparse RGB Cameras —Supplementary Material—

Ashwath Shetty[1,2] Marc Habermann[1,3] Guoxing Sun[1] Diogo Luvizon[1,3]
Vladislav Golyanik[1] Christian Theobalt[1,3]
[1] Max Planck Institute for Informatics, Saarland Informatics Campus [2] Saarland University
[3] Saarbrücken Research Center for Visual Computing, Interaction and AI

We first present more technical details on our character model (Sec. A), and the depth testing approach for texel visibility (Sec. B). Then, we provide more implementation as well as data processing details (Sec. D and Sec. C), additional results, and applications of our method (Sec. E, Sec. F). We further show additional comparisons (Sec. G), ablations (Sec. H), and analysis of methods closest to our setting (Sec. I). Finally, we talk about the limitations of our approach (Sec. J).

A. Deformable Character Model

Our character model takes a temporal motion $M = \{(\boldsymbol{\theta}_{t-W}, \boldsymbol{\alpha}_{t-W}, \boldsymbol{z}_{t-W})....(\boldsymbol{\theta}_t, \boldsymbol{\alpha}_t, \boldsymbol{z}_t)\}$ as input and deforms a template mesh capable of modelling loose clothing. Here $\boldsymbol{\theta}_t \in \mathbb{R}^P, \boldsymbol{\alpha}_t \in \mathbb{R}^3, \boldsymbol{z}_t \in \mathbb{R}^3$ refer to the skeleton degrees of freedom, root translation, and root rotation, respectively. We leverage the explicit character representation of Habermann et al. [3]

$$C_i(\boldsymbol{\theta}_t, \boldsymbol{\alpha}_t, \boldsymbol{z}_t, \boldsymbol{A}, \boldsymbol{T}, \boldsymbol{d}_i) = \boldsymbol{v}_i, \tag{1}$$

as it is differentiable, real-time, and models loose clothing. In their character formulation, the initial template T is downsampled to an embedded graph [15] G with K nodes, and the parameters $\boldsymbol{A} \in \mathbb{R}^{K\times 3}, \boldsymbol{T} \in \mathbb{R}^{K\times 3}$ are the rotation and translation of each of the K nodes stacked on top of each other, describing the coarse deformation of T in canonical space. $\boldsymbol{d}_i \in \mathbb{R}^3$ is the per-vertex displacement in canonical space. The final location

$$\boldsymbol{y}_i = \boldsymbol{d}_i + \sum_{k\in N_{\mathrm{vn}(i)}} w_{i,k} R(\boldsymbol{a}_k)(\hat{\boldsymbol{v}}_i - \boldsymbol{g}_k) + \boldsymbol{g}_k + \boldsymbol{t}_k \tag{2}$$

of a vertex in canonical space $\boldsymbol{y}_i \in \mathbb{R}^3$ is determined by the weighted addition of the rotation and translation of its neighbours in the embedded graph $N_{\mathrm{vn}(i)}$, and finally adding the per-vertex deformation $\boldsymbol{d}_i$. Here, $w_{i,k}$ is the weight the ith vertex assigns to node k. $R(\cdot)$ is the function that converts Euler angles to matrices. $\boldsymbol{a}_k$, $\boldsymbol{t}_k$ are kth rows of $\boldsymbol{A}$, $\boldsymbol{T}$ respectively. $\hat{\boldsymbol{v}}_i$ is the ith vertex of T and $\boldsymbol{g}_k$ is the kth node of G.

Their model predicts $\boldsymbol{A}$, $\boldsymbol{T}$, and $\boldsymbol{d}_i$ using structure-aware graph neural networks[3], referred to as $f_{\mathrm{eg}}(e(M))$ and $f_{\mathrm{delta}}(d(M))$. $e(\cdot)$ and $d(\cdot)$ are their proposed motion to embedded graph embedding and motion to vertex embedding. Finally, they apply the skeleton pose to the deformed vertex in canonical space $\boldsymbol{y_i}$ to obtain the final deformed vertex location

$$\boldsymbol{v}_i = \boldsymbol{z} + \sum_{k\in N_{\mathrm{vn(i)}}} w_{i,k}(R_{\mathrm{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha})\boldsymbol{y}_i + \boldsymbol{t}_{\mathrm{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha})), \tag{3}$$

where the rotation $R_{\mathrm{sk},k}$, and translation $\boldsymbol{t}_{\mathrm{sk},k}$ for node k are determined by Dual Quaternion Skinning [5]. The vertex matrix $\boldsymbol{V} \in \mathbb{R}^{N\times 3}$ can be obtained by stacking $\boldsymbol{v_i}$.

