SUPPLEMENTARY MATERIALS OF EFFICIENT MESHY NEURAL FIELDS FOR ANIMATABLE HUMAN AVATARS

Anonymous authors
Paper under double-blind review

Thank you for reading our supplementary materials! Here we provide in-depth descriptions of our method, including details about:

1. Loss functions (Sec. A).
2. Image-based lighting (Sec. B).
3. Image-based lighting (Sec. C).
4. Ablations (Sec. D).
5. Additional experimental results (Sec. E and Sec. H).
6. Challenges in ZJU Mocap dataset (Sec. F).
7. Application examples in Sec. G.
8. Limitations and social impact in Sec. I.

We strongly encourage our readers to view the supplemental video for a more comprehensive visual perception.

A LOSS FUNCTIONS

Our loss function $L = L_{\text{img}} + L_{\text{mask}} + L_{\text{reg}}$ is composed of three parts: an image loss L_{img} using ℓ_1 norm on tone mapped colors, and mask loss L_{mask} using squared ℓ_2, and regularization losses L_{reg} to improve the quality of canonical geometry, materials, lights, and motion.

Image loss: our renderer utilizes physically-based shading to produce high-dynamic range (HDR) images. Then the complex materials and environmental lights are elaborately optimized. Thus our loss function requires a full range of floating point values. We follow (Hasselgren et al., 2021; Munkberg et al., 2022; Hasselgren et al., 2022) to compute ℓ_1 norm on tone mapped colors. Specifically, we first transform linear radiance values i according to a tone-mapping operator $T(i) = \Gamma(\log(i+1))$, in which $\Gamma(i)$ is a linear RGB to sRGB transformation function (Stokes et al., 1996):

$$\begin{aligned}\Gamma(i) &= \begin{cases}12.92i & i \leq 0.0031308 \\ (1+a)i^{1/2.4} - a & i > 0.0031308\end{cases} \\ a &= 0.055,\end{aligned} \tag{1}$$

Mask loss: The renderer (Laine et al., 2020) renders both the shaded images and the corresponding rasterization masks in a differentiable manner. Therefore, we compute the ℓ_2 norm between the masks and the preprocessed mattings (in both ZJU-MoCap and H36M benchmarks, we use the provided preprocessed subject masks from (Peng et al., 2021b; Gong et al., 2018)), akin to the traditional shape-from-silhouette (Ma et al., 2004) technique. The mask loss is parallel with the image loss, yet facilitates the course of shading optimization by making shape convergence super fast in about a hundred training steps.

Regularizers: We need various priors to encourage the optimization to converge at a place where the geometry, materials, and lighting are well separated and smooth enough (Munkberg et al., 2022; Hasselgren et al., 2022). Therefore, we choose to minimize regularization during training.

We introduce smoothness to PBR materials in terms of albedo $\mathbf{k}_d$, specular parameters $\mathbf{k}_{\text{orm}}$, and surface geometry nomral $\mathbf{n}$ as following:

$$L_{\mathbf{k}} = \frac{1}{|\mathbf{x}_{\text{surf}}|} \sum_{\mathbf{x}_{\text{surf}}} \left|\mathbf{k}\left(\mathbf{x}_{\text{surf}}\right) - \mathbf{k}\left(\mathbf{x}_{\text{surf}} + \epsilon\right)\right|, \tag{2}$$

where $|\mathbf{x}_{\text{surf}}|$ is a surface point on the surface in canonical space and $\epsilon \sim \mathcal{N}(0, \sigma{=}0.01)$ is a small random offset. We regularize the geometry normal on the surface of the canonical mesh derived from the SDF field for a seek of a smoother surface and avoidance of holes in the surface.

We regularize light by assuming the neutral spectrum in the real world. Specifically, given the per-channel average radiance densities $\bar{c_i}$, we penalize the color shifts as:

$$L_{\text{light}} = \frac{1}{3}\sum_{i=0}^{3}\left| c_i - \frac{1}{3}\sum_{i=0}^{3} c_i \right|, \tag{3}$$

To encourage a watertight surface and reduce floating meshes both inside and outside the subject, we impose the regularization of binary cross-entropy loss H on the SDF field as:

$$\begin{aligned} L_{\text{sdf}} = \sum_{i,j \in \mathrm{S}_e} & H\left(\sigma\left(s_i\right), \operatorname{sign}\left(s_j\right)\right) \\ & + H\left(\sigma\left(s_j\right), \operatorname{sign}\left(s_i\right)\right), \end{aligned} \tag{4}$$

where S_e is the set of all vertex along their edges in which the signs of the SDF values are different (*i.e.*, $\operatorname{sign}(s_i) \neq \operatorname{sign}(s_j)$). To remove the floating meshes outside the surface, we impose an additional loss. For a triangle surface f extracted by marching tetrahedra, if f is invisible, we encourage its SDF values to be positive as:

$$L_{\text{invis}} = \sum_{i \in \mathrm{S}_{\text{invis}}} H(\sigma\left(s_i\right), 1). \tag{5}$$

We weigh the above terms and use the loss for all our experiments:

$$\begin{aligned} L = {} & L_{\text{image}} + L_{\text{mask}} \\ & + \underbrace{\lambda_{\mathbf{k}_{\text{d}}}}_{=0.03} L_{\mathbf{k}_{\text{d}}} + \underbrace{\lambda_{\mathbf{k}_{\text{orm}}}}_{=0.05} L_{\mathbf{k}_{\text{orm}}} + \underbrace{\lambda_{\mathbf{n}}}_{=0.025} L_{\mathbf{n}} \\ & + \underbrace{\lambda_{\text{light}}}_{=0.005} L_{\text{light}} + \underbrace{\lambda_{\text{sdf}}}_{=0.02} L_{\text{sdf}} + \underbrace{\lambda_{\text{invis}}}_{=0.01} L_{\text{invis}}. \end{aligned} \tag{6}$$

B IMAGE-BASED LIGHTING

The split sum shading model is widely used in real-time rendering (Möller et al., 2008), giving both realism and efficiency against spherical Gaussians (SG) and spherical harmonics (SH) (Boss et al., 2021; Chen et al., 2019; Zhang et al., 2021a). We use a differentiable split sum (Karis, 2013) shading model to approximate rendering equation (Kajiya, 1986) for image-based environment light learning as (Munkberg et al., 2022):

$$\begin{aligned} L\left(\omega_o\right) \approx & \int_{\Omega} f\left(\omega_i, \omega_o\right)\left(\omega_i \cdot \mathbf{n}\right) d\omega_i \\ & \int_{\Omega} L_i\left(\omega_i\right) D\left(\omega_i, \omega_o\right)\left(\omega_i \cdot \mathbf{n}\right) d\omega_i. \end{aligned} \tag{7}$$

where D is the GGX normal distribution function (NDF) (Walter et al., 2007) in a Cook-Torrance microfacet specular shading model (Cook & Torrance, 1982). The first term contributes to the specular BSDF *wrt.* a solid white environment light, which depends solely on the roughness r of the BSDF and the light-surface angles $\cos\theta = \omega_i \cdot \mathbf{n}$. The second term contributes to the integral of the incoming radiance with the GGX normal distribution function, D. Both terms can be pre-computed and filtered to reduce computation (Karis, 2013).

