Animatable Neural Implicit Surfaces for Creating Avatars from Videos

Sida Peng[1] Shangzhan Zhang[1] Zhen Xu[1] Chen Geng[1]
Boyi Jiang[2] Hujun Bao[1] Xiaowei Zhou[1]
[1]Zhejiang University [2]Image Derivative Inc

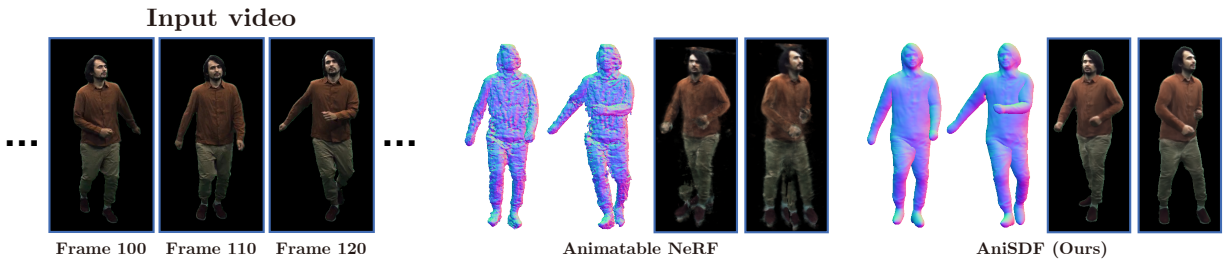

Figure 1. Given a monocular video of a performer, our method reconstructs high-quality geometries and renders photorealistic images under novel human poses, which outperforms Animatable NeRF [56]. More results can be found at https://zju3dv.github.io/animatable_sdf/.

Abstract

This paper aims to reconstruct an animatable human model from a video of very sparse camera views. Some recent works represent human geometry and appearance with neural radiance fields and utilize parametric human models to produce deformation fields for animation, which enables them to recover detailed 3D human models from videos. However, their reconstruction results tend to be noisy due to the lack of surface constraints on radiance fields. Moreover, as they generate the human appearance in 3D space, their rendering quality heavily depends on the accuracy of deformation fields. To solve these problems, we propose Animatable Neural Implicit Surface (AniSDF), which models the human geometry with a signed distance field and defers the appearance generation to the 2D image space with a 2D neural renderer. The signed distance field naturally regularizes the learned geometry, enabling the high-quality reconstruction of human bodies, which can be further used to improve the rendering speed. Moreover, the 2D neural renderer can be learned to compensate for geometric errors, making the rendering more robust to inaccurate deformations. Experiments on several datasets show that the proposed approach outperforms recent human reconstruction and synthesis methods by a large margin.

1. Introduction

Reconstruction and rendering of dynamic humans have many applications in VR and AR, such as video games, sports broadcasting, and telepresence. In all these applications, the reconstructed human models are expected to have high-quality geometry and appearance to enable photorealistic rendering of images under novel views and novel human poses. Moreover, high rendering efficiency is desired to enable real-time interactive applications.

Traditional methods [10, 11, 14, 18] represent the human geometry as a mesh and store the appearance in 2D texture maps. For example, given a dense array of cameras, [18] reconstructs the mesh with multi-view stereo methods [67,68] and infers texture maps based on the spherical gradient illumination [16]. Although this pipeline achieves impressive reconstruction results and efficient rendering, it requires dense camera arrays to ensure reliable stereo matching.

Some recent works [36,56–58] reconstruct humans from sparse multi-view videos by integrating temporal information into canonical models. Animatable NeRF [56] represents the human model as a neural radiance field (NeRF) [45] in the canonical space. To learn this representation from videos, it defines deformation fields that transform 3D points in the observation space to the canonical space, and renders the canonical human model into video frames. Although Animatable NeRF can recover reasonably well human models from videos, it has several limitations. First, its recovered surface tends to be noisy due to the lack of surface constraints on the learned geometry (volume density), as shown in Figure 1. Second, to render images under a particular human pose, it produces the deformation field by fitting 3D points to the parametric human model in the observation space, which is slow. Moreover, for unseen human poses, it tends to generate blurry images, as observation-to-

canonical deformations may be inaccurate.