B. Depth Testing

We address the texel visibility problem by using a depth testing approach. We assign a 3D depth to each texel $T_{\mathrm{pos}} \in \mathbb{R}^{TW\times TH\times 3}$, using the barycentric coordinates of the face it lands on. Then, we compute the depth $D_{c,V} \in \mathbb{R}^{H\times W\times 3}$ and the texel to pixel mapping $F_{\mathrm{warp_{c,V}}} : (u, v) \to (x, y)$ for a particular view c using differentiable rasterisation from DeepCap [2, 4] given the camera parameters and the deformed vertex positions $\boldsymbol{V}$. If the depth at a particular texel (u, v) is within a ϵ norm ball to the depth of the pixel it lands on, we mark it as visible, i.e.

$$T_{\mathrm{vis},i}(u, v) = |D_{i,V}(F_{\mathrm{warp_{i,V}}}(u, v)) - T_{\mathrm{pos}}(u, v)| < \epsilon. \tag{4}$$

C. Additional Data Processing Details

Each actor in our dataset is scanned using a commercially available 3D scanner [17] where the mesh is obtained from multi-view stereo reconstruction[1]. Following this, we

[1] https://www.agisoft.com

	Novel View		Novel Pose	
Method	**CD↓**	**HD↓**	**CD↓**	**HD↓**
DDC	11.53	11.24	15.5	15.6
HDHumans	11.52	11.21	13.7	13.5
Ours	**10.4**	**9.3**	**13.1**	**12.9**

Table 1. **Quantitative Comparison on Surface Distance.** We evaluate the surface tracking in the *novel view* and *novel pose* setting on subject *S1* and report the Chamfer (CD) and Hausdorff (HD) distance with respect to the ground truth. Note that our approach outperforms previous works, thanks to the additional Chamfer penalty. This is significant in enhancing the quality of our projective texturing pipeline. Error reported in mm.

Component	**FPS↑**	**Latency↓**
Character Model	100	$0.010s$
Projective Texturing	27	$0.037s$
TexfeatNet	100	$0.010s$
SRNet	31.25	$0.032s$

Table 2. **Runtime Breakdown.** Here, we present a component-wise runtime breakdown of our pipeline in terms of frames per second (fps) and latency (in seconds s). Note that all of our components run within the real-time limit of 25 fps.

downsample the high-resolution mesh to around 9000 faces for each character. For character rigging, we apply markerless motion capture [16] on the multi-view images from the scanner to obtain the skeletal pose. Given the pose and the template scan, we apply Blender's[2] automated skinning weight computation. The UV parameterization is obtained from a photometric stereo reconstruction software. However, the effect of UV parameterization and optimizing the UV parameterization is a future work that we believe merits further investigation.

D. Implementation Details

The obtained multi-view frames are processed with foreground segmentation [9] and per-frame mesh reconstruction using NeuS2 [18]. Motion tracking is obtained with a commercial markerless capture system [16]. Our method is implemented in TensorFlow [1] and trained using the Adam optimizer [6] with a constant learning rate of 10^{-4} until convergence.

The character model and the TexFeatNet are supervised on images with a resolution of 1028×752 pixels, and the SRNet is trained with a full image resolution of 4112×3008 pixels. The projective texturing module uses images of resolution 4112×3008 as input, and the texture maps are generated at a resolution of 1024×1024 pixels. We randomly sample 10K points from the reconstructed surface for the Chamfer loss for every frame.

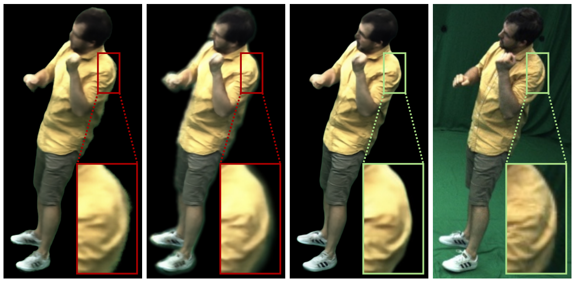

Figure 1. **Qualitative Ablations.** Note that our design choices for the SRNet module, lead to qualitatively better results, especially at the borders of the human.

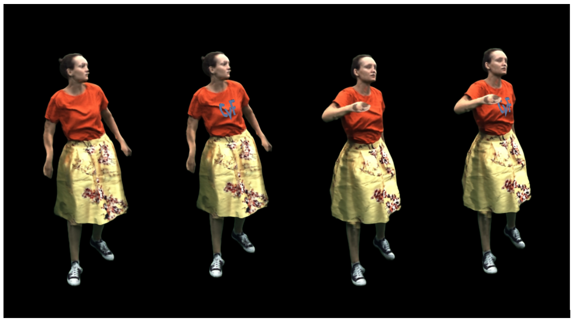

Figure 2. **Application: Texture Editing.** As we use explicit textures as an underlying latent for appearance, and they are temporally and spatially aligned, we can perform 2D texture edits such as adding a logo onto the character's shirt.

Method	**PSNR↑**	**LPIPS↓** (×1000)	**FID↓**	**Res.**
w/ Naive $\mathcal{L}_{\text{sr}}$	27.01	55.10	37.25	4K
w/ High RF	28.13	36.18	19.97	4K
Ours	28.75	**32.4**	**17.42**	4K

Table 3. **Quantitative Ablations.** Here, we ablate some design choices in our SRNet module, in the novel pose setting on subject *S1*. Note that choosing our $\mathcal{L}_{\text{sr}}$ formulation, and keeping a shallower architecture lead to better performance.

TexFeatNet utilizes a UNet architecture [14], and SRNet is implemented as a shallow architecture with enhanced residual blocks [7]. Our complete framework is trained using two Nvidia A100 GPUs with 80GB memory and a batch size of four. Tab. 2 provides a runtime breakdown of our modules.