The training parameters are texels of a cube light map whose resolution is $6 \times 512 \times 512$. The pre-integrated lighting for the least roughness values is derived from the base level, and multiple smaller mip levels are constructed from it (Karis, 2013). Each mip-map is filtered by average-pooling the base level of the current resolution and is convolved with the GGX normal distribution function. The per mip-level filter bounds are pre-computed as well. We leverage a PyTorch implementation with

CUDA extensions from (Munkberg et al., 2022). Moreover, a cube map is created to represent the diffuse lighting in a low resolution, akin to the filtered specular probe. It shares the same optimizable parameters and is average-pooled to the mip level with $r = 1$ roughness. The pre-filtering only involves the first term in Eq. 7.

C IMPLEMENTATION DETAILS

SDF network. We parametrize the SDF field with an MLP to increase surface water-tight and smoothness. We choose the MLP architecture from (Mildenhall et al., 2022), which consists of 6 frequency bands for positional encoding, and 8 linear layers, each having 256 neurons, followed by ReLU activations. We implicitly regularize the smoothness by increasing the Lipschitz property in the SDF field(Liu et al., 2022).

Material network. The material model is a small MLP with hash-encoding (Müller et al., 2022) as the materials query is computationally extensive. The MLP consists of two linear layers, each having 32 neurons, followed by ReLU activations. The hash-encoding has a spatial resolution of 4096 and the rest configures are the same as (Munkberg et al., 2022). To reduce computation, we predict all material channels at once with one backbone network. Besides, we introduce inductive bias of materials of clothed humans in the real world, by providing minimum and maximum values for each materials channel. We follow (Zhang et al., 2021b) to limit the albedo $\mathbf{k}_d \in [0.03, 0.8]$, and the roughness $\mathbf{k}_r \in [0.08, 1]$. The texels in the environment light are randomly initialized between $[0.25, 0.75]$.

Motion networks. For the motion module, we use the same MLP architecture as (Chen et al., 2021; Wang et al., 2022), which is similar to our SDF MLP. To resolve the problem where the training pose variation is too limited for skinning field learning (*e.g.*, self-rotation video without any limbs movements), we initialize the MLP with the pre-trained skinning model provided by (Wang et al., 2022), and impose ℓ_2 norm for the skinning weights logits between our predictions and the ground truth from SMPL (Loper et al., 2015). We ablate the design choices in Sec. D. For the non-rigid modeling, we use another 4-layer ReLU MLP with a 4-frequency-band positional encoding. We also progressively anneal its encoding for 5k iterations as (Park et al., 2021). The weights of the last layer are initialized with a uniform distribution $\mathcal{U}(-10^{-5}, 10^{-5})$, *i.e.* initializing the non-rigid offsets to be close to zero and not interfering with the major optimizations of geometry and materials.

Optimization. We use Adam (Kingma & Ba, 2015) as our default optimizer. We optimize the subject for 5k steps for 1024×1024 images or 10k steps 512×512 images. We disable the perturbed normal map during optimization as it leads to SDF collapsing abruptly at a certain step (*i.e.*, all SDF values are positive or negative where marching tetrahedra fails). The optimization process takes about an hour on a single NVIDIA GTX3090 GPU. The indicative results with plausible quality appear after a few minutes, which is quite faster than our counterparts (Peng et al., 2021b;a; Wang et al., 2022; Xu et al., 2022). Such superior efficiency could largely accelerate downstream applications. The training visualization is presented in the supplemental video.

Tetrahedra grids. We start with a tetrahedra grid with 128×128 resolution, including 192k tetrahedra and 37k vertices. Each tetrahedron can produce at most 2 triangles by marching tetrahedra algorithm (Munkberg et al., 2022; Shen et al., 2021; Gao et al., 2020). To increase the resolution of the tetrahedra grid, we subdivide the grid at the 500th step. To avoid the out-of-memory problem caused by the vast amount of floating meshes in the void space at the beginning of training, we pre-train the SDF network to **match a visual hull** of humans in canonical space. The hull could be derived from either the skeleton capsules or the SMPL (Loper et al., 2015) mesh. Note that we only pre-train for 500 iterations, which leads to **a very coarse shape** akin to the visual hull rather than the given ground truth mesh. The initialized mesh is presented in the training visualization part of the supplemental video.

D ABLATIONS

The parametrization type for SDF field. The SDF fields can either be parameterized as either MLPs or value fields. Table 1 and Figure 1 show that using MLP to predict SDF values results in a smoother mesh surface that is watertight. MLP offers extrapolation ability to predict invisible parts

Table 1: **The ablation on each module from our method**. The mesh tends to be noisy and poor for rendering novel poses without MLP parametrization for the geometry module; Removing the non-rigid module harms the convergence of our model due to the disability to solve multi-view inconsistency; PBR materials improve the overall shading quality by joint modeling both decomposed materials and lighting.

	Training Pose		Novel Pose	
	PSNR	SSIM	PSNR	SSIM
w/o SDF MLP	25.17	0.913	23.37	0.874
w/o Non-rigid	25.03	0.909	23.45	0.877
w/o PBR	25.10	0.914	23.44	0.878
w/o Specular	25.24	0.915	**23.58**	**0.879**
Full	**25.26**	**0.916**	23.52	**0.879**

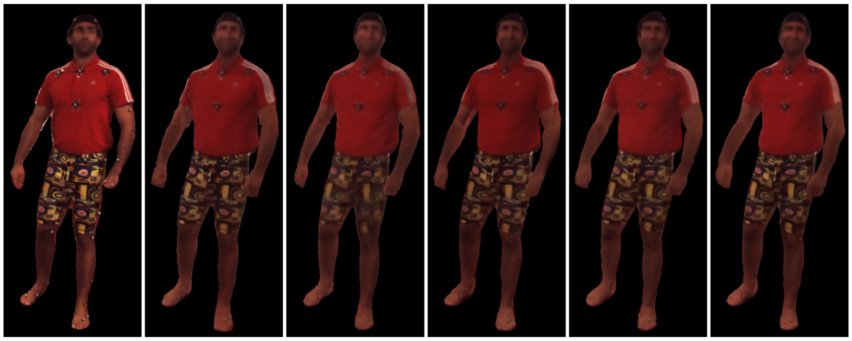


Figure 1: **Qualitative ablation on each module**. The SDF MLP improves the mesh smoothness; non-rigid modeling proves the texture quality by solving the multi-view consistency of cloth dynamics; The PBR materials have a larger capacity for modeling complex materials and lighting against the only-RGB and the no-specular counterparts, which further facilitates both mesh and material learning.

and keep the mesh watertight. While directly optimizing SDF value fields leads to a jiggling mesh surface and holes in invisible parts during training (*e.g.*, underarm).

The shading model type in geometry module. We compare PBR shading models with directly predicting RGB colors and PBR without shading specular. Table 1 and Figure 1 show that PBR shading models lead to higher metrics against RGB predications, which indicates that PBR materials can better model complex textures and lights for dynamic humans. Removing the specular term in PBR does not affect the performance much. We conjecture that there is less specularity in human skin and clothes materials.