In this paper, we introduce a human representation, named Animatable Neural Implicit Surface (AniSDF), for reconstruction and rendering of dynamic humans. Specifically, we use a signed distance field (SDF) to represent the human geometry in the canonical space. Compared with the density field, SDF has a well-defined surface at the zero-level set, which facilitates more direct regularization on the geometry learning. A challenge here is how to learn the canonical SDF from the video. Surface rendering [22, 35, 39] is a classical way to render SDFs. However, as there could be complex human motions between the observation and canonical spaces, it is difficult to find surface points along the camera ray in the observation space, as shown in [69]. To solve this problem, we utilize a deformation field to represent the non-rigid transformations between the canonical and observation spaces. Then, the SDF-based volume rendering techniques [78, 84] are used to render the SDF model in the observation space. Experiments demonstrate that this scheme effectively learns the canonical SDF from the video and produces high-quality geometries.

We additionally adopt two strategies to improve the efficiency and quality of image synthesis under novel human poses. First, based on the learned SDF, we develop a surface-guided rendering pipeline that leverages the reconstructed geometry to establish the observation-to-canonical deformation field only near the surface, enabling the efficient rendering of animated human models. Second, to improve the rendering quality under novel human poses, we additionally represent the human appearance with a neural feature field [48] and a 2D neural renderer, which performs the image-space rendering. This is motivated by our observation that if the appearance is generated in 3D space, the final rendering quality heavily relies on the observation-to-canonical deformation. By deferring the appearance generation to 2D image space, our model can be trained to compensate for inaccurate deformations.

In summary, we propose a novel human representation named AniSDF, which enables high-quality geometry reconstruction and efficient photorealistic rendering. We evaluate our approach on both monocular and multi-view videos. Across all videos, our approach outperforms prior works by a large margin on 3D reconstruction and image synthesis. Furthermore, our approach has a much faster rendering speed than NeRF-based methods [56, 57].

2. Related work

3D human reconstruction. Traditional human reconstruction methods require complicated hardware that is expensive for daily use, such as depth sensors [10, 14, 47, 75] or dense camera arrays [10, 11, 18]. To reduce the requirement on the capture device, [46, 62, 63, 88] train a network to reconstruct human models from single RGB images. However, their generalization ability remains as an issue. Recently, [45, 49, 51, 72] reconstruct 3D scenes from images with differentiable renderers. Neural radiance fields [45] employs the volume rendering to learn density and color fields from dense camera views. [36, 56–58] augment neural radiance fields with deformation fields, enabling them to reconstruct 3D human models from sparse multi-view videos. However, their reconstruction results are often incomplete and noisy since there is no regularization on the reconstructed surface. [35, 78, 84] represent the scene geometry with a signed distance field, which leads to improvements on the performance of 3D reconstruction. They focus on the reconstruction of static scenes and require dense views to optimize the scene representation.

3D human animation. The linear blend skinning model (LBS) and its variants [28, 33] are time-tested techniques for animating 3D human models, which transform surface points based on the skeleton transformations. Skinned multi-person linear model (SMPL) [41] learns the LBS parameters on a set of 3D human scans for human animation. Based on the SMPL model, some methods [13, 15, 25, 27, 30, 55, 73] reconstruct animatable human characters from sparse camera views. However, their reconstruction results often lack details, since SMPL only models the minimally clothed human. To improve the reconstruction quality, [6, 23, 83] combine the SMPL model with implicit functions. Taking a point cloud as input, [6] predicts the occupancy field and semantic correspondences to the SMPL model, which is used to fit the reconstructed geometry to the SMPL model. Given point cloud sequences, [9, 44, 64, 77] define the human geometry in the canonical space and utilize the LBS model to transform point clouds to the canonical space for training. Their reconstructed human models are animated based on the skeleton-driven framework.