[2] https://www.blender.org

Method	PSNR↑	LPIPS↓ (×1000)	FID↓	Res.
w/o Texture	25.82	41.36	55.17	1K
w/o Features	28.37	31.30	21.05	1K
w/o Chamfer	27.83	30.35	15.76	1K
w/o SR	28.42	31.11	20.85	1K
w/o 4K	**28.85**	30.50	18.01	1K
Ours	28.03	**28.49**	**13.26**	1K
Ours w/o 4K	**27.72**	34.49	20.89	4K
Ours	27.22	**33.17**	**15.26**	4K

Table 4. **Quantitative Ablations.** Here, we ablate some major design choices in the novel pose setting on subject *S2* (loose clothing). Note that the efficacy of our design choices translates along subjects.

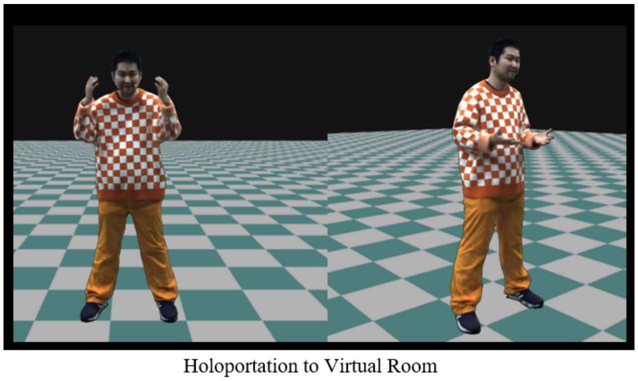

Figure 3. **Application: Holoportation.** Our method produces high-quality expressions, hands, and wrinkles in real time; hence, it is well suited for telepresence applications like placing a character in a virtual room.

E. Additional Results

We also provide additional results for our method on more subjects in the novel view and novel pose setting (Fig. 7 and Fig. 8). Our method also allows for exciting applications like texture editing (Fig. 2) and placing the character in a virtual room (Fig. 3).

F. End-to-end Sparse Camera Demo

We additionally show the result of our method using a 3D skeletal pose recovered from four and eight cameras, respectively, for different subjects (Fig. 4). Note that the result with four cameras demonstrates that our method can also be integrated into an end-to-end sparse camera setup.

G. Additional Comparisons

We present additional qualitative comparisons in the novel view and novel pose setting to animatable representations (Fig. 9) and real-time sparse image-driven methods (Fig. 10 and Fig. 12). We also present a zoomed-in face comparison of our method to show our efficacy in capturing face expressions (Fig. 5). In Tab. 1, we compare the quality of our geometry reconstruction against competing methods. Our approach provides quantitative improvements in the Chamfer (CD) and Hausdorff metrics (HD).

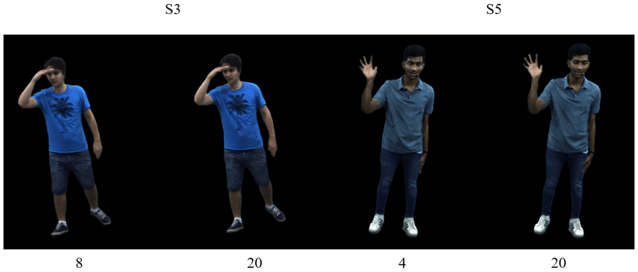

Figure 4. **Sparse Camera Pose Tracking.** We present results using 3D pose tracking from fewer cameras (numbers below image represent number of cameras used). The tracking inaccuracies lead to artifacts; however, the quality remains relatively high even with four cameras.

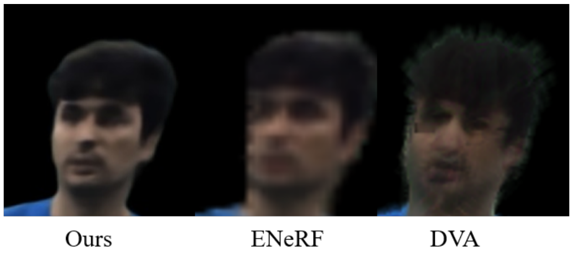

Figure 5. **Comparison with a zoomed-in virtual view.** Here we present a result, where we render a view close to the face. Notice the ability of our method to capture facial expressions with much higher fidelity.

H. Additional Ablations

Here, we provide some additional ablation results (Fig. 1 and Tab. 3). We ablate our SRNet module by replacing it with a naive loss that is the same as $\mathcal{L}_{\text{ren}}$ and find that this performs quantitatively worse and also qualitatively (especially at the borders). Also, replacing a shallow SRNet architecture with a deeper architecture that utilizes UNet [14], and additional upsampling layers leads to worse multi-view consistency, which can be seen quantitatively and qualitatively. Additionally, we also add an ablation for our major components, on a subject in loose clothing (Tab. 4).

