Relightable and Animatable Neural Avatar from Sparse-View Video

Zhen Xu [1] Sida Peng [1] Chen Geng [1] Linzhan Mou [1]
Zihan Yan [2] Jiaming Sun [3] Hujun Bao [1] Xiaowei Zhou [1*]
[1]State Key Lab of CAD&CG, Zhejiang University [2]MIT Media Lab [3]Image Derivative Inc.

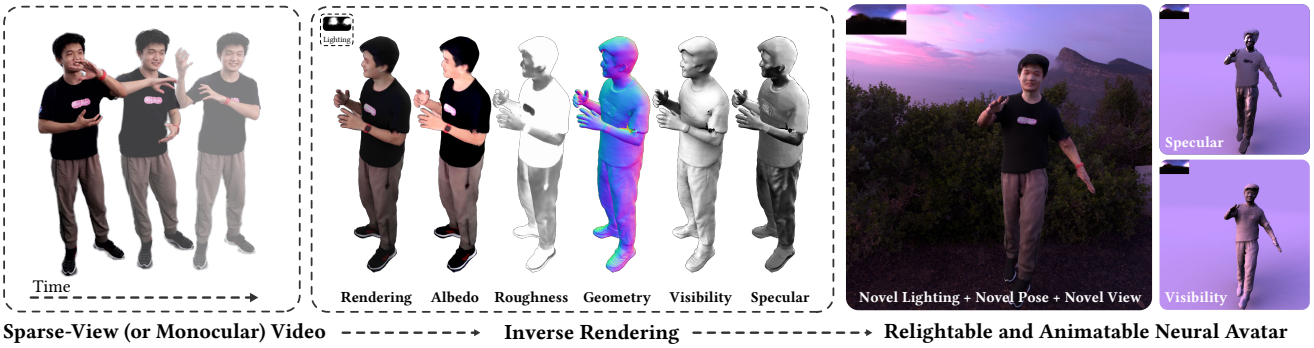

Figure 1. **Reconstructing relightable and animatable neural avatar from sparse-view (or monocular) video.** Our method takes only a sparse-view (or monocular) video as input and reconstructs a relightable and animatable neural avatar under unknown illumination, which can then be relit with arbitrary environment lights and animated with arbitrary motion sequences. **Note that our method successfully captures the shininess of the skin and pants as well as the specular highlights on the t-shirt's plastisol printings.**

Abstract

This paper tackles the challenge of creating relightable and animatable neural avatars from sparse-view (or even monocular) videos of dynamic humans under unknown illumination. Compared to studio environments, this setting is more practical and accessible but poses an extremely challenging ill-posed problem. Previous neural human reconstruction methods are able to reconstruct animatable avatars from sparse views using deformed Signed Distance Fields (SDF) but cannot recover material parameters for relighting. While differentiable inverse rendering-based methods have succeeded in material recovery of static objects, it is not straightforward to extend them to dynamic humans as it is computationally intensive to compute pixel-surface intersection and light visibility on deformed SDFs for inverse rendering. To solve this challenge, we propose a Hierarchical Distance Query (HDQ) algorithm to approximate the world space distances under arbitrary human poses. Specifically, we estimate coarse distances based on a parametric human model and compute fine distances by exploiting the local deformation invariance of SDF. Based on the HDQ algorithm, we leverage sphere tracing to efficiently estimate the surface intersection and light visibility. This allows us to develop the first system to recover animatable and relightable neural avatars from sparse view (or monocular) inputs. Experiments demonstrate that our approach is able to produce superior results compared to state-of-the-art methods. Our code will be released for reproducibility.

1. Introduction

Realistic human avatars have a range of applications [13, 61] in various domains, e.g., virtual reality, filmmaking, and video games. This work focuses on the specific setting of creating animatable and relightable human avatars from sparse-view or monocular RGB videos. This problem is challenging due to the inherent ambiguity of acquiring human geometry, materials, and motions from sparse view images [21,50]. Traditional methods [17,20,21,25,35,61,67] resolve this ambiguity via customized and costly capture devices, e.g., light stages with controllable illumination and dense camera arrays. However, such devices are restricted to professional users, impeding their universality and generalization.

Recent neural scene representation-based methods [42, 49, 65] have demonstrated the ability to extract detailed geometry and photorealistic appearance of human performers from sparse-view videos without sophisticated studio setup.

*Corresponding author.

These methods typically define the human model in canonical space and warp it into world space through a deformation module to represent human performers observed in videos. For example, AniSDF [49] models the human geometry and appearance as neural signed distance and radiance fields, and deforms them using linear blend skinning (LBS) [39] and learned local deformation networks. Albeit showing the capability of novel pose synthesis, the reconstructed avatars in these works [42,48,65] are not relightable as they bake the shading and shadow into the appearance model. As a result, the shading of the avatars under novel poses is unrealistic and the environment illumination cannot be changed, which restricts the applicability of the avatars.

Another line of works attempts to create relightable models under natural illumination through inverse rendering techniques [9, 59, 72, 75, 76], which estimate surface material parameters from input images through differentiable physically-based rendering. Computing the visibility of 3D points to the environment light is essential for accurate estimation [75, 76], but the cost of visibility computation is high. To improve efficiency, L-Tracing [15] adopts a signed distance field to represent the scene geometry and estimates the light visibility through sphere tracing, which iteratively marches along a ray using distance values until hitting the surface. Although this strategy works well on static objects, it is not suitable for animatable neural avatars [49, 65, 68], which warp the canonical SDF to world space based on a non-rigid motion field, producing a deformed SDF. The reason is that sphere tracing might not converge on the deformed SDF [56] since the SDF is inherently defined in the canonical space, thereby yielding incorrect world-space distance.

In this work, we propose a novel approach for creating relightable and animatable human avatars from sparse-view (or monocular) videos via neural inverse rendering. Inspired by previous methods [49, 65], we parameterize the human avatar as MLP networks, which predict material parameters and signed distance for any 3D point in canonical space. These values are transformed into world space for rendering through a neural deformation field. Our innovation lies in designing a hierarchical query scheme that enables a consistent approximation of 3D points' distance to the surface of the neural avatar under arbitrary human poses. This allows us to perform sphere tracing for 3D points' pixel-surface intersection and light visibility for physically-based rendering. Specifically, we smoothly blend the world-space KNN (when query points are far from the surface) distances and canonical-space neural SDF (when query points are close to the surface), approximating an SDF defined on the world-space geometry of the neural avatar. In this way, vanilla sphere tracing [29] can be performed on the deformed SDF in world space when animating and relighting the avatar, avoiding the non-linearity of canonical sphere tracing, as well as the pitfalls of world space tracing with incorrect world-space distance.

Based on the Hierarchical Distance Query algorithm, we further develop a soft visibility computation scheme by incorporating traditional distance field soft shadow (DFSS) [47] onto the deformed SDF, which is essential to the photo-realism of the relightable neural avatar. The soft shadow produced by an area light source typically requires multiple light samples to compute, while DFSS utilizes distance values to approximate the soft shadow coefficient with only a single sample. Note that it is not trivial to combine DFSS with previous methods [48, 65, 68], as they cannot produce world-space distance values from 3D points to the scene surface along an arbitrary direction.

To validate our approach, we collect a real-world multi-view dataset dubbed *MobileStage*, which captures the complex shading and shadow effects of dynamic humans with an array of mobile phone cameras. Furthermore, we extend the *SyntheticHuman* dataset [49] with novel illuminations, enabling the evaluation of relightable neural avatars with ground-truth photometric properties and relighting results. Experiments on relighting ability and novel pose synthesis show that our method outperforms the state-of-the-art with superior visual quality and physical accuracy on both real-world and synthetic datasets. Our code will be made publicly available for reproducibility.

Our contributions can be summarized as follows: (a) We propose a novel system for reconstructing relightable and animatable neural avatars from sparse-view (or monocular) videos. (b) We design a hierarchical distance query algorithm for efficient pixel-surface intersection and light visibility computation using sphere tracing. (c) We extend DFSS to drivable neural SDF, efficiently producing realistic soft shadows for the neural avatars. (d) We demonstrate quantitative and qualitative improvements compared to prior work.

2. Related work

Human avatars. To produce animatable human avatars, previous methods [13, 26, 27, 61, 63, 69] generally adopt a three-stage pipeline: they first reconstruct the human shape and appearance, then bind the shape to a skeleton, and finally animate the human model through linear blend skinning (LBS) algorithm [39]. Traditional methods tend to leverage complicated hardware, such as dense camera arrays [17,23,35,60,61] or depth sensors [4,8,57,62], to create high-fidelity human models. Recently, some optimization-based methods [5,33,50,66,68] have attempted to reconstruct human models given sparse multi-view videos. For example, Neural Body [50] represents a dynamic human model by combining SMPL model [43] with neural radiance field (NeRF) [45]. Its model parameters are learned from images through differentiable volume rendering.

To animate the reconstructed human model, some [5, 30] retrieve the skinning weights of the SMPL model for per-

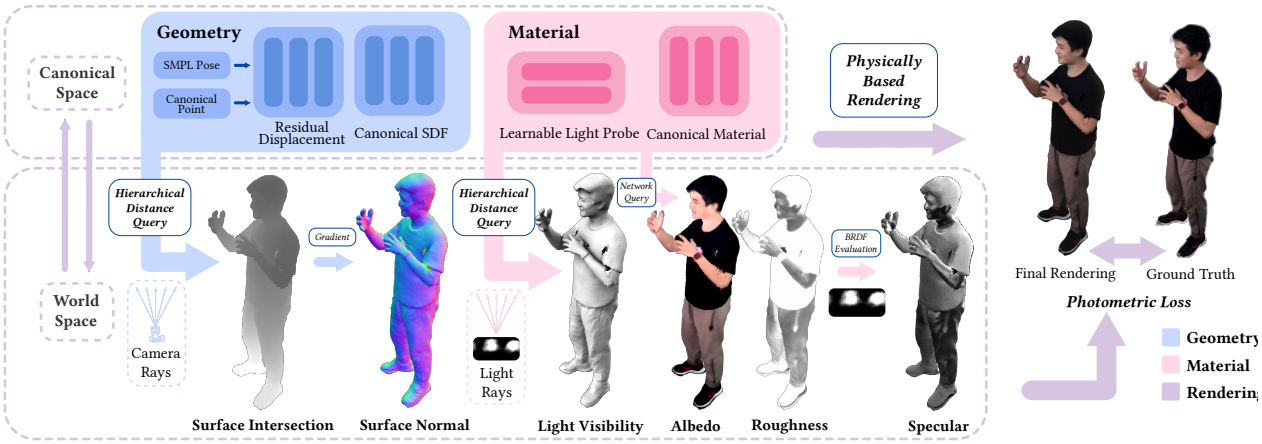

Figure 2. **Overview of the proposed approach.** Given world space camera rays, we perform sphere tracing on the hierarchically queried distances (Section 3.2) to find surface intersections and canonical correspondences (Section 3.3). Light rays generated by an optimizable light probe are also sphere traced with HDQ to compute the closest distances along the ray for soft visibility (Section 3.3). Material properties (Section 3.4) and surface normals are queried on the canonical correspondences and warped to world space. Then, the final pixel colors are computed using the rendering equation (Section 3.5).

forming the LBS algorithm. Several methods [14, 30, 48, 53] opt to optimize personalized skinning weights for the target human subject, where they represent the skinning weights as an MLP network and learn it from input data, such as human shapes [14, 53] or multi-view videos [30, 48]. Another line of works [42, 49] introduce a neural displacement field to improve animation realism. The articulated deformation is represented by the LBS model of SMPL, and the non-rigid deformation is predicted using an MLP network. While neural animatable methods can produce dynamic avatars that appear realistic, they do not model the material properties of the avatars, making them unable to adapt to different lighting conditions.