The impact of the non-rigid net in motion module. As shown in Table 1 and Figures 1, modeling pose-dependent non-rigid dynamics of clothes improves the overall reconstruction quality. It facilitates the aggregation of shading information for multi-view inputs during training.

The impact of human tracking quality. Table 2 (a) and Figure 2 show that using marker-based pose-tracking data can give better results. The same phenomenon has been stated in (Peng et al., 2021a). Noisy marker-less pose-tracking harms the optimization process by damaging the multi-view consistency and the exact pose for shading optimization, which leads to blurry textures.

The impact of training view amount. Table 2 (b) and Figure 3 reveal that giving one camera of view degrades the overall reconstruction quality, and multi-view consistency improves the final results. The model can aggregate multi-view information for better shading optimization, thus leading to clearer surface materials.

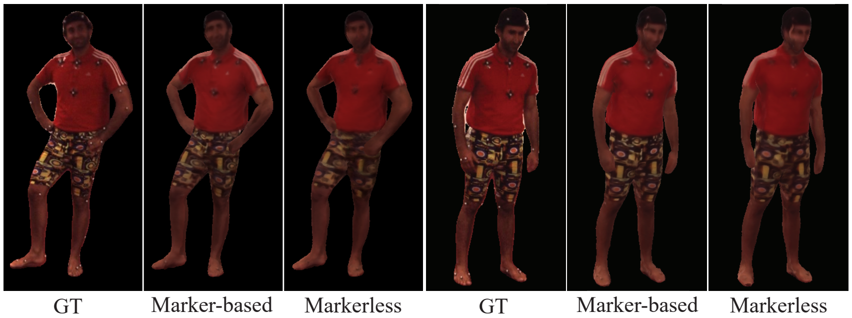


Figure 2: **Qualitative results of models trained on poses** from marker-less and marker-based systems.

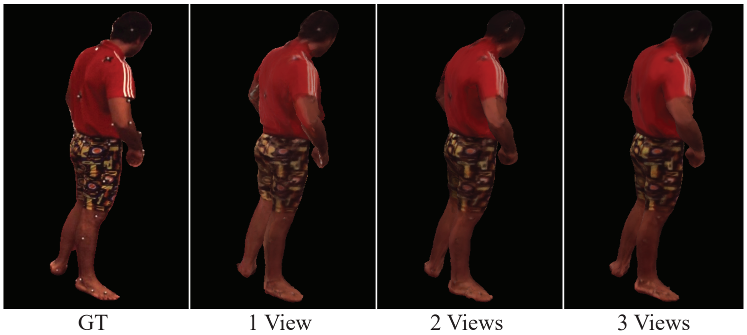


Figure 3: **Comparison of models trained with different numbers of camera views** on the subject "S9".

The impact of training frame amount. As the number of training frames increases, the rendering quality on novel view and novel pose increases as well (Table 2 (c) and Figure 4). Notice that the reconstruction quality saturated after using a certain amount of training frames, the same results can be observed in (Peng et al., 2021a) as well.

Number of Training Views. Table 3 and Figure 5 show that giving one camera of view degrades the overall reconstruction quality, and multi-view consistency improves the final results. The model can aggregate multi-view information for better shading optimization, thus leading to clearer surface materials.

The effect of Skinning Module Design Table4 and Figure8 reveal that the initialization with a pre-trained skinning net and the regularization on surface skinning improve the overall reconstruction quality. The initialization provides skinning prior which helps to speed up geometry convergence. From Figure 6-7, the geometry details improve with the initialization under the same training time.

The regularization on surface skinning prevents geometry degradation. Figure 8 indicates that our model can not learn correct canonical geometry without initialization and regularization. The mesh distortion is reduced with the regularization.

Effect of SDF Network The MLP parametrization of the SDF field keeps our surface both watertight and smooth, as shown in Figure 9.

E MORE COMPARISONS

We present full quantitative comparisons in Table 7, Table 5, Table 8, and Table 6.

We present the visual comparisons with Neuman (Jiang et al., 2022) and HumanNeRF (Weng et al., 2022). Notably, Neuman is designed for monocular video, so the comparison is just for reference.

Table 2: **The ablations results on data quality and quantity** on H36M S9 subject, in terms of PSNR and SSIM (higher is better). The better the data quality, the better the reconstruction results.

	Training pose		Novel pose	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
(a) type of pose tracking				
w/o marker	24.73	0.893	22.60	0.853
w/ marker	**25.53**	**0.911**	**23.80**	**0.879**
(b) number of training views				
1 view	25.09	0.906	22.97	0.866
2 views	25.56	**0.911**	**23.76**	**0.878**
3 views	**25.57**	**0.911**	23.67	0.876
(c) number of training frames				
1 frame	20.93	0.817	19.58	0.785
100 frames	23.99	0.882	22.49	0.856
200 frames	**25.27**	**0.905**	**23.32**	**0.873**
800 frames	24.89	0.900	23.16	**0.873**

Table 3: **Ablation results of training views on the ZJU-MoCap 313 subject**.

	Training pose		Novel pose	
ZJU-MoCap 313	PSNR↑	SSIM↑	PSNR↑	SSIM↑
1 view	24.39	0.913	21.45	0.869
2 views	28.06	0.945	22.81	0.888
3 views	28.50	0.956	23.17	0.894
4 views	**29.04**	**0.961**	**23.20**	**0.896**

Table 4: **The ablation on the skinning module of ZJU-MoCap 313 dataset**.

	Training Pose		Novel Pose	
ZJU-MoCap 313	PSNR	SSIM	PSNR	SSIM
w/o skinning init. & reg.	27.46	0.949	20.31	0.831
w/ skinning initialization	28.82	0.958	23.08	0.893
w/ skinning regularization	28.80	0.959	23.14	0.895
Full	**29.05**	**0.961**	**23.27**	**0.897**

Table 5: **Quantitative results of training pose novel view synthesis of H36M dataset**.

	Training pose									
	PSNR					SSIM				
	NB	SA-NeRF	Ani-NeRF	ARAH	Ours	NB	SA-NeRF	Ani-NeRF	ARAH	Ours
S1	22.87	23.71	22.05	24.45	24.56	0.897	0.915	0.888	0.919	0.919
S5	24.60	24.78	23.27	24.54	24.51	0.917	0.909	0.892	0.918	0.920
S6	22.82	23.22	21.13	24.61	24.55	0.888	0.881	0.854	0.903	0.902
S7	23.17	22.59	22.50	24.31	24.05	0.914	0.905	0.890	0.919	0.916
S8	21.72	24.55	22.75	24.02	23.94	0.894	0.922	0.898	0.921	0.920
S9	24.28	25.31	24.72	26.20	25.99	0.910	0.913	0.908	0.924	0.919
S11	23.70	25.83	24.55	25.43	25.48	0.896	0.917	0.902	0.921	0.915
Average	23.31	24.28	23.00	24.79	24.72	0.902	0.909	0.890	0.918	0.916

Table 6: **Quantitative results of unseen pose novel view synthesis of H36M dataset**.