I. ENeRF and DVA Analysis

We additionally provide more details on how we compared to ENeRF [8] and DVA [13], as they are closely related to our setting. ENeRF is a general method that generates novel views from sparse source camera views and even

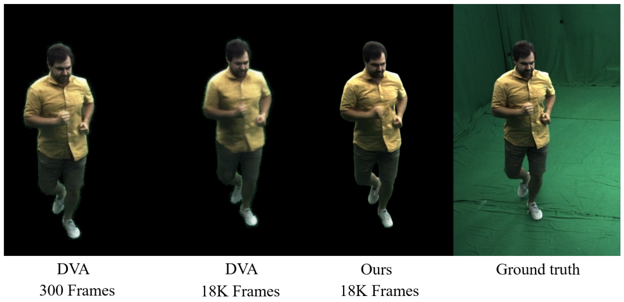

Figure 6. **Large-scale Training.** We present a novel view synthesis (replay), the result from DVA when it is trained on 300 frames, and when it's trained on 18000 frames. Note that the frame visualized is part of the training frames. While DVA can capture high-fidelity details, when trained with fewer frames, the performance deteriorates as the number of frames increases. In contrast, ours maintains high fidelity, even as the number of frames increases.

demonstrates impressive generalizability to completely unseen scenes. However, in their setting, they utilize *nearest* views *at test time* from a dense setup to produce the target view, which violates our requirement for only a fixed number of cameras at test time. Hence, to compare with them, we retrain their method in the same setting as ours (inference using four *fixed cameras*). We observe their method faces multi-view consistency artifacts due to a lack of reliable human priors for depth, and as their final color is a weighted combination of the source colors (see Fig. 10 and Fig. 12).

DVA achieves photorealistic telepresence in real-time from sparse cameras and 3D skeletal pose. They released a version of their source code, which relies on SMPLX [11] for mesh tracking and provided scripts to reproduce results on the ZJU dataset [12]. First, we reproduce results on the ZJU dataset. Then, to test their robustness to loose clothing and challenging poses, we evaluate on the DynaCap dataset. We used SMPLX tracking from twenty cameras and trained their method in the same setting as ours. We observe while they can do replay and novel view synthesis with high quality if we train only on small sequences (300 frames), their result becomes blurry when trained on more frames (see Fig. 6). We hypothesize that this is caused by the fact that their volume primitives have to model fine-scale deformations and appearance at the same time, while the capacity of the network is too limited to model both of those aspects of the human. Additionally, we observe that the model fails to converge on loose clothing, as their volume regularizer prevents primitives from moving far away from the SMPLX initialization, which is imperative in the case of loose clothing (see Fig. 12). Ours, in contrast, deals with deformations separately in the explicit character model, which allows us to maintain appearance quality even as the number of training frames increases.

J. Limitations and Future Work

Though our work is a clear step towards more immersive and photorealistic avatars, there are remaining challenges yet to be addressed in the future. For example, our method, similar to other prior works, does not allow to model topological changes, e.g. opening a jacket. Future work could explore layered human representations, potentially able to model such effects. Flickering artifacts occur in our method, due to inconsistent color calibration of the multi-view cameras. We believe a joint optimization of camera parameters, i.e. color, extrinsic, and intrinsic calibration, can potentially resolve this limitation. Additionally, tracking errors in the case of fast motions, e.g. jumping, may result in artifacts in the renderings. Tightly entangling the tracking and rendering might resolve this in the future. Moreover, we currently require a dense studio setup to acquire the photoreal avatar. In the future, we plan to explore more lightweight setups even for training the model. Last, our projective texturing takes the learned geometry from the deformable character model as input while not being able to further refine it throughout the training. We believe differentiable projective texturing could be an interesting direction to tackle this problem.