Relighting. To relight objects, a typical approach is first acquiring their material properties and then rendering with new illumination through physically-based rendering. Traditional methods [21, 54] mostly require a known illumination for calculating the material parameters through photometric stereos. Light stage-based approaches [20, 21, 25, 67] build a controllable light array to capture images of target subjects under multiple illuminations. Based on these captured images and the known illuminations, they solve for the unknown material properties. More recently, neural inverse rendering methods [9–11, 15, 38, 59, 72, 75, 76] explore more flexible capture settings, where the illumination is unknown or even variable.

Motivated by its potential for many human-centric applications, research on human relighting has been widely conducted in the literature [44, 46, 71, 74]. Same as other objects, the material properties of human subjects can be recovered using neural inverse rendering methods. The difference is that human subjects exhibit more strong material priors. Therefore, some methods [6, 31, 32, 36, 46, 71] attempt to train neural networks to predict human materials from a single image. Recently, Relighting4D [16] have attempted to acquire human materials from sparse multi-view videos. However, Relighting4D is not designed to relight animatable avatars realistically, limiting its applicability.

3. Method

Given a sparse-view (or monocular) video of a human performer under natural and unknown illumination, we learn to reconstruct the drivable geometry and photometric properties of the human performer to create an animatable and relightable neural avatar. We assume the human poses and the foreground masks are provided as in [42, 48–50].

3.1. Relightable and Animatable Neural Avatar

We formulate the relightable and animatable avatar using a set of canonical space neural fields and a warping between world and canonical space defined by the linear blend skinning algorithm [39] and a displacement field [42, 48, 49, 66]. In the canonical space, we define a set of geometry ($S(\boldsymbol{x})$) and material neural fields ($A(\boldsymbol{x})$ and $\Gamma(\boldsymbol{x})$) for the animated human model. The canonical space displacement field $F_{\Delta\boldsymbol{x}}$ provides additional pose-dependent deformation on top of SMPL inverse LBS. More details about the warping process are provided in Section 3.2 and the supplementary.

The relightable and animatable neural avatar will be rendered by casting camera rays in world space and finding the surface intersection points $\boldsymbol{x}_s$ and their normals $\boldsymbol{n}_s$ using the Hierarchical Distance Query (HDQ) algorithm, whose material properties albedo α_s and roughness γ_s can be ob-

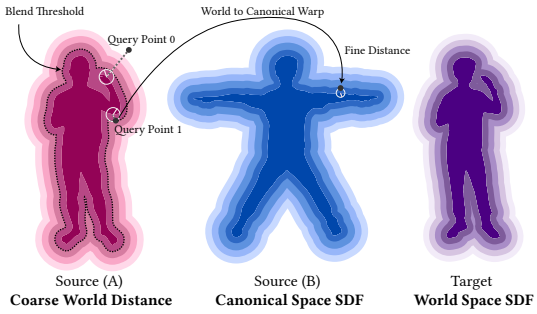

Figure 3. **Illustration of the Hierarchical Distance Query algorithm.** In this figure, query point 0 falls out of the cut-off threshold thus its coarse distance is used directly as the world space distance, while query point 1 blends the coarse world space distance and fine canonical distance together since it is within the range of local SDF values.

tained from the canonical mateiral MLPs $A(\boldsymbol{x})$ and $\Gamma(\boldsymbol{x})$, composing the BRDF $R_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{n}_s)$. Light visibility $V_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i)$ can be computed by performing HDQ sphere tracing on all incoming light directions. We also incorporate Distance Field Soft Shadow (DFSS) algorithm [3, 7, 47] onto our drivable neural distance fields for soft-visibility computation. These properties will be integrated around the hemisphere $\boldsymbol{\omega}_i \in \Omega$ using the rendering equation [34]:

$$L_o = \int_\Omega L_s(\boldsymbol{\omega}_i) R_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{n}_s) V_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i)(\boldsymbol{n}_s \cdot \boldsymbol{\omega}_i) d\boldsymbol{\omega}_i, \tag{1}$$

where $L_o(\boldsymbol{x_s}, \boldsymbol{\omega}_o) \in \mathbb{R}^3$ is the outgoing radiance at the surface intersection point $\boldsymbol{x}_s$, $\boldsymbol{\omega}_o$ is the outgoing radiance direction and $\boldsymbol{\omega}_i$ is the incoming radiance direction. In this paper, we use the Microfacet BRDF model in [64] which is defined in the canonical space of the animatable avatar, and an optimizable light probe image $L_s(\boldsymbol{\omega}_i) \in \mathbb{R}^{16\times32\times3}$. An overview of the relightable and animatable avatar can be found in Figure 2.

3.2. Hierarchical Distance Query

Given the world space query point $\boldsymbol{x}$, we approximate its world space distance $d^{world}(\boldsymbol{x})$ to the closest surface point on the neural avatar with the Hierarchical Distance Query algorithm $d^{world}(\boldsymbol{x}) \approx \tilde{d}^{world}(\boldsymbol{x}) = \text{HDQ}(\boldsymbol{x})$, which is later used for Sphere Tracing [29]. The query algorithm consists of four stages: (a) coarse distance query, (b) inverse warping, (c) fine distance query, and (d) smooth distance blending.

Coarse distance query. We first perform a geodesically-aware signed K Nearest Neighbor (GS-KNN) algorithm [52] on the posed vertices $\boldsymbol{v} \in \mathcal{V}$ of the driven parametric human model (SMPL-H [51]). GS-KNN produces the indices $\mathcal{I}_K = \{i_0, ..., i_K\}$ of the K closest points to $\boldsymbol{x}$ in $\mathcal{V}$, and its corresponding world-space closest vertices $\mathcal{V}_K = \{\boldsymbol{v}_0 ..., \boldsymbol{v}_K\}$, distances $\mathcal{D}_K = \{d_0 ..., d_K\}$, normals $\mathcal{N}_K = \{\boldsymbol{n}_0 ..., \boldsymbol{n}_K\}$ and blend weights $\mathcal{W}_K = \{\boldsymbol{w}_0 ..., \boldsymbol{w}_K\}$. We set $K = 10$ through all experiments. The unsigned distance $\mathcal{D}$ is augmented with the sign of the dot product between $\boldsymbol{x} - \boldsymbol{v}$ and $\boldsymbol{n}$ to produce a coarse SDF. We additionally discard the k-th neighbor $\boldsymbol{v}_k$ if its canonical correspondence (T-Pose of SMPL-H) is too far from the canonical correspondence of the nearest neighbor, K is set to 10 for all experiments. The coarse level world space SDF is defined as $d^{world}_{coarse} = \frac{\sum_{k=0}^{K} d_k}{K}$.

Inverse warping. We follow the previous literature [42, 48] and use the linear blend skinning algorithm [39] to perform the inverse warping. The details can be found in the supplementary material.

Fine distance query. Given the warped query point $\boldsymbol{x}'$, the pose-dependent displacement field $F_{\Delta \boldsymbol{x}}$ adds small perturbation to produce the final canonical space query point $\boldsymbol{x}''$. We implement $F_{\Delta \boldsymbol{x}}$ as an MLP with the human pose at the fth frame Θ_f and $\boldsymbol{x}'$ as input. The displaced canonical point $\boldsymbol{x}''$ fed into the canonical distance model S is defined as

$$\boldsymbol{x}'' = \boldsymbol{x}' + F_{\Delta \boldsymbol{x}}(\Theta_f, \boldsymbol{x}'). \tag{2}$$

Then, the fine canonical distance value can be obtained by querying the network $d^{can}_{fine} = S(\boldsymbol{x}'')$.

Smooth distance blending. Since SDF values of points close to the surface are hardly affected by LBS (Figure 2 and 5 of the supplementary), we propose to blend the fine canonical space distance value d^{can}_{fine} and the coarse world space distance d^{world}_{coarse} using a smooth function to produce the final approximated world space distance value $\tilde{d}^{world}$

$$\tilde{d}^{world} = \begin{cases} d^{world}_{coarse} & \text{, if } d^{world}_{coarse} > \tilde{T}_d \\ d^{can}_{fine}(1 - \frac{d^{can}_{fine}}{T_d}) + d^{world}_{coarse} \frac{d^{can}_{fine}}{T_d} & \text{, otherwise} \end{cases} \tag{3}$$

where $\tilde{T}_d$ is the distance threshold for cutting off coarse and fine distances, which is empirically set to 0.1. Note that we only perform the evaluation of S on points that satisfy the cutoff criteria $d^{world}_{coarse} \leq \tilde{T}_d$ for efficiency.

3.3. Geometry

Our physically based renderer requires the pixel-surface intersection $\boldsymbol{x}_s \in \mathbb{R}^3$, surface normal $\boldsymbol{n}_s \in \mathbb{R}^3$, and light

visibility $V(\boldsymbol{x}_s, \boldsymbol{\omega}_i) \in \mathbb{R}$ as input. Using the Hierarchical Distance Query, these values can be easily obtained from the world space SDF of the neural avatar under arbitrary human poses.

Surface intersection. Given a camera ray and the neural avatar's SDF, we compute the location $\boldsymbol{x}_s$ at which the ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ from the camera origin $\boldsymbol{o}$ along the ray direction $\boldsymbol{d}$ intersects the surface of the posed neural avatar. Specifically, we perform N_{st} Sphere Tracing iterations with the world space distance $\tilde{d}^{world} = \text{HDQ}(\boldsymbol{x})$ using Hierarchical Distance Query until the ray converges to the surface intersection point $\boldsymbol{x}_s$. The detailed algorithm is listed in the supplementary. N_{st} is set to 16 across all experiments.

Surface normal. The analytic normal direction $\boldsymbol{n}$ of any 3D points could be computed as the gradient of the neural SDF using $\nabla\tilde{d}^{world}(\boldsymbol{x})$. Although the hierarchical distance is differentiable, computing gradient through the whole query process is not efficient. Instead, we notice that surface intersections should satisfy the cutoff criteria of smooth distance blending in Section 3.2, that is

$$\tilde{d}^{world}(\boldsymbol{x}_s) = d^{can}_{fine}(\boldsymbol{x}_s), d^{world}_{coarse}(\boldsymbol{x}_s) \leq \tilde{T}_d. \tag{4}$$

Thus, the world space normal can be computed using $\nabla S(\boldsymbol{x}^{can}_s)$ and transformed from canonical to world space using the forward warping process. More details can be found in the supplementary.

Light visibility. Light visibility $V(\boldsymbol{x}, \boldsymbol{\omega}_i)$ from any 3D point $\boldsymbol{x}$ along any light direction $\boldsymbol{\omega}_i$ can be computed as whether the light path $\boldsymbol{x} + t\boldsymbol{\omega}_i$ is occluded by the geometry of the posed neural avatar, which is later integrated in the rendering equation [34] around the hemisphere. Since we use a discrete light probe $L_s(\boldsymbol{\omega}_i) \in \mathbb{R}^{16\times32\times3}$, the visibility term for every light direction needs to be integrated on the area of the pixel of $L_s(\boldsymbol{\omega}_i)$, which is time-consuming. Thanks to the global meaning of distance field, this occlusion and integration process can be approximated using Distance Field Soft Shadow (DFSS) [3, 7, 47], producing soft visibility with a single light sample. Specifically, we compute the visibility as the soft penumbra coefficient $p_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i)$:

$$p_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i) = \min(\frac{\tilde{d}^{world}(\boldsymbol{x}_s + t_0\boldsymbol{\omega}_i)}{2t_0\sqrt{\frac{a}{\pi}}}, ..., \frac{\tilde{d}^{world}(\boldsymbol{x}_s + t_{N^{vis}_{st}}\boldsymbol{\omega}_i)}{2t_{N^{vis}_{st}}\sqrt{\frac{a}{\pi}}}), \tag{5}$$

for each surface point $\boldsymbol{x}_s$ along one of the 512 light directions $\boldsymbol{\omega}_i$ defined by $L_s(\boldsymbol{\omega}_i)$ during the N^{vis}_{st} sphere tracing steps, which is set to 4 for all experiments. The ratio between the two tangent values $\frac{\tilde{d}^{world}}{t}$ and $2\sqrt{\frac{a}{\pi}}$ serves as an approximation of the ratio of light being occluded by the geometry from $\boldsymbol{x}_s$ along $\boldsymbol{\omega}_i$. Thanks to the smooth blending of d^{world}_{coarse} and d^{can}_{fine} in Section 3.2, our soft visibility scheme produces realistic and smooth soft shadow even when distance values from the parametric human model [51] and the canonical neural SDF are not perfectly aligned. A detailed listing of this algorithm is provided in the supplementary.