Novel pose synthesis. Some methods [4, 8, 43, 65, 66] synthesize images under novel human poses with image-to-image translation techniques, which produce photorealistic rendering results. However, these methods have difficulty in generating high-fidelity images under novel views. To overcome this problem, [32, 37, 38, 59, 70, 76, 80, 85] explicitly consider the underlying 3D representation when synthesizing images. [80] takes a human point cloud as input and renders it into feature maps. Then, a 2D network is used to generate the image from feature maps. To improve the multi-view consistency, [34, 36, 40, 45, 50, 56, 57, 74, 81] generate the scene appearance in 3D space and render images with volume rendering techniques. Recently, [21, 48] combine volume rendering and image-space rendering. They define feature fields in 3D space and utilize volume rendering to produce 2D feature maps, which are interpreted into images with a 2D neural renderer.

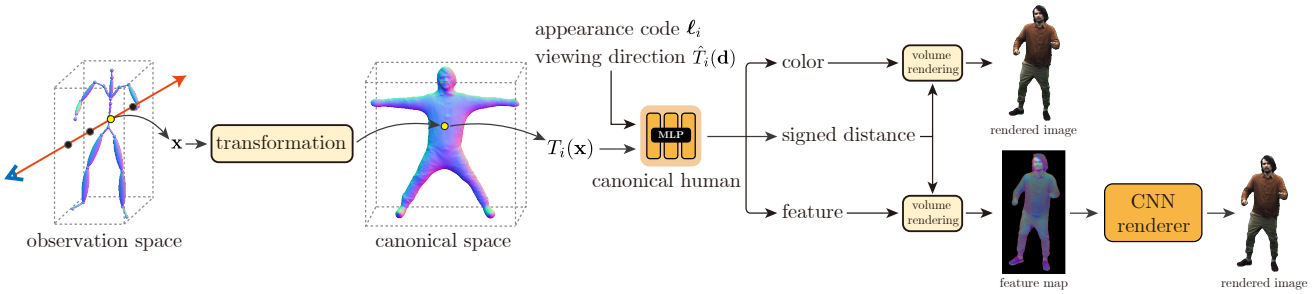

Figure 2. **Overview of AniSDF.** We represent the human geometry with a signed distance field (SDF) and model the appearance with two ways: i) a color field or ii) a neural feature field with a 2D neural renderer. To render images, we sample points along camera rays and transform them to the canonical space, which are fed into MLP networks to predict signed distances, colors and features. The images can be obtained by i) accumulating the colors with volume rendering techniques, or ii) accumulating the feature field into 2D feature maps, which are then interpreted into images with a CNN renderer. The learned SDF can be used to guide the point sampling in volume rendering.

3. Method

This paper aims to reconstruct the human geometry and appearance from a sparse multi-view video, and render free-viewpoint videos given human motions. The cameras are synchronized, and the camera parameters are known. Following [56, 57], we assume that the human masks and 3D human poses are given. We use the human mask to set the values of the background image pixels as zero.

Figure 2 shows the overview of the proposed model. AniSDF represents the human geometry and appearance as signed distance and color fields in the canonical space, and models the dynamic human in the input video with deformation fields (Section 3.1). Then we learn the representation from the RGB video with volume rendering (Section 3.2). Based on the learned human model, we are able to efficiently render images under input human poses (Section 3.3). To improve the rendering quality, AniSDF additionally represents the human appearance with neural feature field and 2D neural renderer (Section 3.4).

3.1. Modeling dynamic human bodies

To reconstruct the human model from the input video, inspired by [53, 56, 58], we represent the dynamic human body in the video with a canonical human model and a set of deformation fields. Given an observation-space point $\mathbf{x}$ at the video frame i, a deformation field T_i transforms $\mathbf{x}$ to the canonical space, which is then fed into the canonical human model to obtain its signed distance and color.