	Unseen pose									
	PSNR					SSIM				
	NB	SA-NeRF	Ani-NeRF	ARAH	Ours	NB	SA-NeRF	Ani-NeRF	ARAH	Ours
S1	21.93	22.67	19.96	23.08	23.72	0.873	0.890	0.855	0.899	0.904
S5	23.33	23.27	20.02	22.79	23.13	0.893	0.881	0.840	0.890	0.898
S6	23.26	23.23	23.64	24.04	24.17	0.888	0.888	0.882	0.900	0.903
S7	22.40	22.51	21.76	22.58	22.72	0.888	0.898	0.869	0.891	0.889
S8	20.78	23.06	21.63	22.34	22.71	0.872	0.904	0.877	0.896	0.902
S9	22.87	23.84	21.95	24.36	24.54	0.880	0.889	0.871	0.894	0.895
S11	23.54	24.19	22.55	24.78	24.47	0.879	0.891	0.875	0.902	0.900
Average	22.59	23.25	21.64	23.42	23.64	0.882	0.892	0.867	0.896	0.899

Table 7: **Quantitative results of training pose novel view synthesis of ZJU-MoCap dataset**.

	Training pose									
	PSNR					SSIM				
	NB	SA-NeRF	Ani-NeRF	ARAH	Ours	NB	SA-NeRF	Ani-NeRF	ARAH	Ours
Twirl(313)	30.56	31.32	29.80	31.60	29.67	0.971	0.974	0.963	0.973	0.947
Taichi(315)	27.24	27.25	23.10	27.00	24.21	0.962	0.962	0.917	0.965	0.919
Swing1(392)	29.44	29.29	28.00	29.50	27.58	0.946	0.946	0.931	0.948	0.899
Swing2(393)	28.44	28.76	26.10	27.70	25.91	0.940	0.941	0.916	0.940	0.890
Swing3(394)	27.58	27.50	27.50	28.90	27.67	0.939	0.938	0.924	0.945	0.902
Warmup(377)	27.64	27.67	24.20	27.80	26.69	0.951	0.954	0.925	0.956	0.926
Punch1(386)	28.60	28.81	25.60	29.20	27.65	0.931	0.931	0.878	0.934	0.881
Punch2(387)	25.79	26.08	25.40	27.00	25.68	0.928	0.929	0.926	0.945	0.908
Kick(390)	27.59	27.77	26.00	27.90	24.08	0.926	0.927	0.912	0.929	0.840
Average	28.10	26.19	28.27	28.51	26.57	0.944	0.945	0.921	0.948	0.901

Table 8: **Quantitative results of unseen pose novel view synthesis of ZJU-MoCap dataset**.

	Unseen pose									
	PSNR					SSIM				
	NB	SA-NeRF	Ani-NeRF	ARAH	Ours	NB	SA-NeRF	Ani-NeRF	ARAH	Ours
Twirl(313)	23.95	24.33	22.80	24.40	23.63	0.905	0.908	0.863	0.914	0.878
Taichi(315)	19.56	19.87	18.47	20.00	20.42	0.852	0.863	0.795	0.881	0.850
Swing1(392)	25.76	26.27	18.44	26.20	25.49	0.909	0.927	0.670	0.927	0.883
Swing2(393)	23.80	24.96	21.87	24.40	24.31	0.878	0.900	0.836	0.915	0.883
Swing3(394)	23.25	24.24	17.69	25.20	24.72	0.893	0.908	0.792	0.908	0.870
Warmup(377)	23.91	25.34	23.28	25.50	24.80	0.909	0.928	0.901	0.933	0.894
Punch1(386)	25.68	27.30	25.55	27.00	26.24	0.881	0.905	0.872	0.910	0.853
Punch2(387)	21.60	23.08	21.92	24.20	24.06	0.870	0.890	0.838	0.917	0.889
Kick(390)	23.90	24.43	23.90	24.80	25.79	0.870	0.889	0.887	0.896	0.873
Average	23.49	24.42	21.55	24.63	24.38	0.885	0.902	0.828	0.911	0.875

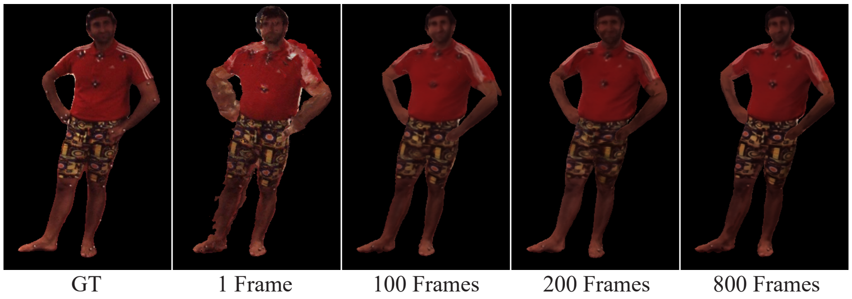

GT 1 Frame 100 Frames 200 Frames 800 Frames

Figure 4: **Comparison of models trained with different numbers of video frames** on the subject "S9".

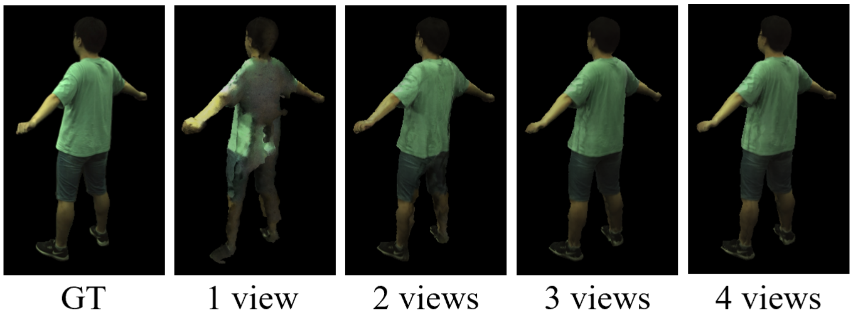

GT 1 view 2 views 3 views 4 views

Figure 5: **Ablation study of training views on the ZJU-MoCap 313 subject.**

F CHALLENGES IN ZJU-MOCAP DATASET

We found that the challenges in ZJU-Mocap datasets impede our methods to get better quantitative performance. Li et al. (2022) mentions this in their paper and here we refer to their findings in Figure 11.

The variation of exposures in cameras breaks our assumption of constant lighting, which hurts the performance.

We also compare the ground truths and our renderings side by side to demonstrate the problem in Figure 12 and Figure 13. Even the successive cameras have different exposures. While our method renders images with the same exposure due to our **constant lighting assumption**.

G APPLICATIONS

We showcase **relighting**, **texture editing**, and **novel poses synthesis** on AIST dataset (Li et al., 2021a) in Figure 14, Figure 15, and Figure 16 separately. All the above applications are presented in the supplemental video.