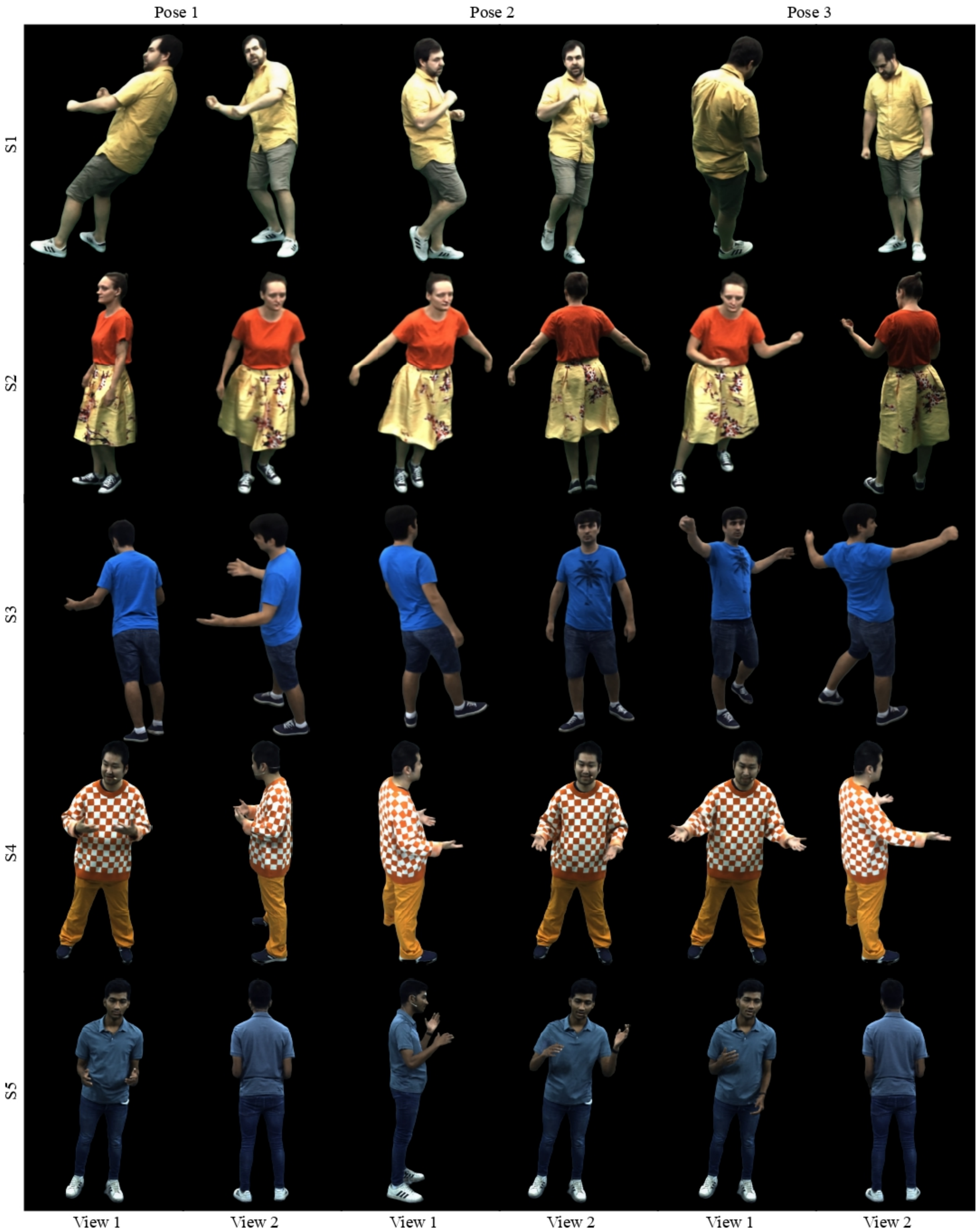

Figure 7. **Results from our method in the *novel view synthesis* setting for five different subjects**. Our method faithfully reconstructs fine details such as wrinkles (S1, S3), loose clothes with large deformations (S2), rich textures (S4), and hand poses.

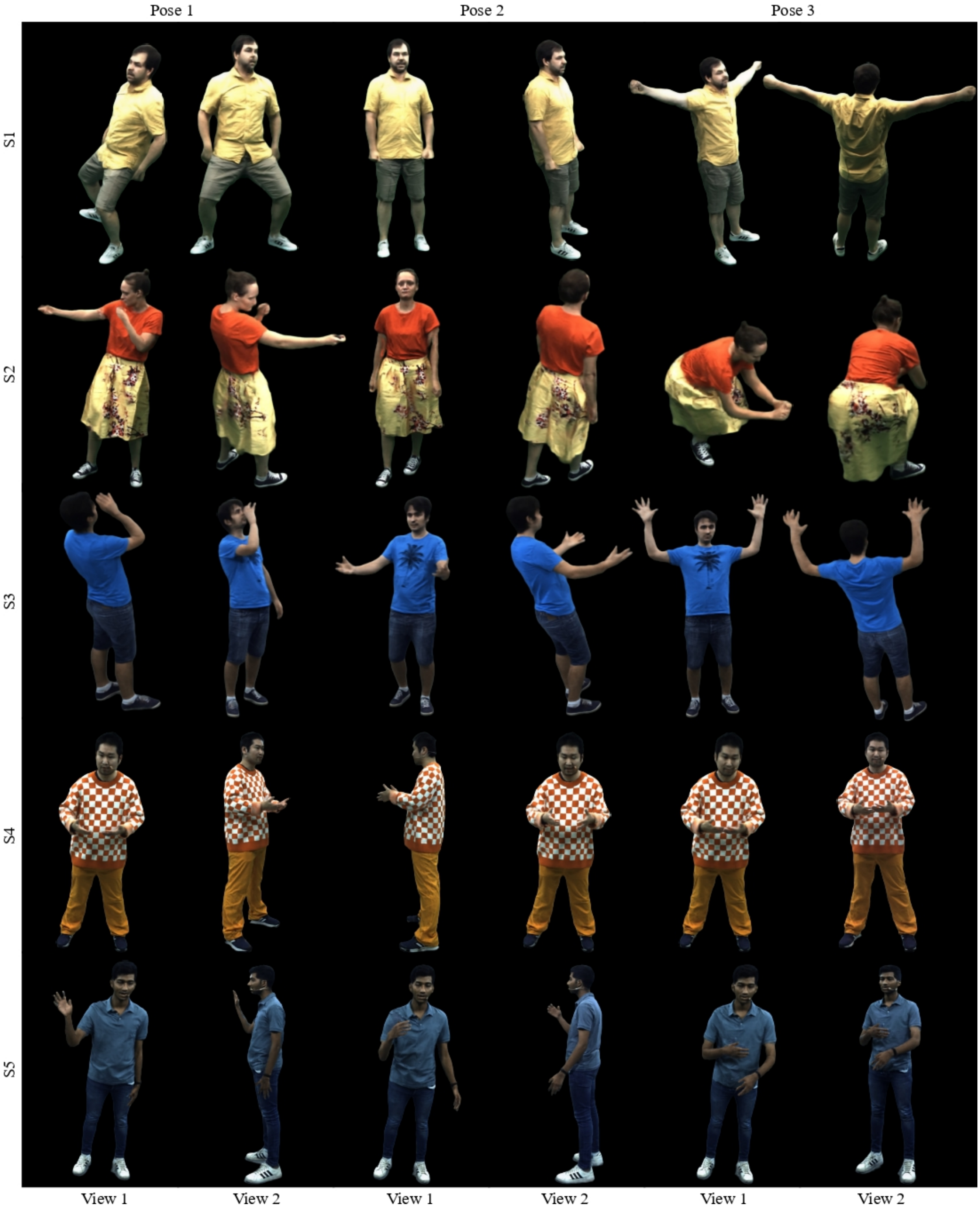

Figure 8. **Results from our method in the *novel pose synthesis* setting for five different subjects**. The results are rendered from a novel view point not seen during training. Our method results in impressive renderings, providing very realistic wrinkle patterns and high-frequency details.