Canonical human model. In contrast to previous methods using a density field [53, 56, 58], AniSDF represents the human geometry with a signed distance field [35, 52], which predicts a signed distance s for any 3D point in the canonical space. Denote the canonical-space geometry model as F_s. Then, the geometry model of dynamic human body at video frame i is defined as:

$$(s_i(\mathbf{x}), \mathbf{z}_i(\mathbf{x})) = F_s(T_i(\mathbf{x})), \tag{1}$$

where $\mathbf{z}_i(\mathbf{x})$ is the geometry feature, and F_s is implemented as an MLP network with nine layers.

Similar to [35], the canonical-space color model takes spatial location, geometry feature, normal, and viewing direction as inputs. To better approximate the radiance function, we transform the observation-space viewing direction $\mathbf{d}$ to the canonical space using the deformation $\hat{T}_i$. We also take a per-frame latent code $\boldsymbol{\ell}_i$ as input to encode the scene state at frame i. With the canonical-space color model $F_\mathbf{c}$, the observation-space color model at frame i is defined as:

$$\mathbf{c}_i(\mathbf{x}) = F_\mathbf{c}(T_i(\mathbf{x}), \mathbf{z}_i(\mathbf{x}), \mathbf{n}_i(\mathbf{x}), \hat{T}_i(\mathbf{d}, \mathbf{x}), \boldsymbol{\ell}_i), \tag{2}$$

where the normal $\mathbf{n}_i(\mathbf{x})$ is calculated as the gradient of the signed distance $s_i(\mathbf{x})$ at point $T_i(\mathbf{x})$. The deformation T_i and $\hat{T}_i$ for spatial points and viewing directions will be described later. $F_\mathbf{c}$ is implemented as an MLP network with five layers. For novel human poses, we use the appearance code at the first frame for the color prediction. More details can be found in the supplementary material.

Deformation field. To establish observation-to-canonical correspondences, AniSDF decomposes the human motion in the video into articulated and non-rigid deformations, which is represented by the LBS model [33, 56] and a neural displacement field, respectively.

Specifically, for an observation-space point $\mathbf{x}$ at frame i, we have a 3D human skeleton that produces K transformation matrices $\{G_i^k\} \in SE(3)$. Based on the linear blend skinning algorithm [33, 56], we can transform the point to the canonical space using

$$\bar{\mathbf{x}}' = \left(\sum_{k=1}^{K} w_i^k(\mathbf{x}) G_i^k \right)^{-1} \bar{\mathbf{x}}, \tag{3}$$

where $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ are homogeneous coordinates of $\mathbf{x}$ and $\mathbf{x}'$, and $w_i^k(\mathbf{x})$ is the blend weight of k-th part. Given the transformed point $\mathbf{x}'$, we use a displacement field to deform it to the surface. Denote the displacement field as

$F_{\Delta \mathbf{x}} : (\mathbf{x}, \boldsymbol{\psi}_i) \rightarrow \Delta \mathbf{x}_i$, where $\boldsymbol{\psi}_i$ is a learnable latent code for frame i. The deformation field $T_i(\mathbf{x})$ is defined as:

$$T_i(\mathbf{x}) = \mathbf{x}' + F_{\Delta \mathbf{x}}(\mathbf{x}', \boldsymbol{\psi}_i), \tag{4}$$

where $F_{\Delta \mathbf{x}}$ is implemented as an MLP network with nine layers. In practice, we use SMPL [41] as the body model. The blend weight $\mathbf{w}_i(\mathbf{x})$ is obtained by retrieving the blend weight of the nearest SMPL vertex at frame i, similar to [7, 23, 56]. Note that other parametric human models [54, 61, 82] can also be used in our approach.