3.4. Reflectance

We adopt the Microfacet BRDF model in [64] for our material representation, which is composed of a diffuse albedo $\alpha \in \mathbb{R}^3$ term and a specular roughness $\gamma \in \mathbb{R}$ term. We use a fixed Fresnel term of 0.04. Similar to [16, 75, 76], we paramterize the albedo and roughness map with two MLPs $\alpha = A(\boldsymbol{x}'')$ and $\gamma = \Gamma(\boldsymbol{x}'')$, which is defined in the same canonical frame as $S(\boldsymbol{x}'')$ and $F_\Delta(\boldsymbol{x}')$ in Section 3.2. The BRDF model is denoted $R_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{n}_s)$ where $\boldsymbol{\omega}_i$ is the incoming radiance direction, $\boldsymbol{\omega}_o$ is the outgoing radiance direction and $\boldsymbol{n}_s$ is the surface normal.

Given world space query point $\boldsymbol{x}_s$ and its corresponding canonical space point $\boldsymbol{x}''$, we obtain the albedo α and roughness γ by querying their canonical neural fields A and Γ, which can then be converted to BRDF values as defined in [64]. Our physically-based renderer also takes a light probe $L_s(\boldsymbol{\omega}_i) \in \mathbb{R}^{16\times32\times3}$ as input, which is represented by an optimizable neural texture during training and replaced with the designated environment map during relighting [16, 19, 75].

3.5. Training

We use 512 discrete incoming light directions defined by the light probe $L_s(\boldsymbol{\omega}_i) \in \mathbb{R}^{16\times32\times3}$ to approximate the Rendering Equation [34] as

$$L_o = \sum_{\boldsymbol{\omega}_i} L_s(\boldsymbol{\omega}_i) R_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{n}_s) V_s(\boldsymbol{x}_s, \boldsymbol{\omega}_i)(\boldsymbol{n}_s \cdot \boldsymbol{\omega}_i)\Delta\boldsymbol{\omega}_i, \tag{6}$$

where $\Delta\boldsymbol{\omega}_i$ is the solid angle of the incoming light $\boldsymbol{\omega}_i$ sampled from the light probe $L_s(\boldsymbol{\omega}_i)$ and $L_o(\boldsymbol{x_s}, \boldsymbol{\omega}_o) \in \mathbb{R}^3$ is the outgoing radiance at the surface intersection $\boldsymbol{x}_s$.

We optimize our relightable and animatable neural human avatar by rendering the image with given camera poses and comparing the pixel values L_o against the ground truth ones L_{gt}. The main loss function is defined as $\mathcal{L}_{data} = \sum_{\boldsymbol{r}\in\mathcal{R}} \|L_o(\boldsymbol{r}) - L_{gt}(\boldsymbol{r})\|_2$, where $\boldsymbol{r} = \boldsymbol{o} + t\boldsymbol{d} \in \mathcal{R}$ denotes all camera rays in the forward rendering process. Due to the ill-posed nature of the problem, we adopt a two-stage training strategy and add additional regularizations one the geometry (eikonal loss $\mathcal{L}_{eik}$) and material (sparsity loss $\mathcal{L}_{ent}$ and smoothness loss $\mathcal{L}_a, \mathcal{L}_r$). We elaborate on the details of each loss term and the training stragety in the supplementary. The training takes 20 hours on an Nvidia RTX 3090. Rendering a 512×512 image takes 5s.

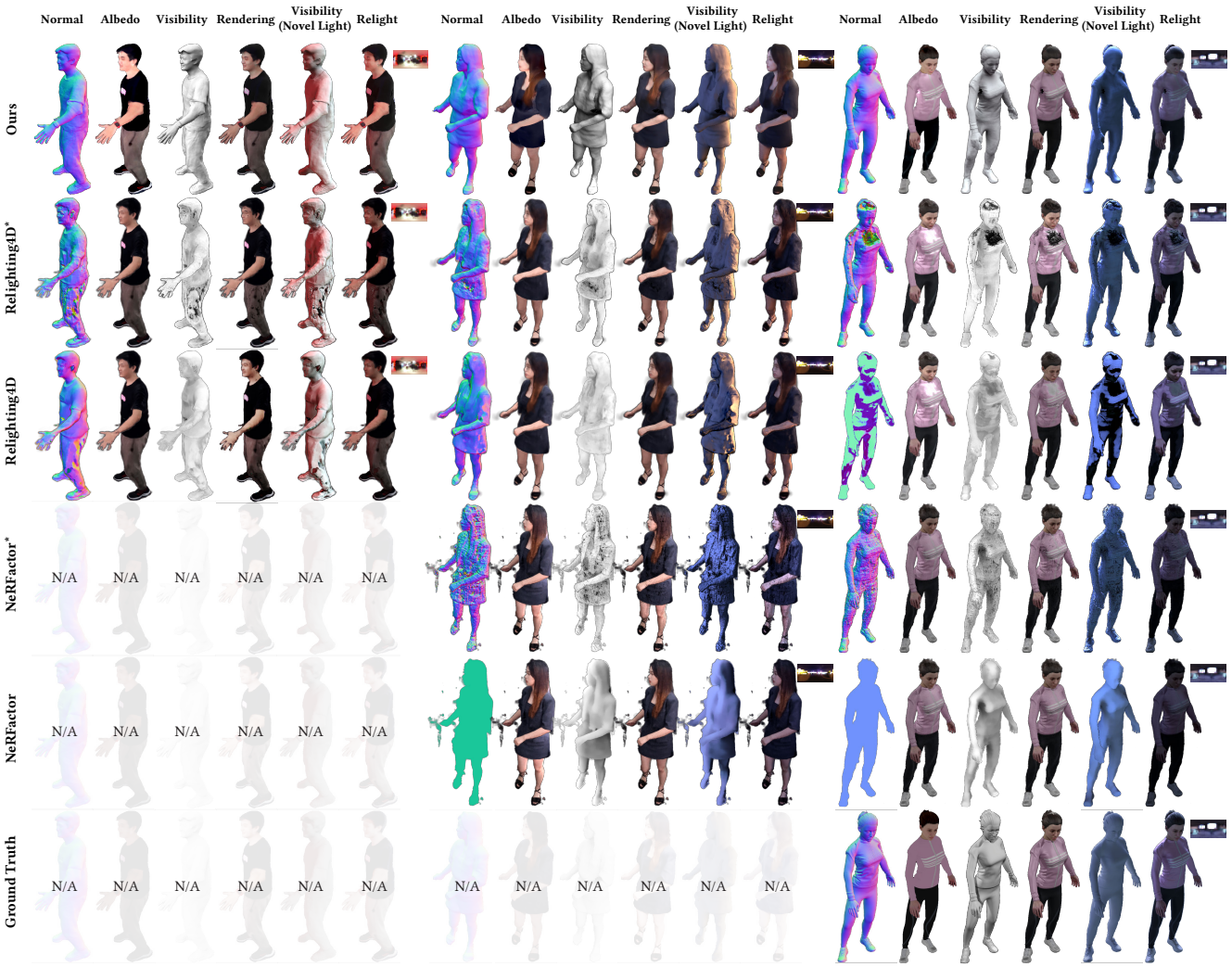

Figure 4. **Qualitative comparison of our method and baselines.** The first six columns display the results of synthesizing a character in a novel pose from the *MobileStage* dataset. The middle six columns depict a character in a training pose from the *MobileStage* dataset. For the last six columns, we show results from *SyntheticHuman++*, for which we have ground truth as reference. Note that NeRFactor is only trained on 1 frame. Relighting4D* and NeRFactor* denote directly computing normal and visibility using their density MLPs.

4. Experiments

In this section, we conduct qualitative and quantitative experiments to investigate the performance of our relightable neural avatars. All hyperparameters are fixed through out the experiments unless otherwise specified. In Section 4.1, we briefly introduce the datasets used for evaluation. Then we make quantitative and qualitative comparisons with three baseline methods in Section 4.2. Finally, we conduct ablation studies to investigate the effectiveness of our Hierarchical Distance Query and the soft visibility scheme in Section 4.3.

4.1. Datasets

We collect two datasets *MobileStage* and *SyntheticHuman++* for evaluation. *MobileStage* is a real-world multi-view (36 views) dataset created with synchronized mobile phone cameras on 4 real-world humans. We uniformly select 12 views for training. *SyntheticHuman++* contains 4 sequences (20 views) of dynamic 3D human models with ground truth shape and relighting information. We uniformly select 10 views for training for the sparse-view setting and we use the fourth view for the monocular setting. Please refer to the supplementary material for more detail.

4.2. Baseline Comparisons

Baselines. To the best of our knowledge, there are very few prior works that study the exact same setting as ours, i.e. training with unknown illumination and sparse-view (or monocular) videos while rendering with novel illumination

Table 1. **Quantitative comparison.** We compare our method with baselines on the *SyntheticHuman++* dataset. All image metrics are computed in the foreground region in linear color space. "$*$" denotes variants without the normal and visibility MLPs. AniSDF produces baked lighting in the rendering network, while NeRFactor [75] and Relighting4D [16] perform poorly on the uniform shading task for visibility since the reconstructed geometry is too rough. Our method works well even on the challenging monocular setting.

		Normal	**Diffuse Albedo**			**Relighting**			**Visibility**		
		Degree ↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Sparse-View	Ours	**12.44**	**29.01**	**0.933**	**0.119**	**22.69**	**0.861**	**0.206**	**20.20**	**0.848**	**0.155**
	Relighting4D*	29.38	24.70	0.885	0.183	22.13	0.835	0.237	15.22	0.763	0.252
	Relighting4D	93.83	24.71	0.885	0.183	20.87	0.774	0.276	5.366	0.514	0.375
	NeRFactor* (1 frame)	34.29	22.23	0.817	0.226	21.04	0.758	0.313	11.37	0.581	0.387
	NeRFactor (1 frame)	51.92	22.23	0.817	0.226	20.70	0.757	0.299	10.56	0.597	0.361
	AniSDF	14.72	22.13	0.862	0.202	17.55	0.799	0.262	-	-	-
Monocular	Ours	**18.71**	23.42	**0.873**	**0.176**	**22.45**	**0.831**	**0.224**	**17.95**	**0.761**	**0.212**
	Relighting4D*	26.17	**25.37**	0.864	0.210	21.81	0.802	0.254	17.10	0.709	0.286
	Relighting4D	81.74	25.36	0.864	0.210	21.85	0.806	0.268	16.18	0.726	0.302
	AniSDF	20.36	21.51	0.812	0.255	18.29	0.745	0.297	-	-	-

and novel human poses. We take NeRFactor [75] and Relighting4D [16] as baselines and make comparisons with them on both real and synthetic datasets. Since NeRFactor is designed to handle static objects, we only train and evaluate it on the multi-view images of the first frame of each video. We observe that their normal and visibility MLPs often fail under complex human motions, thus we additionally compare with a Relighting4D* and NeRFactor* variant where we use the normal and visibility computed from the density MLP instead of the normal and visibility MLPs. To illustrate the effectiveness of our proposed components, we additionally render AniSDF [49] as a baseline.