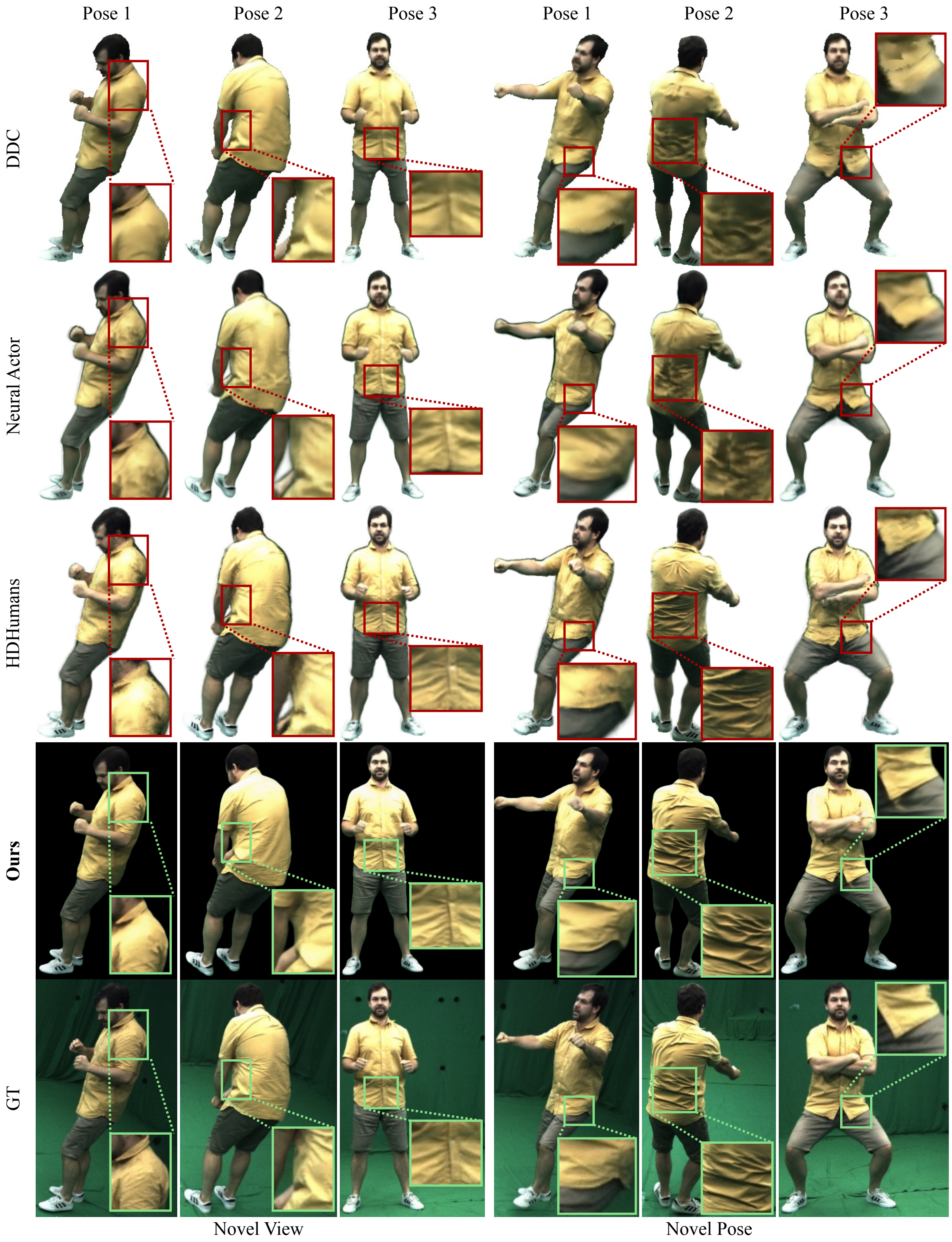

Figure 9. **Comparison of our method with previous pose-driven approaches**. Note how DDC [3] and Neural Actor (NA) [10] fail to produce high-frequency details and cloth wrinkles, while our method produces high-quality renderings for both novel view and novel pose settings.