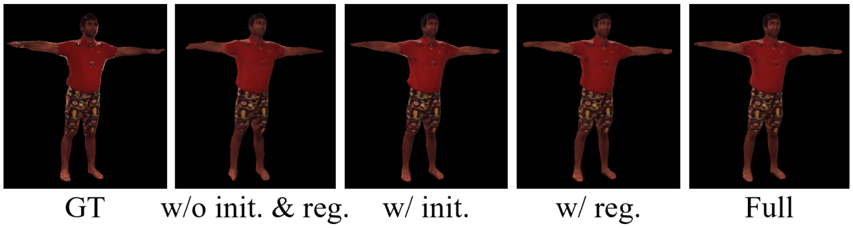
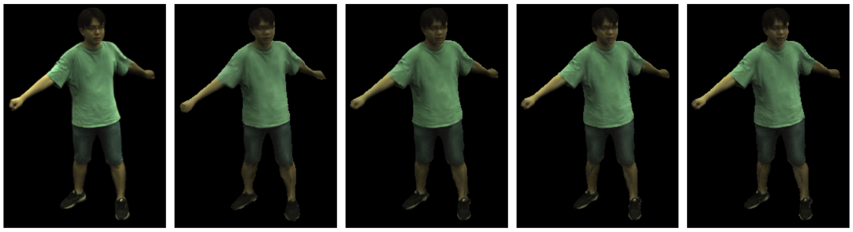
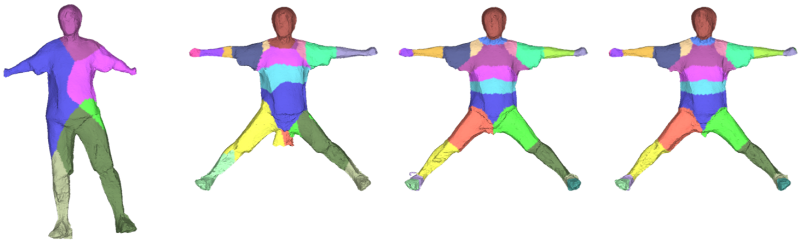

GT w/o init. & reg. w/ init. w/ reg. Full

Figure 6: **Ablation study of the skinning module on the H36M S9 subject.**



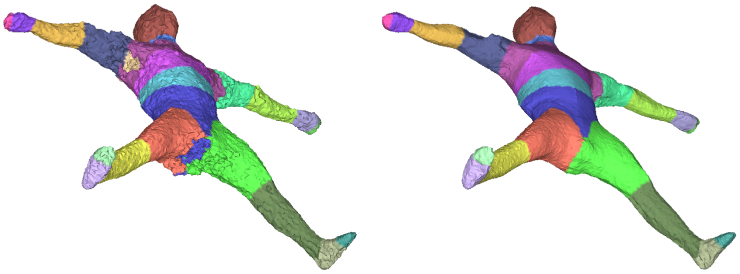
GT w/o init. & reg. w/ init. w/ reg. Full

Figure 7: **Ablation study of the skinning module on the ZJU-MoCap 313 subject.**



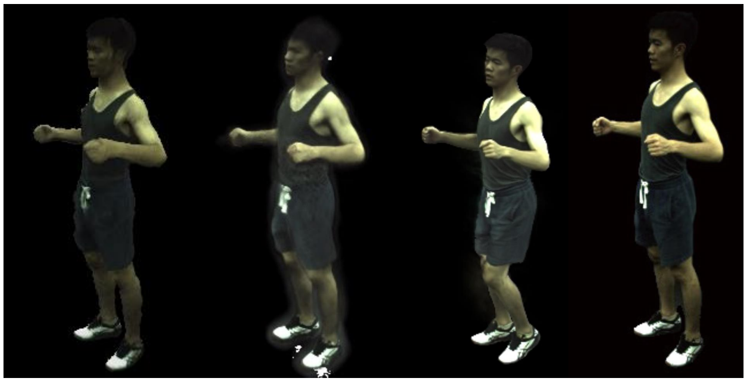
w/o init. & reg. w/ init. w/ reg. Full

Figure 8: **Ablation study of the skinning module on the ZJU-MoCap 313 subject.**



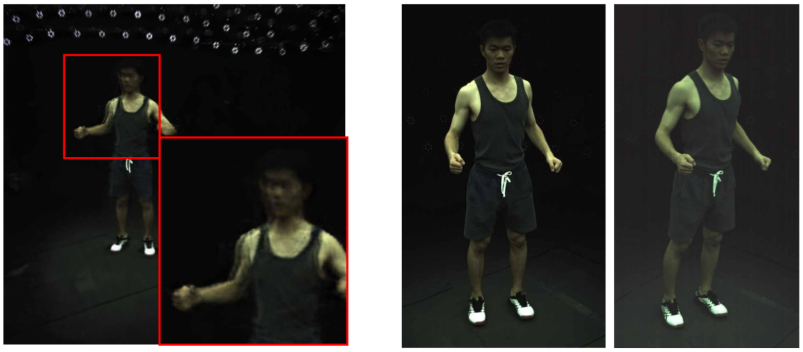
w/o SDF network w/ SDF network

Figure 9: **Ablation study of SDF field parametrization.**



Ours Neuman HumanNeRF Reference

Figure 10: **Qualitative comparison with Neuman and HumanNeRF.**



Imperfect Camera Calibrations

Various Camera Exposures

Figure 11: Challenges in ZJU-Mocap Dataset. Left: Train a NeRF (Mildenhall et al., 2022) in a frame with all cameras; Right: The variation of exposures in cameras breaks our assumption of constant lighting. Figure from (Li et al., 2022).

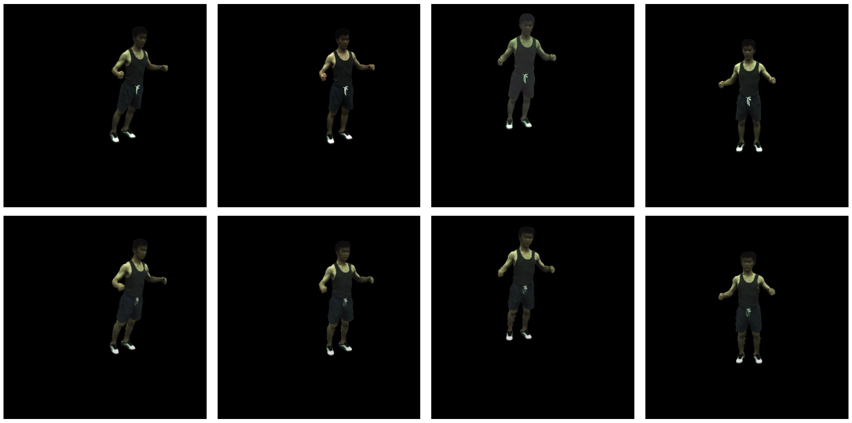


Figure 12: Compare the ground truths and our renderings. Upper: ground truths; Lower: our renderings. The successive cameras have different exposures, which breaks our **assumption of constant lighting**.

H MESH VISUALIZATIONS

We visualize the canonical mesh and present the number of faces of each mesh in Figure 17 and Figure 8. Note that the number of faces for each mesh is quite small. Though increasing the resolution of tetrahedra grids may improve the details of both geometry and materials, we do not conduct this experiment for it is orthogonal to our technical contributions.