Metrics. For quantitative analysis, we compare the normal (in degrees), albedo (in PSNR, SSIM and LPIPS [73]), light visibility rendered with uniform BRDF (in PSNR, SSIM, and LPIPS) and relighting (in PSNR, SSIM, and LPIPS) results on 6 different light probes obtained from Polyhaven [2]. Following [72], we align the diffuse albedo and rendered images with ground truth ones before computing metrics to mitigate the inherent scale ambiguity in the inverse rendering problem. We do not compare the roughness term since Blender uses a different Principled BRDF model from [64]. Environment map of *SyntheticHuman* [49] is not available since they used programmatically defined light sources. Qualitative results can be found in the supplementary video. We compare the uniform shading results to evaluate the visibility quality, where the BRDFs of the reconstructed avatars are set to 0.8 across all radiance directions (denoted "Visibility") when rendering. Since *SyntheticHuman* [49] does not provide ground truth models for novel poses, we only perform quantitative comparisons on training poses in Table 1, while qualitative analysis of animating the avatars can be found in Figure 4 and the supplementary video.

Results. As shown in Figure 4, our approach can successfully decompose the material and dynamic geometry of the neural avatar, generating a relightable neural avatar from only sparse-view (or monocular) video inputs. In comparison, NeRFactor [75] trained on 1 video frame overfits the training image when training views are sparse. Relighting4D [16] passes structured latent codes [50] to NeRFactor's MLPs, enabling it to relight a dynamic video of human performance. However, its quality decreases greatly when synthesizing novel poses. This is mainly because the visibility and normal MLP used in [16] is not generalizable to novel human motions. For the Relighting4D* variant, the density backbone still fails to generalize to novel poses [42, 49]. AniSDF [49] bakes illuminations effects like self-occlusions onto the rendering network, thus the reconstructed neural avatar looks unrealistic under novel illuminations. Qualitative results on monocular inputs can be found in Figure 13 and the supplementary video. Relighting4D and NeRFactor take 3s to render a 512×512 image, their "*" variants take 50s and our method takes 5s.

4.3. Ablation Studies

In this part, we ablate the effectiveness of our proposed Hierarchical Distance Query and soft visibility scheme on relighting quality with the *jody* model of *SyntheticHuman++* under the sparse-view setting. We provide more detailed ablation on the soft visibility scheme, the number of sphere tracing steps, the number of vertices for GS-KNN, and the number of canonical material samples in the supplementary.

Effectiveness of Hierarchical Distance Query and soft visibility scheme. In Figure 5, We compare the results of performing sphere tracing on the canonical space distance ("w/o d^{world}_{coarse}"), world space distance (("w/o d^{can}_{fine}")) and our proposed hierarchically queried distance ("Ours"). As

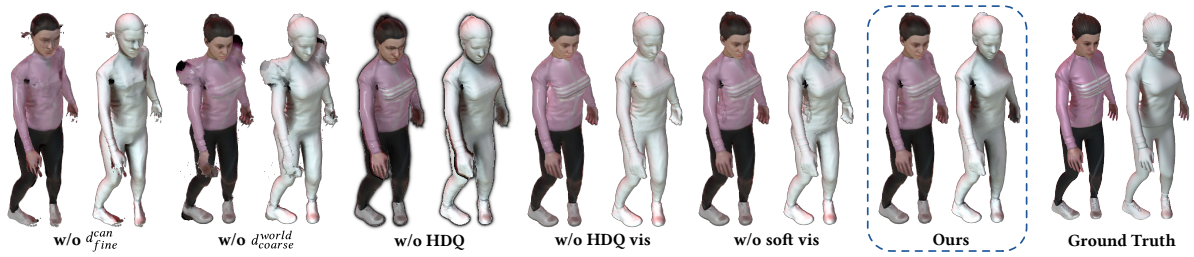

Figure 5. **Effectiveness of Hierarchical Distance Query**. Performing sphere tracing using only the canonical distance d^{can}_{fine} or coarse world distance d^{world}_{coarse} results in incorrect surface intersection $\boldsymbol{x}_s$ and soft visibility p_s, while tracing with our proposed Hierarchical Distance Query produces correct results. Using hard shadow ("w/o soft vis") or local shadow ("w/o HDQ vis") leads degraded perceptual quality.

Table 2. **Ablation study on the proposed Hierarchical Distance Query algorithm.** Qualitative results can be found in Figure 5. The "w/o HDQ" variant uses naive volume rendering (128 samples per ray) to compute pixel-surface intersection and visibility, which is not only slow (60s per image at 512 × 512 resolution compared to our 5s per image) but also exhibits inferior rendering quality.

	Relighting			Visibility		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Ours	**21.57**	**0.853**	**0.168**	20.53	0.869	**0.142**
w/o soft vis	21.19	0.848	0.173	**21.27**	**0.873**	0.145
w/o HDQ vis	21.00	0.844	0.175	20.88	0.869	0.143
w/o HDQ	21.36	0.753	0.196	20.00	0.760	0.173
w/o d^{world}_{coarse}	20.69	0.792	0.236	18.75	0.767	0.250
w/o d^{can}_{fine}	19.56	0.784	0.245	14.63	0.758	0.233

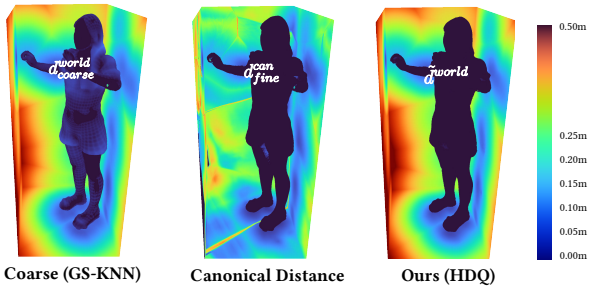

Figure 6. **Visualization of hierarchically queried distance**. Hot colors indicate large distance values. We visualize d^{world}_{coarse}, d^{can}_{fine} and $\tilde{d}^{world}$ on (a) surface intersection points, which should be zero everywhere, and (b) bounding box of the human model, which should reflect the real world space SDF. Note that the geometry is fixed to visualize the distance values. The coarse distance d^{world}_{coarse} of near surfaces points is not strictly zero (indicated by the light blue regions on the body), and d^{can}_{fine} is incorrect in world space since it is defined by the canonical space network S. This leads to both of them reporting errornous geometry for the relighting task as seen in Figure 5. **In contrast to both, our proposed HDQ algorithm is well behaved both when near and far from the geometry.**

shown in the figure, the canonical space distance is incorrect when the query point is far from the actual surface of the human geometry, resulting in incorrect surface intersection points after the termination of the sphere tracing algorithm. Additionally, computing light visibility on this incorrect distance field would lead to false black regions since distances far from surface points are not reported correctly. Performing surface intersection and visibility computation on the coarse distance results in distorted rendering results. The "w/o HDQ" variant uses volume rendering of 128 samples per ray for surface intersection and light visibility computation, leading to an excessive rendering time of 60s per image for a resolution of 512×512, while our HDQ algorithm is able to obtain 10x speed-up at 5s per image with superior rendering quality.

We demonstrate the effectiveness of our HDQ-DFSS algorithm by comparing it with two other variants where (a) hard visibility is used ("w/o soft vis") and (b) only local visibility computed from normal is used ("w/o HDQ vis"). The quantitative comparison of all three variants can be found in Table 2. Note that although the "w/o soft" variants report higher PSNR and SSIMs, the visual quality of hard cast shadows is worse than ours, as indicated by the LPIPS metric and visible in Figure 5.

5. Conclusion and Discussion

This paper presents a novel framework to reconstruct relightable and animatable neural avatars from only sparse-view (or monocular) video input. We generalize the canonical distance field to arbitrary human poses via a hierarchical distance query scheme, with which the photometric properties of the neural avatar can be easily retrieved for relighting. We demonstrate that together with other innovative components, our approach reconstructs high-quality animatable geometry and material, supporting realistic relighting. This work also has some limitations. Although the proposed method produces high-quality relighting results from challenging sparse-view or monocular settings, it has the natural limitation of neural field methods in that it requires a long training time of 20 hours. Future work could consider recent neural field acceleration methods to further increase the training speed.

References

[1] Easymocap - make human motion capture easier. Github, 2021. 13

[2] Poly Haven, 2023. 7, 12

[3] Sebastian Aaltonen. Gpu-based clay simulation and ray-tracing tech in claybook. *San Francisco, CA*, 2(5), 2018. 4, 5, 16

[4] Kairat Aitpayev and Jaafar Gaber. Creation of 3d human avatar using kinect. *Asian Transactions on Fundamentals of Electronics, Communication & Multimedia*, 1(5):12–24, 2012. 2

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8387–8397, 2018. 2

[6] Thiemo Alldieck, Mihai Zanfir, and Cristian Sminchisescu. Photorealistic monocular 3d reconstruction of humans wearing clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1506–1515, 2022. 3

[7] Róbert Bán, Csaba Bálint, and Gábor Valasek. Area lights in signed distance function scenes. In *Eurographics (Short Papers)*, pages 85–88, 2019. 4, 5, 16

[8] Federica Bogo, Michael J Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 2300–2308, 2015. 2

[9] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 2, 3

[10] Mark Boss, Andreas Engelhardt, Abhishek Kar, Yuanzhen Li, Deqing Sun, Jonathan T Barron, Hendrik Lensch, and Varun Jampani. Samurai: Shape and material from unconstrained real-world arbitrary image collections. *arXiv preprint arXiv:2205.15768*, 2022. 3

[11] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34:10691–10704, 2021. 3

[12] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *arXiv preprint arXiv:2206.15258*, 2022. 18

[13] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003. 1, 2

[14] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11594–11604, 2021. 3

[15] Ziyu Chen, Chenjing Ding, Jianfei Guo, Dongliang Wang, Yikang Li, Xuan Xiao, Wei Wu, and Li Song. L-tracing: Fast light visibility estimation on neural surfaces by sphere tracing. In *European Conference on Computer Vision*, pages 217–233. Springer, 2022. 2, 3, 16

[16] Zhaoxi Chen and Ziwei Liu. Relighting4d: Neural relightable human from videos. In *European Conference on Computer Vision*, pages 606–623. Springer, 2022. 3, 5, 7, 12, 13, 17, 20, 21

[17] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 1, 2

[18] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 13

[19] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Acm siggraph 2008 classes*, pages 1–10. 2008. 5

[20] Paul Debevec. The light stages and their applications to photoreal digital actors. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES, 2012. 1, 3

[21] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000. 1, 3

[22] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In *CVPR*, 2021. 14

[23] Oliver Grau. Studio production system for dynamic 3d content. In *Visual Communications and Image Processing 2003*, volume 5150, pages 80–89. SPIE, 2003. 2

[24] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 16

[25] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (ToG)*, 38(6):1–19, 2019. 1, 3

[26] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2):1–17, 2019. 2

[27] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063, 2020. 2

[28] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 17

[29] John C Hart et al. Sphere tracing: Simple robust antialiased rendering of distance-based implicit surfaces. In *Siggraph*, volume 93, pages 1–11, 1993. 2, 4, 18

[30] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2020. 2, 3

[31] Umar Iqbal, Akin Caliskan, Koki Nagano, Sameh Khamis, Pavlo Molchanov, and Jan Kautz. Rana: Relightable articulated neural avatars. *arXiv preprint arXiv:2212.03237*, 2022. 3

[32] Chaonan Ji, Tao Yu, Kaiwen Guo, Jingxin Liu, and Yebin Liu. Geometry-aware single-image full-body human relighting. In *European Conference on Computer Vision*, pages 388–405. Springer, 2022. 3

[33] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *European Conference on Computer Vision*, pages 402–418. Springer, 2022. 2

[34] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 4, 5

[35] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 4(1):34–47, 1997. 1, 2

[36] Yoshihiro Kanamori and Yuki Endo. Relighting humans: occlusion-aware inverse rendering for full-body human images. *arXiv preprint arXiv:1908.02714*, 2019. 3

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 17

[38] Zhengfei Kuang, Kyle Olszewski, Menglei Chai, Zeng Huang, Panos Achlioptas, and Sergey Tulyakov. Neroic: Neural rendering of objects from online image collections. *arXiv preprint arXiv:2201.02533*, 2022. 3