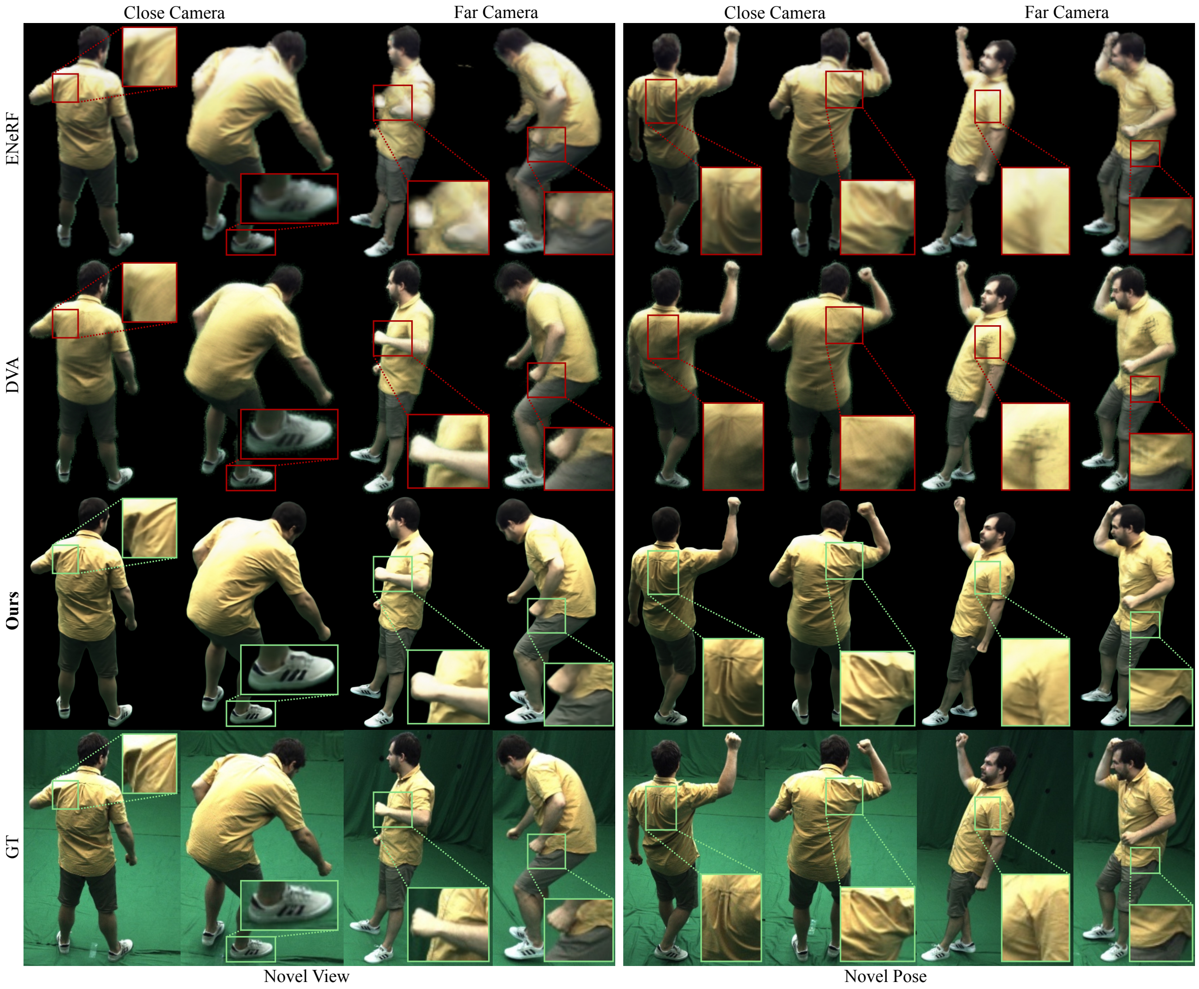

Figure 10. **Comparison of our method with existing real-time approaches that take images as input**. Note how ENeRF [8] produces artifacts under novel camera views and how DVA [13] suffers from inaccurate tracking, resulting in blurry renderings. On the contrary, our method generalizes well to far camera viewpoints and produces sharp results.