I LIMITATIONS AND DISCUSSIONS

Our method leverage mesh as our core representation, which enables us efficiency for both training and rendering. However, the resolution of mesh is fixed in our pipeline, preventing fine-grained geometry and texture reconstruction. One possible solution could be tetrahedra grids subdivision (Schaefer et al., 2004; Gao et al., 2022; Kalischek et al., 2022). But it may break the SDF values around the derived meshes since there is no regularization over the whole SDF field. Our non-rigid modeling has less capacity, since we assume there is no topology change of mesh *wrt.* the non-rigid motion. Otherwise, we cannot query materials and motions in the canonical shape. One can solve it via the dense correspondence between the meshes before and after applying non-rigid motions (Ahmed et al., 2008; Zeng et al., 2020), yet such an operation may increase computation drastically.

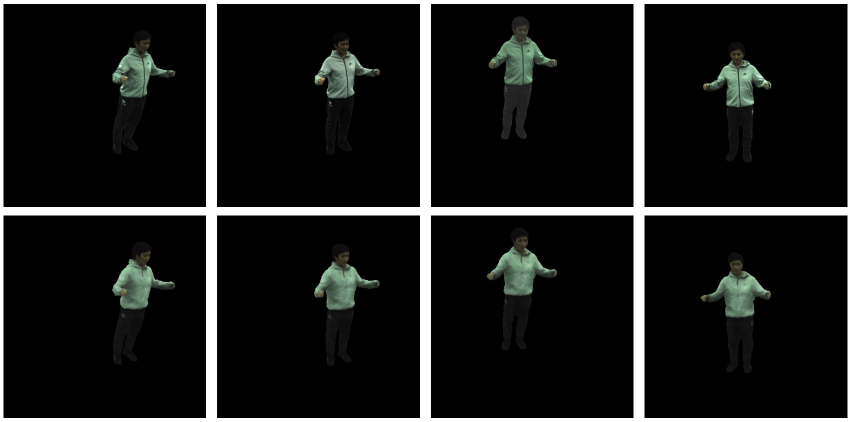


Figure 13: Compare the ground truths and our renderings. Upper: ground truths; Lower: our renderings. The successive cameras have different exposures, which breaks our **assumption of constant lighting**.

Our method needs foreground masks to facilitate mesh optimization, which is akin to shape-from-silhouette. One future direction might be equipping our method with the ability to separate foreground and background automatically (Jiang et al., 2022; Guo et al., 2023). It is also promising to model the background simultaneously during foreground subject optimization (Jiang et al., 2022; Guo et al., 2023), which eliminates the requirement of foreground mask processing.

Our method can digitize humans from visual footage, which may involve avatar misuse without the permission of the owners. Methods like implicit adversarial watermarks (Chen et al., 2020; Li et al., 2021b) that disable the neural nets inference could assist the video creation to protect their portrait rights. Another concern is the deep fake misuse (Nguyen et al., 2022), which corrupts the identity in the visual footage rendered by our model. Methods like deep fake detection (Pan et al., 2020) could help to discover and prevent deep fake creations. Besides, our method involves training with GPUs, which leads to carbon emissions and increasing global warming (Patterson et al., 2021).

REFERENCES

Naveed Ahmed, Christian Theobalt, Christian Rössl, Sebastian Thrun, and Hans-Peter Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. In *CVPR*, pp. 1–8, 2008.

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *ICCV*, pp. 12664–12674, 2021.

Lu Chen, Jiao Sun, and Wei Xu. FAWA: fast adversarial watermark attack on optical character recognition (OCR) systems. In *ECML*, pp. 547–563, 2020.

Wenzheng Chen, Huan Ling, Jun Gao, Edward J. Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, pp. 9605–9616, 2019.

Xu Chen, Yufeng Zheng, Michael J. Black, Otmar Hilliges, and Andreas Geiger. SNARF: differentiable forward skinning for animating non-rigid neural implicit shapes. In *ICCV*, pp. 11574–11584, 2021.

Robert L. Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, 1982.

Jun Gao, Wenzheng Chen, Tommy Xiang, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Learning deformable tetrahedral meshes for 3d reconstruction. In *NeurIPS*, 2020.

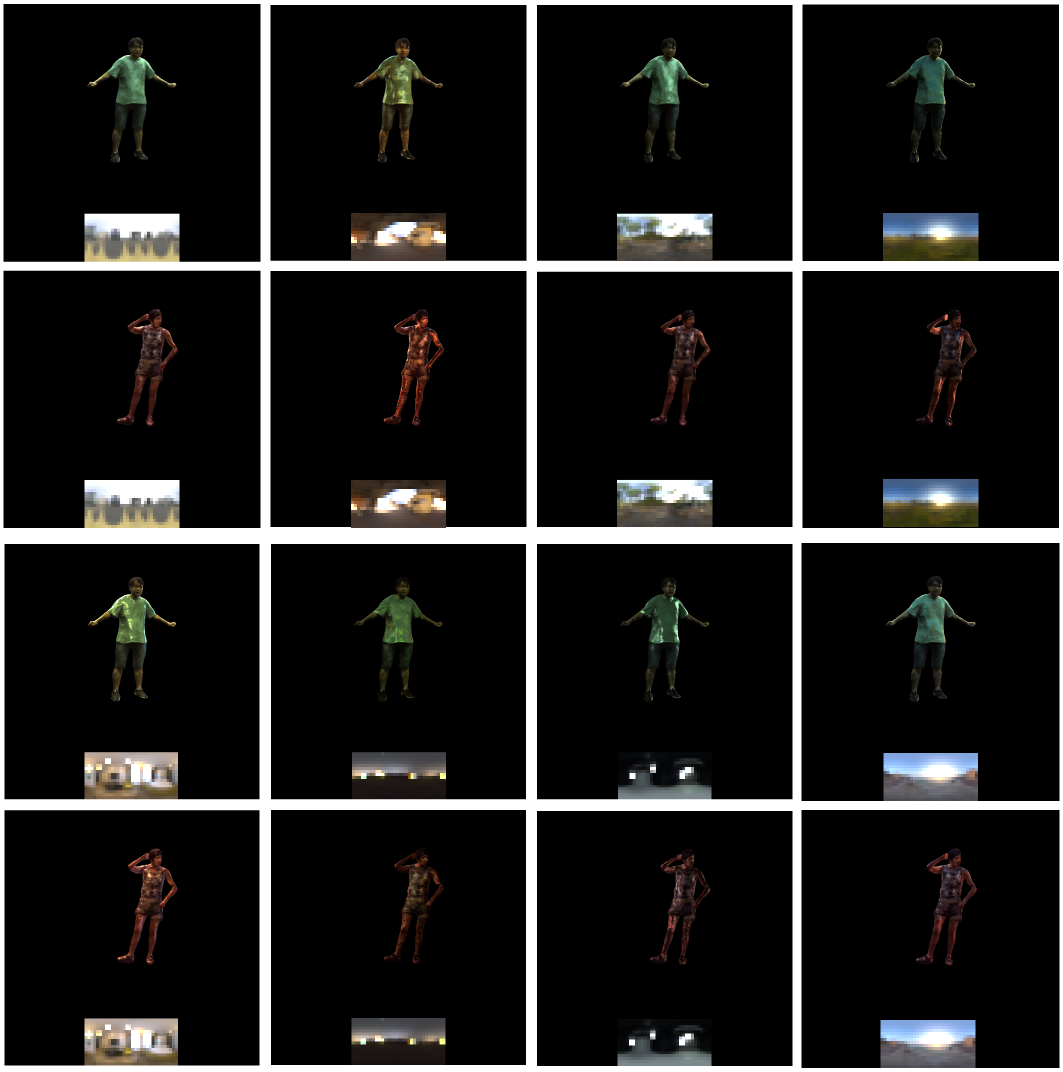


Figure 14: **Relighting visualization.** Zoom in for a better view. We strongly encourage our readers to view the supplemental video for a more comprehensive visual perception.