[39] 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*, pages 165–172, 2000. 2, 3, 4, 18

[40] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021. 12, 14

[41] Shanchuan Lin, Linjie Yang, Imran Saleemi, and Soumyadip Sengupta. Robust high-resolution video matting with temporal guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 238–247, 2022. 13

[42] 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 Transactions on Graphics (TOG)*, 40(6):1–16, 2021. 1, 2, 3, 4, 7

[43] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 2

[44] Abhimitra Meka, Rohit Pandey, Christian Haene, Sergio Orts-Escolano, Peter Barnum, Philip David-Son, Daniel Erickson, Yinda Zhang, Jonathan Taylor, Sofien Bouaziz, et al. Deep relightable textures: volumetric performance capture with neural rendering. *ACM Transactions on Graphics (TOG)*, 39(6):1–21, 2020. 3

[45] 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. *Communications of the ACM*, 65(1):99–106, 2021. 2, 16

[46] Rohit Pandey, Sergio Orts Escolano, Chloe Legendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. Total relighting: learning to relight portraits for background replacement. *ACM Transactions on Graphics (TOG)*, 40(4):1–21, 2021. 3

[47] Steven Parker, Peter Shirley, and Brian Smits. Single sample soft shadows. Technical report, Technical Report UUCS-98-019, Computer Science Department, University of Utah, 1998. 2, 4, 5, 14, 16

[48] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. 2, 3, 4

[49] Sida Peng, Zhen Xu, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable implicit neural representations for creating realistic avatars from videos. *arXiv preprint arXiv:2203.08133*, 2022. 1, 2, 3, 7, 13, 16, 17, 18, 20, 21

[50] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021. 1, 2, 3, 7, 14

[51] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM ToG*, 2017. 4, 5, 14

[52] Nick Roussopoulos, Stephen Kelley, and Frederic Vincent. Nearest neighbor queries. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 71–79, 1995. 4

[53] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2897, 2021. 3

[54] Carolin Schmitt, Simon Donne, Gernot Riegler, Vladlen Koltun, and Andreas Geiger. On joint estimation of pose, geometry and svbrdf from a handheld scanner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3493–3503, 2020. 3

[55] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE confer-*

ence on computer vision and pattern recognition, pages 4104–4113, 2016. 13

[56] Dario Seyb, Alec Jacobson, Derek Nowrouzezahrai, and Wojciech Jarosz. Non-linear sphere tracing for rendering deformed signed distance fields. *ACM Transactions on Graphics*, 38(6), 2019. 2, 17

[57] Ari Shapiro, Andrew Feng, Ruizhe Wang, Hao Li, Mark Bolas, Gerard Medioni, and Evan Suma. Rapid avatar capture and simulation using commodity depth sensors. *Computer Animation and Virtual Worlds*, 25(3-4):201–211, 2014. 2

[58] Nicholas Sharp and Alec Jacobson. Spelunking the deep: guaranteed queries on general neural implicit surfaces via range analysis. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 18

[59] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2, 3

[60] Jonathan Starck and Adrian Hilton. Virtual view synthesis of people from multiple view video sequences. *Graphical Models*, 67(6):600–620, 2005. 2

[61] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE computer graphics and applications*, 27(3):21–31, 2007. 1, 2

[62] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics*, 18(4):643–650, 2012. 2

[63] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. In *Acm Siggraph 2008 papers*, pages 1–9. 2008. 2

[64] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007. 4, 5, 7

[65] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *European conference on computer vision*, pages 1–19. Springer, 2022. 1, 2

[66] 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 preprint arXiv:2201.04127*, 2022. 2, 3

[67] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (TOG)*, 24(3):756–764, 2005. 1, 3

[68] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems*, 34:14955–14966, 2021. 2

[69] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (ToG)*, 37(2):1–15, 2018. 2

[70] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 16

[71] Yu-Ying Yeh, Koki Nagano, Sameh Khamis, Jan Kautz, Ming-Yu Liu, and Ting-Chun Wang. Learning to relight portrait images via a virtual light stage and synthetic-to-real adaptation. *ACM Transactions on Graphics (TOG)*, 41(6):1–21, 2022. 3

[72] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 2, 3, 7

[73] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[74] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)*, 40(1):1–17, 2021. 3

[75] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 2, 3, 5, 7, 16, 19

[76] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022. 2, 3, 5

[77] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15893–15903, 2022. 15

6. Addtional Results

6.1. Addtional Comparisons

In this section, we provide more qualitative results. Dynamic reconstruction and animation results under novel illuminations can be found in the supplementary video, which better demonstrates the effectiveness of our method to reconstruct the animatable human avatar than figures.

We provide more relighting and animation results in Figure 13 and Figure 14 where we show the reconstructed neural avatar being driven by novel human poses extracted from the AIST++ [40] dataset and relight with HDRi images from Polyhaven [2].

Additionally, we provide comparisons on monocular inputs for our method and baselines in Figure 7. Relighting4D [16] failed to reconstruct the normal and visibility MLP under this challenging monocular input, leading to incorrect relighting results and visibility estimation. Our method is able to recover the relightable properties and successfully relight the human avatar given even only monocular input.

6.2. Additional Ablation Studies

Table 3. **Ablation on other hyperparameters used in the model.** N_{st} and N_{st}^{vis} denote the number of sphere tracing steps for pixel-surface intersection and soft light visibility estimation respectively. K is the number of vertices sampled in the inverse warping process of GS-KNN and N_s is the number canonical material samples for volume rendering albedo α and roughness γ.

	Relighting			Visibility		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
$N_{st}=32$	20.84	0.810	0.201	**18.45**	**0.798**	0.193
$N_{st}=16$	**21.08**	**0.815**	**0.197**	18.19	**0.798**	**0.187**
$N_{st}=8$	20.67	0.773	0.242	17.50	0.726	0.266
$N_{st}=4$	20.43	0.680	0.350	15.88	0.578	0.402
$N_{st}=2$	19.22	0.589	0.428	13.38	0.494	0.465
$N_{st}^{vis}=8$	20.99	0.811	**0.196**	18.27	0.802	0.187
$N_{st}^{vis}=4$	**21.08**	**0.815**	0.197	18.19	0.798	0.187
$N_{st}^{vis}=2$	20.71	0.812	0.199	**18.63**	**0.806**	**0.184**
$N_{st}^{vis}=1$	20.55	0.804	0.202	18.49	0.804	0.186
$K=10$	**21.08**	**0.815**	**0.197**	18.19	0.798	**0.187**
$K=5$	20.76	0.810	0.203	18.17	0.795	0.195
$K=2$	20.78	0.812	0.198	**18.42**	**0.803**	0.188
$N_s=10$	20.65	0.813	0.199	17.98	0.800	0.195
$N_s=5$	20.80	0.811	**0.202**	18.42	0.797	0.194
$N_s=3$	**21.08**	**0.815**	0.197	18.19	0.798	0.187
$N_s=1$	20.81	0.813	0.200	**18.48**	**0.804**	**0.183**

Effectiveness of soft visibility scheme. We also study the effect of introducing the soft visibility scheme in Figure 9 and Table 3. All hyperparameters are fixed through out the experiments unless otherwise specified. We show the results of lighting a model with uniform material, where the BRDFs are equal everywhere, using a single area light source in Figure 9. This effectively produces ambient occlusion for the models. The quantitative comparison of all three variants can be found in Table 3. Note that although the "w/o soft" variants report higher PSNR and SSIMs, the visual quality of hard cast shadows is worse than ours, as indicated by the LPIPS metric. Additionally, we compare the effect of modeling visibility on material acquisition in Figure 11. The results indicate not modeling visibility leads to baked shadow on the diffuse albedo, while our method reconstructs a much more uniform diffuse map.

Other hyperparameter choices. In this section, we ablate the influence of other hyperparameters used in the model. The quantitative results are shown in Table 3. N_{st} denotes the number of sphere tracing steps for finding pixel-surface intersections. Table 3 shows that too low a number of steps for sphere tracing would lead to worse rendering quality, however, a too high number of steps only leads to diminishing returns. Thus we chose $N_{st}=16$ in all our experiments. N_{st}^{vis} is the number of sphere tracing steps for computing the soft visibility term, which shares a similar diminishing return pattern as N_{st}. The number of vertices K used in the inverse warping process and the number of samples N_s taken in canonical space for material properties (Section 7.6) does not significantly impact the rendering quality. Thus their values are empirically set to 10 and 3 in our experiments.

Table 4. **Ablation study on the accuracy of the proposed HDQ algorithm and sphere tracing procedure.** We provide comparison on the average absolute distance (denoted $\text{abs}(\text{SDF}(\boldsymbol{x}_s))$) of the computed pixel-surface intersection on four variants of the HDQ algorithm to compare its accuracy. Our method achieves an average accuracy of 0.00017m while the other variants exhibit at least two orders of magnitude higher error.

	ours	w/o HDQ	w/o d_{coarse}^{world}	w/o d_{fine}^{can}
$\text{abs}(\text{SDF}(\boldsymbol{x}_s))$	0.00017	0.20462	0.06774	0.00961

Accuracy of the proposed Hierarchical Distance Query and sphere tracing algorithm. In Table 4, we provide an additional ablation study on the accuracy of the proposed HDQ-ST (Hierarchical Distance Query and sphere tracing) algorithm. "w/o HDQ" variant uses volume rendering of 128 samples per ray for surface intersection and "w/o d_{fine}^{can}" and "w/o d_{coarse}^{world}" denote variant where sphere tracing is performed without the fine distance and coarse distance of the HDQ algorithm respectively. Our method achieves at least two orders of magnitude higher accuracy at 0.00017m compared to other variants.

Figure 7. **Qualitative Comparison of our method and baselines.** We present qualitative comparisons between our method and various baselines given only monocular input. The brightness difference between the ground truth and our method comes of the inherant scale ambiguity of recovering the environment lighting and albedo at the same time. The normal and visibility MLPs of Relighting4D [16] did not successfully capture the complex motion of the human avatar when only given monocular video as input thus producing overly smooth normal map that looks like a single-colored image. Note that NeRFactor is only trained on 1 frame and all 10 input views.

7. Additional Implementation Details

7.1. Details of the Datasets

MobileStage. To evaluate our approach on real-world humans, we collect a real-world multi-view dataset called *MobileStage*. This dataset is composed of 4 dynamic human videos collected using 36 synchronized mobile phone cameras at 1920×1080 resolution and 30 fps with complex indoor environment illuminations. We uniformly select 12 views for training in all experiments, the detailed frames selected are listed in Table 5. The characters perform complex motions, including talking, walking, and swinging, resulting in complex shading and shadow effects. We calibrate the camera poses of the multi-view system using COLMAP [55] and extract foreground human mask with [41]. The human poses corresponding to each of the characters are estimated using EasyMocap [1].

SyntheticHuman++ *SyntheticHuman++* contains 5 sequences of dynamic 3D human models with detailed shapes and complex materials. We use the original multi-view video in SyntheticHuman [49] for training, where 10 views of the synthetic characters were rendered with obvious self-occlusion and shading variations using Blender Cycles [18]. We uniformly select 6 views from all 20 views for testing. For the monocular setting, we select the fourth view for training.