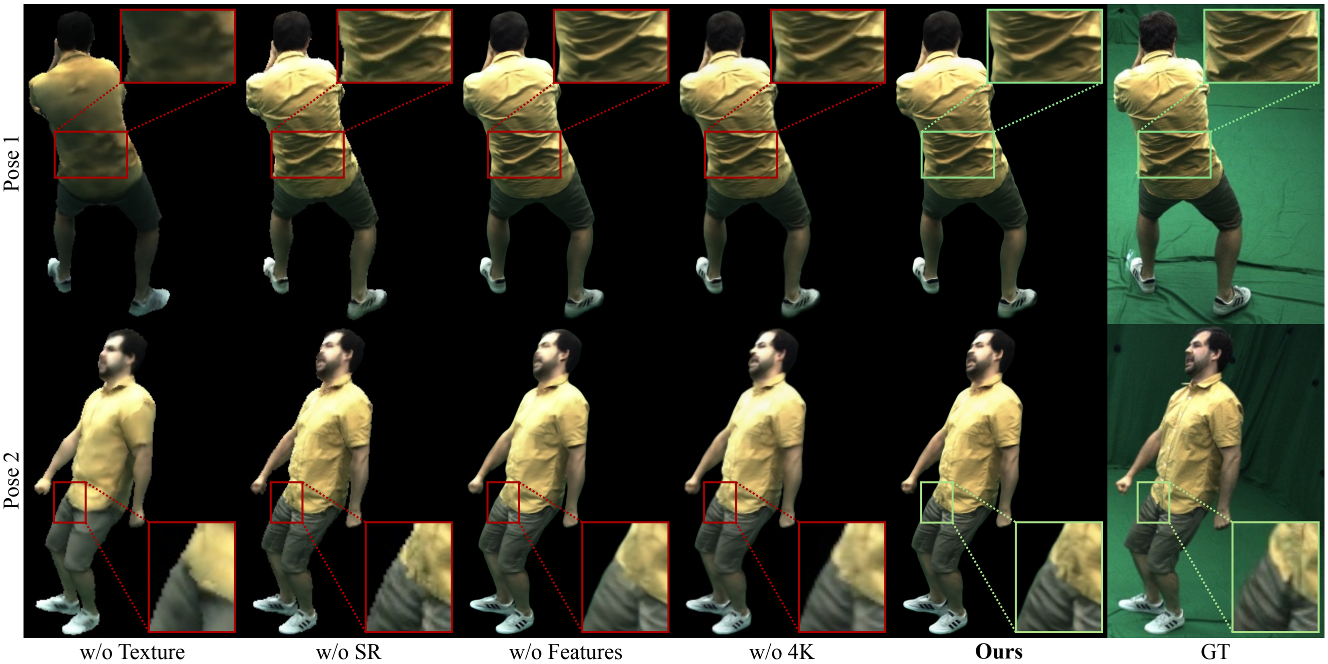

Figure 11. Qualitative results considering the contributions of the components from our method. Without texture, results fail to reproduce fine details. Without the SR module and the features as additional input further degrades the borders and the high-frequency details. Finally, without 4K training, the results are less sharp compared to our method.

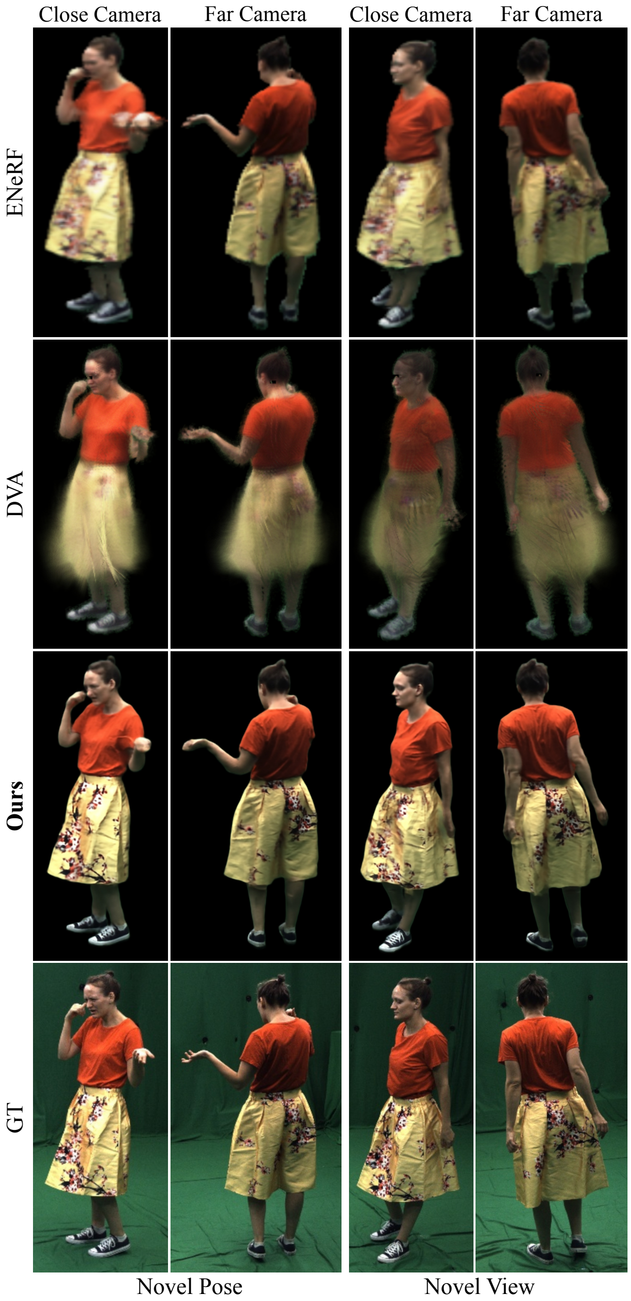

Figure 12. **Comparison to ENeRF and DVA on Loose Clothing**. ENeRF [8] can handle loose clothing reasonably well, but suffers from artifacts in other body parts (like in the hands). DVA [13] fails to converge on loose-clothed subjects, leading to artifacts in areas like the skirt. In contrast, as our character model can handle loose clothing, we produce sharp results with details preserved.

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, Dan 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. Software available from tensorflow.org. 2

[2] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:1, 2020. 1

[3] 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(4):1–16, 2021. 1, 7

[4] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. A deeper look into deepcap. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–1. IEEE, 2021. 1

[5] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46. ACM, 2007. 1

[6] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014. 2

[7] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution, 2017. 2

[8] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields with learned depth-guided sampling. In *SIGGRAPH Asia Conference Proceedings*, 2022. 3, 8, 10

[9] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8762–8771, 2021. 2

[10] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.*, 40(6), 2021. 7

[11] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10975–10985. Computer Vision Foundation / IEEE, 2019. 4

[12] 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*, 1(1): 9054–9063, 2021. 4

[13] Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, et al. Drivable volumetric avatars using texel-aligned features. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 3, 8, 10

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, pages 234–241. Springer, 2015. 2, 3

[15] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3), 2007. 1

[16] TheCaptury. The Captury. `http://www.thecaptury.com/`, 2020. 2

[17] Treedys. Treedys. `https://www.treedys.com/`, 2020. 1

[18] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *International Conference on Computer Vision (ICCV)*, pages 3295–3306, 2023. 2