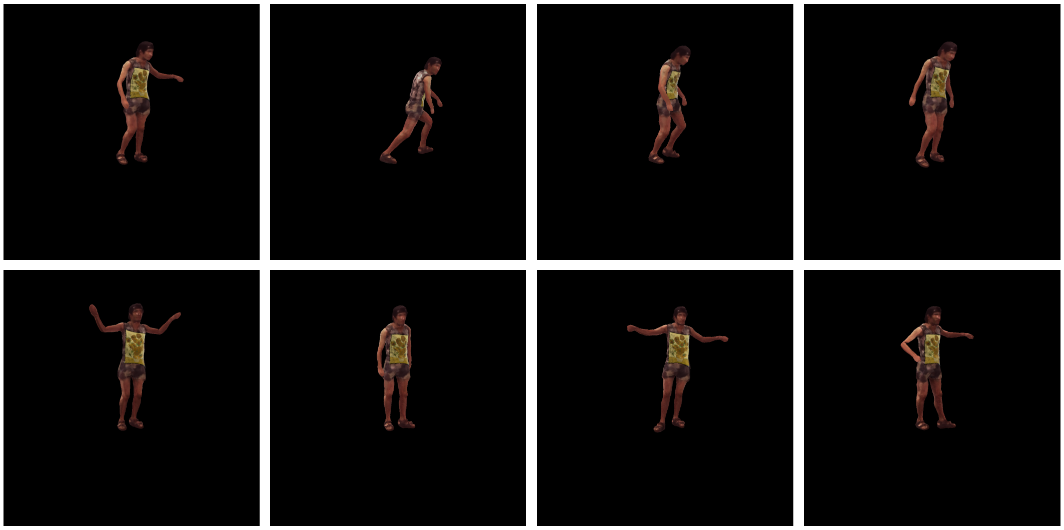


Figure 15: **Texture editing visualization.** Zoom in for a better view. We strongly encourage our readers to view the supplemental video for a more comprehensive visual perception.

William Gao, April Wang, Gal Metzer, Raymond A. Yeh, and Rana Hanocka. Tetgan: A convolutional neural network for tetrahedral mesh generation. In *BMVC*, pp. 365, 2022.

Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-Level Human Parsing via Part Grouping Network. In *ECCV*, pp. 805–822, 2018.

Chen Guo, Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Vid2avatar: 3d avatar reconstruction from videos in the wild via self-supervised scene decomposition. *arXiv:2302.11566*, 2023.

Jon Hasselgren, Jacob Munkberg, Jaakko Lehtinen, Miika Aittala, and Samuli Laine. Appearance-Driven Automatic 3D Model Simplification. *arXiv:2104.03989*, 2021.

Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light & material decomposition from images using monte carlo rendering and denoising. *arXiv:2206.03380*, 2022.

Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *ECCV*, pp. 402–418, 2022.

James T. Kajiya. The rendering equation. In *SIGGRAPH*, pp. 143–150, 1986.

Nikolai Kalischek, Torben Peters, Jan D. Wegner, and Konrad Schindler. Tetrahedral diffusion models for 3d shape generation. *arXiv:2211.13220*, 2022.

Brian Karis. Real shading in unreal engine 4. *SIGGRAPH 2013 Course: Physically Based Shading in Theory and Practice*, 2013.

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

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.*, 39(6):194:1–194:14, 2020.

Ruilong Li, Sha Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, pp. 13381–13392, 2021a.

Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. TAVA: template-free animatable volumetric actors. In *ECCV*, pp. 419–436, 2022.

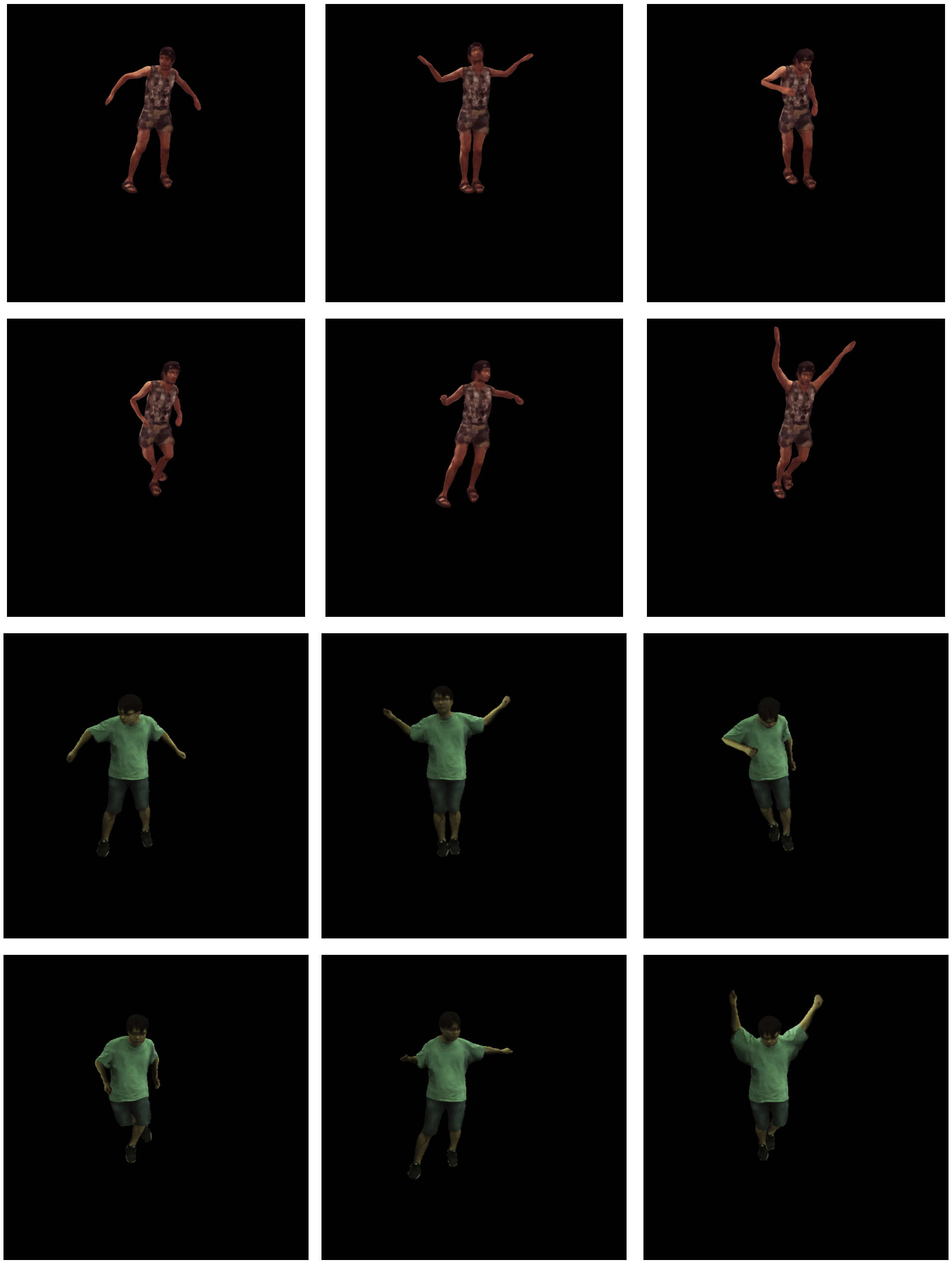
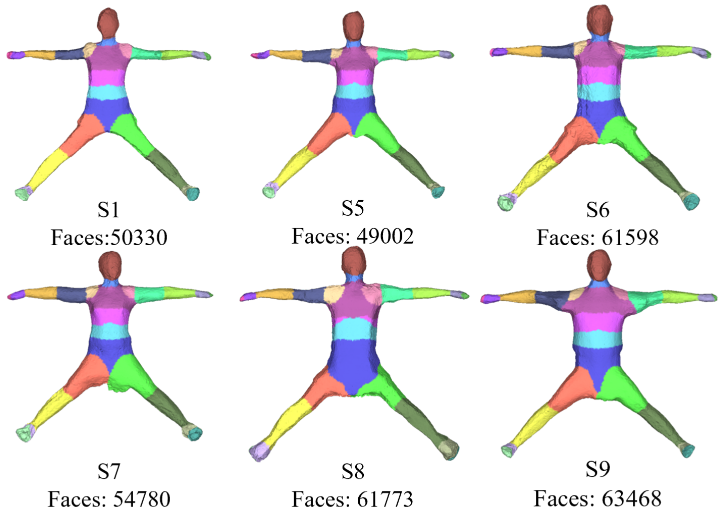