To measure the quality of the visibility term, we additionally render all sequences with uniform diffuse material where the BRDFs in all directions are the same. This factors out the material from the rendering algorithm. For evaluating the performance of inverse rendering photometric properties, we

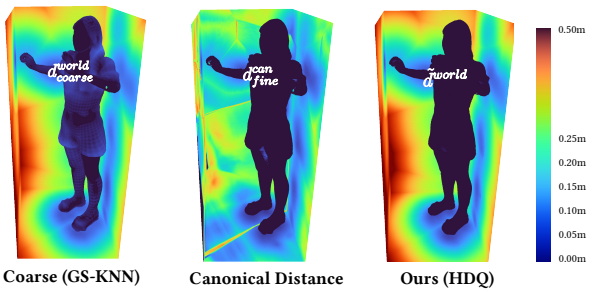

Figure 8. **Visualization of hierarchically queried distance**. Hot colors indicate large distance values. We visualize d^{world}_{coarse}, d^{can}_{fine} and $\tilde{d}^{world}$ on (a) surface intersection points, which should be zero everywhere, and (b) bounding box of the human model, which should reflect the real world space SDF. Note that the geometry is fixed for the comparison since we want to visualize the distance values on the correct geometry. The coarse distance d^{world}_{coarse} of near surfaces points is not strictly zero (indicated by the light blue regions on the body) due to misalignment between SMPL-H [51] and the neural avatar, and d^{can}_{fine} is incorrect in world space since it is defined by the canonical space network S. This leads to both of them reporting errornous geometry for the relighting task as seen in Figure 10. **In contrast to both, our proposed HDQ algorithm is well behaved both when near and far from the geometry.**

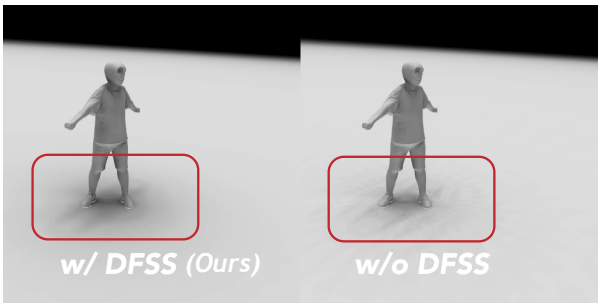

Figure 9. **Comparison of light visibility estimation algorithm**. We visualize the ambient occlusion result of different shadow algorithms by casting shadows from a dome lighting where the upper half of the environment map is set to a constant brightnerss. We use $N_{st} = 16$ sphere tracing iterations to generate shadows on the ground for "w/o DFSS" and "w/ DFSS (ours)". The "w/o DFSS" variant takes the binary light occlusion mask as the visibility term, resulting in unnatural hard shadow. **"w/ DFSS (ours)" generates realistic soft shadows by incorporating [47] onto neural SDF produced by ours Hierarchical Distance Query.**

also render the ground truth albedo and normal. We do not compare with the roughness term since Blender uses a different material model than ours. For novel pose synthesis, we used the motion capture data from AIST++ [40] along with motion capture data from the ZJU-MoCap Dataset [22, 50] (sequence *377*, *386*, *390*).

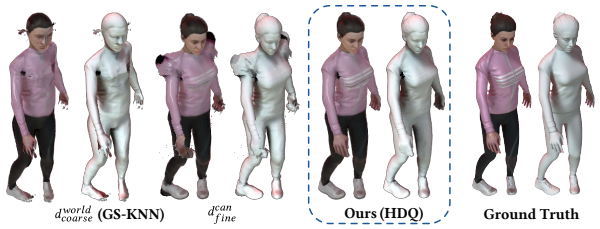

Figure 10. **Geometry quality of GS-KNN and HDQ**. Performing sphere tracing using only the canonical distance d^{can}_{fine} or coarse world distance d^{world}_{coarse} (GS-KNN) results in incorrect surface intersection $\boldsymbol{x}_s$ and soft visibility p_s, while tracing with our proposed Hierarchical Distance Query produces correct results.

Table 5. **Specific training settings.** N_{view} and N_{frame} represent the number of training views and training frames respectively. N^{total}_{view} denotes the total number of views of the dataset and N^{total}_{frame} denotes the total number of frames.

Dataset	Character	$N_{view}/N^{total}_{view}$	$N_{frame}/N^{total}_{frame}$
MobileStage	*dark*	**12** / 36	1600 / 2000
	purple	**12** / 36	600 / 700
	black	**12** / 36	300 / 400
	white	**12** / 36	300 / 600
SynthHuman (**monocular**)	*jody*	**1** / 20	100 / 100
	josh	**1** / 20	100 / 100
	megan	**1** / 20	100 / 100
	leonard	**1** / 20	100 / 100

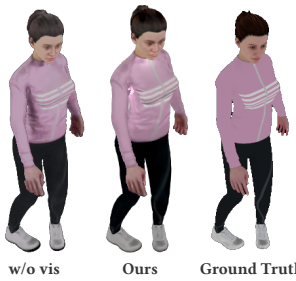

Figure 11. **Modeling visibility helps reconstructing diffuse albedo.** The "w/o vis" variant trained without computing light visibility bakes shadows onto the diffuse albedo, while our approach reconstructs high quality albedo.

7.2. Implementation Details for GS-KNN

Since vanilla KNN only produces unsigned distance to the closest K points, the sign of the distance value is determined by the dot product between $\boldsymbol{x} - v_k$ and the world space normal $\boldsymbol{n}_k$ corresponding to i_k using

$$d_k = \text{sign}(\text{dot}(\boldsymbol{x} - \boldsymbol{v}_k, \boldsymbol{n}_k))d_k. \quad (7)$$

Since the world space query point might be close to multiple parts of the posed vertices of the world space parametric model, which will lead to incorrect inverse LBS results, we

need to avoid using KNN query results from vertices that are far away in the canonical space. To avoid assigning the query point to multiple parts of the posed vertices $\mathcal{V}$ of the world space parametric model, we further update the KNN results using an approximation of the geodesic distance on the parametric human model. Specifically, we take the index of the closest parametric model vertex i_{min}, and compute the approximated canonical Euclidian distance between all other $K-1$ selected vertices by querying their distance $d_k^{\mathcal{V}}$ on the canonical parametric model $\mathcal{V}^{can}$ (e.g. T-Pose) using

$$i_k, d_k, \boldsymbol{v}_k, \boldsymbol{n}_k, \boldsymbol{w}_k = \begin{cases} i_{min}, d_{min}, \boldsymbol{v}_{min}, \boldsymbol{n}_{min}, \boldsymbol{w}_{min} & \text{, if } d_k^{\mathcal{V}} < T_d. \\ i_k, d_k, \boldsymbol{v}_k, \boldsymbol{n}_k, \boldsymbol{w}_k & \text{, otherwise.} \end{cases} \tag{8}$$

where d_{min}, $\boldsymbol{v}_{min}$, $\boldsymbol{n}_{min}$, $\boldsymbol{w}_{min}$ are the vertex properties corresponding to i_{min}, and T_d is the geodesic distance cutoff threshold, which is set to 0.1 through all experiments.

7.3. Implementation Details for Inverse LBS

With K sets of vertex properties produced by GS-KNN, the world-space query point is transformed into canonical space by the inverse LBS module. The human pose defines B body parts, which produces B transformation matrices $\{G_b\} \in SE(3)$. In the inverse LBS module, the world space points are transformed to canonical space using

$$\boldsymbol{T}^{world}(\boldsymbol{x}) = (\sum_{b=1}^{B} w_b(\boldsymbol{x}) G_b)^{-1}, \tag{9}$$

$$\boldsymbol{x'} = \boldsymbol{T}^{world}(\boldsymbol{x})\boldsymbol{x}, \tag{10}$$

where the blend weights w_b of the bth body part is the bth element of the blend weight vector $\boldsymbol{w} = (w_1, ..., w_B)$, which is computed from the KNN results $\mathcal{D}_K = \{d_0..., d_k\}$ and $\mathcal{W}_K = \{\boldsymbol{w}_0..., \boldsymbol{w}_k\}$ using $\mathcal{D}_K = \{d_0..., d_k\}$ and $\mathcal{W}_K = \{\boldsymbol{w}_0..., \boldsymbol{w}_k\}$:

$$\boldsymbol{w} = \sum_{k=1}^{K} \text{softmax}(\frac{-d_0}{2R_w^2}, ..., \frac{-d_K}{2R_w^2})_k \boldsymbol{w}_k, \tag{11}$$

where softmax denotes the softmax normalization operator, and R_w is the blend weight blending radius, which is set to 0.075 through all experiments following the distance-weighted blending [77].

$$\boldsymbol{x} = \boldsymbol{x'} + F_{\Delta \boldsymbol{x}}(\Theta_f, \boldsymbol{x'}). \tag{12}$$

Using Equation 10 and Equation 11, the world space query point $\boldsymbol{x}$ is transformed to canonical space $\boldsymbol{x'}$.

For world space normal direction, we compute the normalized gradient of the canonical deformed SDF and transform it back to world space using:

$$\boldsymbol{n}_s(\boldsymbol{x}_s) = (\boldsymbol{R}^{world})^{-1}(\nabla S(\boldsymbol{x}_s^{can})), \tag{13}$$

where $\boldsymbol{R}^{world}$ is the rotation component of $\boldsymbol{T}^{world}$ the inverse warping step of HDQ, and $\boldsymbol{x}_s^{can} = \boldsymbol{T}^{world}\boldsymbol{x}_s$ is the corresponding canonical space coordinate of the surface intersection $\boldsymbol{x}_s$.

7.4. Listing of the Surface Intersection Sphere Tracing Algorithm

In Algorithm 1 we provide the detailed procedure for finding surface intersections when performing sphere tracing on the hierarchically queried distance. The sphere tracing procedure on ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ is performed with t bounded by near and far distances n and f. Additionally, we add an offset o to the queried distance when updating t to skip through gazing angles, and we linearly interpolate for the surface intersection $\boldsymbol{x}_s$ when the sign of the queried distance changes. We empirically set $N_{st} = 16$ and $o = 0.02$ for all experiments. n and f are provided by intersecting the camera ray with the axis-aligned bounding box of the posed parametric human model corresponding to the neural avatar.

Input: $\boldsymbol{o}$, $\boldsymbol{d}$, N_{st}, n, f
Output: $\boldsymbol{x}_s$

$t \leftarrow n$ // Start from near plane
$t_s \leftarrow f$ // Surface intersection depth
$d_0 \leftarrow \text{inf}$ // Previous closest distance
$d_1 \leftarrow \text{inf}$ // Current closest distance
$d_c \leftarrow \text{inf}$ // All time closest distance
$d_t \leftarrow \text{inf}$ // Update of t during iteration
for $i \leftarrow 0$ **to** N_{st} **do**
  // Hierarchical distance query
  $d_0, d_1 \leftarrow d_1, \tilde{d}^{world}(\boldsymbol{o} + t\boldsymbol{d})$
  $d_0^{abs}, d_1^{abs} \leftarrow \text{abs}(d_0), \text{abs}(d_1)$
  // Update closest distance and intersection
  if $d_1^{abs} < d_c$ **then**
    $d_c \leftarrow d_1^{abs}$
    $t_s \leftarrow t$
  end
  // Linear interpolation upon sign change of SDF
  if $\text{sign}(d_0) \neq \text{sign}(d_1)$ **then**
    $t_s \leftarrow t - d_t \frac{d_1^{abs}}{d_0^{abs} + d_1^{abs}}$
  end
  // Prepare for next iteration
  $d_t \leftarrow d + o$
  $t \leftarrow t + d_t$
  // Constrain t to be within near and far plane
  $t \leftarrow \min(t, f)$
  $t \leftarrow \max(t, n)$
end
return $\boldsymbol{o} + t_s\boldsymbol{d}$

Algorithm 1: Surface intersection sphere tracing.

7.5. Listing of the Sphere Tracing Soft Visibility Algorithm

Light visibility is defined as whether a light ray from the light source to the surface intersection point is occluded by other parts of the geometry. This occlusion test can be achieved by performing the same Sphere Tracing algorithm for surface intersection and checking whether the returned intersection point $\boldsymbol{x}_s$ lies on the far plane f.

However, such a simple binary test can only produce hard shadow for a point light source [15], while perfect point light rarely exists and the more common area lights should produce soft shadows [75]. If we simply perform dense integration or monte-carlo integration to compute the soft light visibility values, the computational cost will be too high. To this end, we propose to integrate traditional distance field soft shadow algorithms [3,7,47] to our hierarchically queried driable neural distance field, which interpret the global meaning of the distance values as the penumbra coefficient for soft shadow effects with a small number of $\boldsymbol{\omega}_i$ samples.

Input: $\boldsymbol{o}$, $\boldsymbol{d}$, N_{st}^{vis}, n, f, a
Output: p

$t \leftarrow n$ // Start from near plane
$d_0 \leftarrow \text{inf}$ // Previous closest distance
$d_1 \leftarrow \text{inf}$ // Current closest distance
$d_t \leftarrow \text{inf}$ // Update of t during iteration
$p_s \leftarrow 1$ // Soft shadow penumbra coefficient
$R_s \leftarrow \sqrt{\frac{a}{\pi}}$ // Per solid angle $a = \pi R_s^2$
for $i \leftarrow 0$ **to** N_{st}^{vis} **do**
  // Hierarchical distance query
  $d_0, d_1 \leftarrow d_1, \tilde{d}^{world}(\boldsymbol{o} + t\boldsymbol{d})$
  // Compute penumbra coefficient
  $p_s \leftarrow \min(p_s, \frac{\max(d_1, 0)}{2tR_s})$
  // Prepare for next iteration
  $d_t \leftarrow d + o$
  $t \leftarrow t + d_t$
  // Constrain t to be within near and far plane
  $t \leftarrow \min(t, f)$
  $t \leftarrow \max(t, n)$
end
return p_s

Algorithm 2: Soft light visibility sphere tracing.

Algorithm 2 illustrates the soft visibility computation algorithm, in which $\boldsymbol{o}$ is set to the surface intersection $\boldsymbol{x}_s$, $\boldsymbol{d}$ is set to the light direction $\boldsymbol{\omega}_i$, n is set to 0.01, f is set to 10.0 and N_{st}^{vis} is set to 4 in all experiments. The ratio between the current distance d_1 and current ray depth t along with the area a (in solid angle) of the light source forms $p_s = \frac{d_1}{2t\sqrt{\frac{a}{\pi}}}$ following [7, 47].

7.6. Details on Network Queries of the Canonical Material Fields

To apply more supervision on the rendering process and the canonical material MLPs, we construct a sparse set of volume sampling $\mathcal{T}_s$ along the camera ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ near the computed surface intersection point $\boldsymbol{x}_s = \boldsymbol{o} + t_s\boldsymbol{d}$ with a fixed step size $t_{step} = 0.005$ and number of samples $N_s = 3$ using

$$\mathcal{T}_s = t_s + \{2\frac{n_s}{N_s}t_{step}, ..., 2\frac{N_s}{N_s}t_{step}\} - t_{step}. \tag{14}$$

Then, $\mathcal{T}_s$ is used to sample the material networks, on which the Volume Rendering algorithm in [49, 70] is applied to compute the final albedo $\alpha_s(\boldsymbol{x}_s)$ and roughness $\gamma_s(\boldsymbol{x}_s)$ corresponding to $\boldsymbol{x}_s$.

7.7. Network Structures

For the canonical space geometry and displacement field, we use 8 layer MLPs of width 256 for S and $F_{\delta\boldsymbol{x}}$ with ReLU and Softplus activation respectively. S takes positionally encoded (PE) [45] coordinate of resolution 8 as input and $F_{\delta\boldsymbol{x}}$ takes point input with 10 levels of PE along with the pose of the current human frame. We follow [49] to initialize S with [24]. For canonical material networks, we use 8-layer MLPs for A and Γ with ReLU activation of points input with 10 levels of PE. We additionally input the current human pose to increase its representational power. An illustration of the network structure is shown in Figure 12.

7.8. Loss Functions

We optimize our relightable and animatable neural human avatar by rendering the image with given camera poses and compare the pixel values L_o, which corresponds to ray $\boldsymbol{r}$, where $\boldsymbol{x}_s = \boldsymbol{o} + t_s\boldsymbol{d}$ and $\boldsymbol{\omega}_o = \boldsymbol{d}$, against the ground truth ones L_{gt}. The loss function is defined as

$$\mathcal{L}_d = \sum_{\boldsymbol{r}\in\mathcal{R}} \|L_o(\boldsymbol{r}) - L_{gt}(\boldsymbol{r})\|_2. \tag{15}$$

where $\mathcal{R}$ denotes all camera rays in the forward rendering process.

Following [49], we additionally regularize the residual displacement field F_Δ and canonical SDF S with the Eikonal term using

$$\mathcal{L}_{eik} = \sum_{\boldsymbol{x}\in\mathcal{X}} \lambda_{e0}(\|\Delta_{\boldsymbol{x}} S(\boldsymbol{x} + \Delta F(\boldsymbol{x}))\|_2 - 1) + \lambda_{e1}(\|\Delta_{\boldsymbol{x}+\Delta F(\boldsymbol{x})} S(\boldsymbol{x} + \Delta F(\boldsymbol{x}))\|_2 - 1) \tag{16}$$

where $\mathcal{X}$ denotes all the warped canonical points meeting the cutoff criteria $d_{coarse}^{world} <= \tilde{T}_d$ during the forward rendering process.

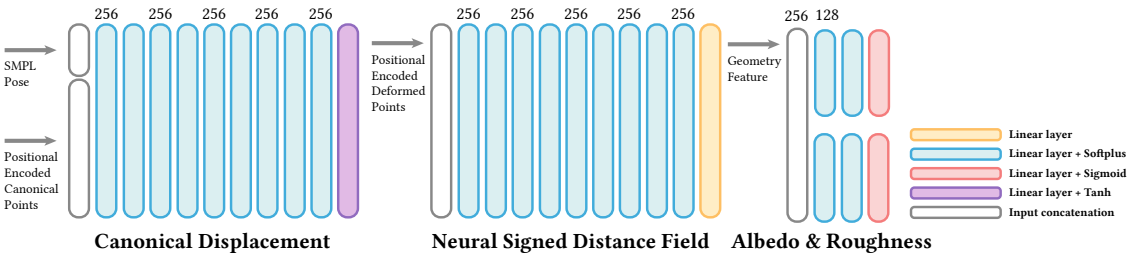

Figure 12. **Detailed network structure.**

Their [49] regularization on the magnitude of the residual displacement field F_Δ is also adopted in our work, which is defined as

$$\mathcal{L}_r = \sum_{\boldsymbol{x}\in\mathcal{X}} \|F_\Delta(\boldsymbol{x})\|_2. \tag{17}$$

We also define a mean intersection of union foreground mask loss on the silhouette $M_o(\boldsymbol{r})$ of the rendered image to regularize the training process as

$$\mathcal{L}_m = \sum_{\boldsymbol{r}\in\mathcal{R}} \frac{M_o(\boldsymbol{r})M_{gt}(\boldsymbol{r})}{M_o(\boldsymbol{r}) + M_{gt}(\boldsymbol{r})}. \tag{18}$$

To alleviate ambiguities in the physically-based rendering model, we follow [16] to add sparsity and smoothness regularizations on our material representation, the albedo MLP $\alpha = A(\boldsymbol{x})$ and the roughness MLP $\gamma = \Gamma(\boldsymbol{x})$. The regularization terms are defined as

$$\mathcal{L}_{ent} = \text{gaussian\_entropy}(\alpha(\mathcal{X})) \tag{19}$$

$$\mathcal{L}_a = \sum_{\boldsymbol{x}\in\mathcal{X}} \|\alpha(\boldsymbol{x}) - \alpha(\boldsymbol{x} + \Delta\boldsymbol{x})\|_2 \tag{20}$$

$$\mathcal{L}_r = \sum_{\boldsymbol{x}\in\mathcal{X}} \|\gamma(\boldsymbol{x}) - \gamma(\boldsymbol{x} + \Delta\boldsymbol{x})\|_2 \tag{21}$$

where $\text{gaussian\_entropy}(\alpha(\mathcal{X}))$ is the entropy of the Gaussian distribution of the albedo map α evaluated on the canonical points $\mathcal{X}$ as in [16] and $\Delta\boldsymbol{x}$ is a random perturbation sampled from a normal distribution $\Delta\boldsymbol{x} \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma = 0.02$ in all experiments.

7.9. Training

To make the optimization process more controllable, we separate the training process into two stages by postponing the training of the material and environment light probe after the geometry of the neural avatar has converged.

We first train the base geometry of the neural avatar including S, ΔF and a neural rendering network $C(\boldsymbol{x}, \boldsymbol{d}')$ as in [49] with the same volume rendering algorithm defined in [49]. The rendering network $C(\boldsymbol{x}, \boldsymbol{d})$ [49] takes the canonical query point $\boldsymbol{x}$ obtained in Equation 12 and the canonical view direction $\boldsymbol{d}' = \boldsymbol{R}^{world}\boldsymbol{d}$ as input where $\boldsymbol{R}^{world}$ is the rotation component of the world space transformation matrix $\boldsymbol{T}^{world}$ in Equation 9. The networks are trained using $\mathcal{L}_d$, $\mathcal{L}_{eik}$ and $\mathcal{L}_r$ for 200k iterations with and Adam optimizer [37] of learning rate $5e^{-4}$ exponentially annealed to $5e^{-6}$ during the first stage training. The full loss function of the first stage training is defined as

$$\mathcal{L}_{1st} = \lambda_d\mathcal{L}_d + \lambda_{eik}\mathcal{L}_{eik} + \lambda_r\mathcal{L}_r + \lambda_m\mathcal{L}_m \tag{22}$$

where λ_d, λ_{eik}, λ_r, λ_m are set to 1.0, 0.025, 0.1 and 0.01 respectively. And the loss weights in the eikonal loss $\mathcal{L}_{eik}$ definition λ_{e0} and λ_{e1} are set to 1.0 and 2.0 respectively.

During the second stage of the training process, we replace the volume rendering algorithm of [49] with Sphere Tracing [28] defined on the neural SDF of the human avatar, and the rendering network C is replaced with the physically-based renderer. Additional losses $\mathcal{L}_{ent}$, $\mathcal{L}_a$, $\mathcal{L}_r$ are added for regularization with the full loss defined as

$$\mathcal{L}_{2nd} = \mathcal{L}_{1st} + \lambda_{ent}\mathcal{L}_{ent} + \lambda_a\mathcal{L}_a + \lambda_r\mathcal{L}_r. \tag{23}$$

The second stage optimization is performed with an Adam optimizer [37] of learning rate $5e^{-4}$ exponentially annealed to $5e^{-6}$ for 25k iterations. Additionally, the starting learning rate on parameters of S and ΔF are tuned down to $1e-5$ if not otherwise specified to preserve the geometry during the first few iterations of the second stage. Loss weights λ_d, λ_{eik}, λ_r, λ_m, λ_{ent}, λ_a and λ_r are set to 10.0, 0.1, 0.1, 0.01, $5e^{-4}$, $5e^{-3}$ and $5e^{-3}$ respectively, where λ_{e0} and λ_{e1} are all set to 1.0.

The first and second stages are all trained with a batch size of 8 with 1024 rays per batch. On two NVIDIA RTX 3090 GPUs, the first stage training (200k iterations) takes about 20 hours and the second stage (25k iterations) takes about 10 hours. The forward rendering process of the second stage is adopted for inference when the neural avatar is driven with novel human poses or relit with novel lighting conditions through all experiments.

8. Additional Discussion

8.1. Discussion on Alternatives to HDQ

NLST [56] also proposes a method on performing sphere tracing on deformed signed distance fields with undeformed-

space distance query and integration of an ODE solver. However, their method is not directly applicable to animatable neural avatars since NLST requires the computation of the Jacobian of the inverse deformation field, which is unavailable in the neural deformation field used in our method. One could try to replace the single-direction displacement field with a bi-directional one like in [12], which might reduce the generalizability to novel human poses. But computing the Jacobian of a neural deformation field requires taking the backward of a large computation graph multiple times, which is time-consuming. Combined with the fact that NLST requires multiple subsets in one single query step, it would not be practical to naively combine the two. In comparison, our method does not require computing the Jacobian and directly returns a single well-approximated distance value for an articulated and neurally deformed distance field thanks to the adoption of the observation space Eikonal loss, which essentially also makes the neurally deformed field a valid distance field.