Figure 16: **Extreme pose visualization.** Zoom in for a better view. We strongly encourage our readers to view the supplemental video for a more comprehensive visual perception.



S1
Faces:50330

S5
Faces: 49002

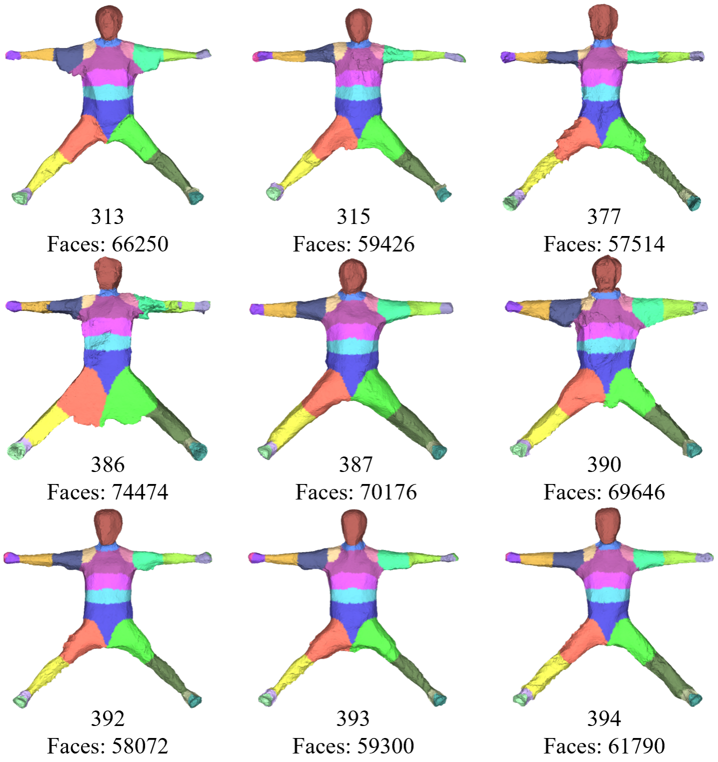
S6
Faces: 61598

S7
Faces: 54780

S8
Faces: 61773

S9
Faces: 63468

Figure 17: **Mesh visualization on the H36M dataset.** Zoom in for a better view.



313
Faces: 66250

315
Faces: 59426

377
Faces: 57514

386
Faces: 74474

387
Faces: 70176

390
Faces: 69646

392
Faces: 58072

393
Faces: 59300

394
Faces: 61790

Figure 18: **Mesh visualization on the ZJU-MoCap dataset.** Zoom in for a better view.

Xiaoting Li, Lingwei Chen, Jinquan Zhang, James R. Larus, and Dinghao Wu. Watermarking-based defense against adversarial attacks on deep neural networks. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pp. 1–8, 2021b.

Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. In *SIGGRAPH*, pp. 31:1–31:13, 2022.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015.

Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision*. Springer, 2004.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2022.

Tomas Möller, Eric Haines, and Nathaniel Hoffman. *Real-time rendering, 3rd Edition*. Peters, 2008.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022.

Jacob Munkberg, Wenzheng Chen, Jon Hasselgren, Alex Evans, Tianchang Shen, Thomas Müller, Jun Gao, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *CVPR*, pp. 8270–8280, 2022.

Thanh Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Comput. Vis. Image Underst.*, 223: 103525, 2022.

Deng Pan, Lixian Sun, Rui Wang, Xingjian Zhang, and Richard O. Sinnott. Deepfake detection through deep learning. In *BDCAT*, pp. 134–143, 2020.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pp. 5845–5854, 2021.

David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv:2104.10350*, 2021.

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, pp. 14294–14303, 2021a.

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. In *CVPR*, pp. 9054–9063, 2021b.

Scott Schaefer, Jan Hakenberg, and Joe D. Warren. Smooth subdivision of tetrahedral meshes. In *SGP*, pp. 147–154, 2004.

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*, pp. 6087–6101, 2021.

Michael Stokes, Matthew Anderson, Srinivasan Chandrasekar, and Ricardo Motta. A Standard Default Color Space for the Internet - sRGB, 1996. URL `https://www.w3.org/Graphics/Color/sRGB.html`.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the Eurographics Symposium on Rendering Techniques, Grenoble, France, 2007*, pp. 195–206, 2007.

Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. ARAH: animatable volume rendering of articulated human sdfs. In *ECCV*, pp. 1–19, 2022.

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. In *CVPR*, pp. 16189–16199, 2022.

Tianhan Xu, Yasuhiro Fujita, and Eiichi Matsumoto. Surface-aligned neural radiance fields for controllable 3d human synthesis. In *CVPR*, pp. 15862–15871, 2022.

Wang Zeng, Wanli Ouyang, Ping Luo, Wentao Liu, and Xiaogang Wang. 3d human mesh regression with dense correspondence. In *CVPR*, pp. 7052–7061, 2020.

Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *CVPR*, pp. 5453–5462, 2021a.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40:237:1–237:18, 2021b.