[58] could also be considered a viable approach for surface intersection and distance queries if the target of the query is a static neural implicit field. However, its core technique Range Analysis not directly applicable to the animated geometry of our reconstructed neural avatar, which requires computing the closest K points on a parametric model and performing space warping through blended skinning methods.

8.2. Difference from AniSDF

Similar to [49], we define the animatable geometry of the human avatar as the combination of a pose-driven deformation field $F(\boldsymbol{x})$ and a canonical SDF network $S(\boldsymbol{x})$, where the deformation field can be further decomposed into a pose-dependent displacement field $F_{\Delta \boldsymbol{x}}$ and KNN-based inverse Linear Blend Skinning (LBS) [39] module. We follow their canonical and deformation field setup to model the animatable human geometry. However, one key difference is that we focus on producing the correct world-space distance values, instead of simply warping query points back to canonical spaces. We intend to use the SDF values in a fixed step Sphere Tracing [29] framework where correct world space distance values are expected for convergence.

The key observation that makes the Hierarchical Distance Query scheme possible is that SDF values are locally deformation invariant under the inverse Linear Blend Skinning (LBS) [39] algorithm. The closer the query point is to the geometry, the less the value changes when transforming under novel human poses, where the distance value of a surface point will always be zero no matter the pose. This makes it possible to use the canonical SDF network $S(\boldsymbol{x})$ for near-surface distance queries, where the accuracy and detail of the distance are well-preserved, and then use the coarse KNN distance values for far-surface distance queries.

Specifically, due to the linearity of the inverse LBS algorithm, close-to-surface points are rarely mapped to the wrong canonical location since they exhibit very little ambiguity, while it is quite likely for far-from-surface points to fall into the region of confusion during the inverse wrapping process. Thus d^{can}_{fine} is rarely inaccurate for close-to-surface query points, while it is more likely to be inaccurate for far-from-surface ones. Utilizing this property of the SDF value, we can interpret the fine canonical distance value d^{can}_{fine} as a good approximation for the actual world-space fine level distance d^{world} when the query points are close to the neural avatar, while the coarse world space distance d^{world}_{coarse} could also serve as a good approximation for the world-space distance d^{world} because the parametric human model is aligned to the neural avatar's geometry.

Table 6. **Addtional quantitative comparison on all characters of SyntheticHuman++.** We report the result of NeRFactor and NeRFactor* on the first frame of each sequence. NeRFactor [75] and NeRFactor* failed to converge on the first frame of *megan* and *leonard*.

		Normal	**Diffuse Albedo**			**Relighting**			**Visibility**		
Character	Method	Degree ↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
jody	Ours	10.98	25.88	0.925	0.144	21.57	0.853	0.168	20.53	0.869	0.142
	Relighting4D*	27.65	16.40	0.827	0.229	20.77	0.826	0.205	15.47	0.785	0.241
	Relighting4D	76.73	16.40	0.827	0.229	17.25	0.752	0.249	9.51	0.604	0.313
	NeRFactor*	33.19	17.74	0.860	0.201	21.66	0.867	0.224	16.12	0.752	0.283
	NeRFactor	79.26	17.74	0.860	0.201	20.31	0.862	0.181	11.78	0.755	0.227
	AniSDF	13.61	16.02	0.815	0.230	19.63	0.818	0.223	-	-	-
josh	Ours	12.88	34.46	0.968	0.072	24.72	0.897	0.206	20.04	0.858	0.149
	Relighting4D*	26.56	29.32	0.917	0.153	24.48	0.881	0.229	16.02	0.790	0.233
	Relighting4D	101.0	29.31	0.917	0.153	23.86	0.830	0.273	3.25	0.554	0.363
	NeRFactor*	35.39	29.14	0.926	0.167	24.68	0.895	0.228	14.85	0.729	0.291
	NeRFactor	24.57	29.14	0.926	0.167	24.65	0.896	0.217	15.96	0.791	0.245
	AniSDF	16.34	30.28	0.924	0.164	20.76	0.843	0.260	-	-	-
megan	Ours	13.09	22.67	0.873	0.198	18.55	0.790	0.234	19.69	0.814	0.167
	Relighting4D*	31.98	22.91	0.857	0.223	17.25	0.742	0.273	14.83	0.713	0.278
	Relighting4D	98.05	22.93	0.857	0.222	17.26	0.671	0.307	5.47	0.364	0.433
	NeRFactor*	-	15.11	0.569	0.394	14.06	0.490	0.472	7.41	0.341	0.538
	NeRFactor	-	15.11	0.569	0.394	14.06	0.490	0.472	7.41	0.341	0.538
	AniSDF	15.18	16.91	0.788	0.258	11.51	0.686	0.316	-	-	-
leonard	Ours	12.81	33.04	0.968	0.064	25.92	0.903	0.218	20.53	0.852	0.160
	Relighting4D*	31.33	30.18	0.938	0.129	26.03	0.891	0.242	14.57	0.766	0.255
	Relighting4D	99.54	30.18	0.938	0.130	25.14	0.845	0.273	3.24	0.533	0.388
	NeRFactor*	-	26.93	0.912	0.143	23.77	0.779	0.326	7.09	0.500	0.435
	NeRFactor	-	26.93	0.912	0.143	23.77	0.779	0.326	7.09	0.500	0.435
	AniSDF	13.75	25.31	0.920	0.155	18.30	0.849	0.246	-	-	-

Table 7. **Addtional quantitative comparison on all characters of SyntheticHuman++ under the monocular setting.**

		Normal	**Diffuse Albedo**			**Relighting**			**Visibility**		
Character	Method	Degree ↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
jody	Ours	16.83	24.09	0.877	0.180	21.08	0.815	0.197	18.19	0.798	0.187
	Relighting4D*	25.16	19.57	0.799	0.261	20.79	0.766	0.245	17.31	0.736	0.273
	Relighting4D	80.80	19.57	0.799	0.262	20.28	0.778	0.233	16.16	0.753	0.278
	AniSDF	18.65	16.25	0.752	0.284	18.65	0.746	0.274	-	-	-
josh	Ours	20.09	26.67	0.905	0.142	24.52	0.873	0.224	17.46	0.761	0.222
	Relighting4D*	24.93	29.61	0.919	0.165	23.95	0.863	0.240	17.29	0.737	0.268
	Relighting4D	81.23	29.61	0.919	0.165	24.06	0.861	0.269	15.88	0.750	0.297
	AniSDF	21.91	26.54	0.879	0.226	19.88	0.790	0.294	-	-	-
megan	Ours	19.64	18.63	0.796	0.262	18.33	0.742	0.246	17.70	0.707	0.222
	Relighting4D*	29.51	22.38	0.798	0.286	17.21	0.696	0.285	16.29	0.640	0.316
	Relighting4D	83.40	22.38	0.798	0.286	17.59	0.705	0.299	16.30	0.648	0.354
	AniSDF	19.80	26.29	0.905	0.179	20.77	0.825	0.274	-	-	-
leonard	Ours	18.30	24.28	0.912	0.121	25.88	0.893	0.231	18.45	0.779	0.215
	Relighting4D*	25.08	29.90	0.940	0.130	25.28	0.884	0.245	17.50	0.726	0.286
	Relighting4D	81.55	29.90	0.940	0.130	25.48	0.879	0.270	16.40	0.753	0.280
	AniSDF	21.08	16.98	0.710	0.332	13.85	0.619	0.346	-	-	-

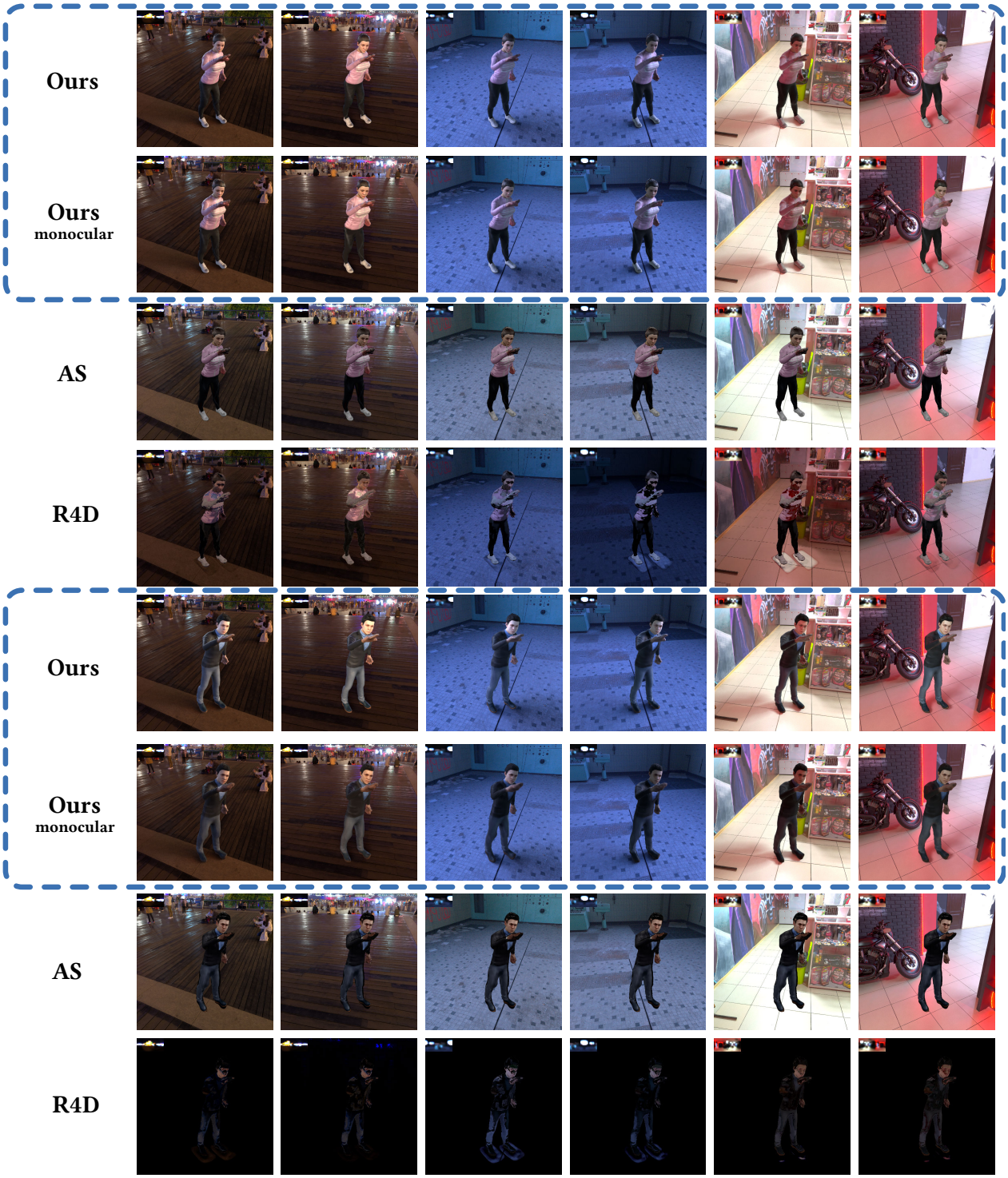

Figure 13. **More novel pose relighting results on the SyntheticHuman dataset.** "AS" denotes AniSDF [49]. "R4D" denotes Relighting4D [16]. All compared relighting methods compute shadows on the ground plane. The visibility MLP of [16] failed to generalize to far-from-human points.

Figure 14. **More novel pose relighting results on the MobileStage dataset.** "AS" denotes AniSDF [49]. "R4D" denotes Relighting4D [16]. All compared relighting methods compute shadows on the ground plane. The visibility MLP of [16] failed to generalize to far-from-human points.