The observation-space viewing direction is transformed to the canonical space based on the LBS model. Denote the weighted sum of transformation matrices in Equation (3) as $[R_i^*(\mathbf{x}); t_i(\mathbf{x})] = \sum w_i^k(\mathbf{x}) G_i^k$. The deformation $\hat{T}_i(\mathbf{d}, \mathbf{x})$ for the viewing direction is defined as:

$$\hat{T}_i(\mathbf{d}, \mathbf{x}) = R_i^*(\mathbf{x})\mathbf{d}, \tag{5}$$

where $R_i^*(\mathbf{x})$ is a 3×3 matrix.

Note that our approach does not optimize blend weights during training, which is different from [56], because we observe that optimizing blend weights makes our model prone to local optima on some videos.

3.2. Training

To learn AniSDF from the input video, we need to render it into images with a differentiable renderer, and minimize the difference between rendered and observed images. Inspired by [78, 84], we utilize the volume rendering scheme in [84] to render dynamic signed distance fields. Given an image pixel at frame i, we first sample N_k points $\{\mathbf{x}_k\}_{k=1}^{N_k}$ along its camera ray $\mathbf{r}$ between near and far bounds. Then, AniSDF predicts signed distances and colors at these points. To perform volume rendering, we convert signed distance $s_i(\mathbf{x}_k)$ into volume density using

$$\sigma_i(\mathbf{x}) = \begin{cases} \frac{1}{\beta}\left(1 - \frac{1}{2}\exp\left(\frac{s_i(\mathbf{x})}{\beta}\right)\right) & \text{if } s_i(\mathbf{x}) < 0, \\ \frac{1}{2\beta}\exp\left(-\frac{s_i(\mathbf{x})}{\beta}\right) & \text{if } s_i(\mathbf{x}) \geq 0, \end{cases} \tag{6}$$

where β is a learnable parameter. Finally, the rendered pixel color $\tilde{\mathbf{C}}_i(\mathbf{r})$ is calculated using the numerical quadrature:

$$\tilde{\mathbf{C}}_i(\mathbf{r}) = \sum_{k=1}^{N_k} \alpha_i(\mathbf{x}_k) \prod_{j<k} (1 - \alpha_i(\mathbf{x}_j)) \mathbf{c}_i(\mathbf{x}_k), \tag{7}$$

where $\alpha_i(\mathbf{x}_k) = 1 - \exp(-\sigma_i(\mathbf{x}_k)\delta_k)$, and δ_k is the distance between adjacent sampled points $||\mathbf{x}_{k+1} - \mathbf{x}_k||_2$.

In practice, the near and far bounds are obtained by intersecting the camera ray with the 3D bounding box of the SMPL model, and we use a stratified sampling approach [45] to sample points. The number of sampled points N_k is set as 64 in all experiments.

Training losses. We optimize the model parameters by minimizing the difference between the rendered pixel colors and observed pixel colors. In addition, the mask loss and the Eikonal term [17] are used for supervision. We also add a regularizer for neural displacement fields. Specifically, the color loss is defined as:

$$L_{\text{rgb}} = \sum_{r \in \mathcal{R}} \|\tilde{\mathbf{C}}_i(\mathbf{r}) - \mathbf{C}_i(\mathbf{r})\|_2, \tag{8}$$

where $\mathbf{C}_i(\mathbf{r})$ is the observed pixel color at frame i, and $\mathcal{R}$ is the set of rays of sampled image pixels. To supervise the SDF with the mask, we find the minimal SDF value $s_i^{\mathbf{r}}$ of sampled points along the camera ray $\mathbf{r}$ and apply the binary cross entropy loss BCE:

$$L_{\text{mask}} = \sum_{r \in \mathcal{R}} \text{BCE}(\text{sigmoid}(-\rho s_i^{\mathbf{r}}), M_i(\mathbf{r})), \tag{9}$$

where $M_i(\mathbf{r}) \in \{0, 1\}$ is the ground-truth mask value. Similar to [35], we set ρ as 50 and multiply it by 2 every 10000 iterations. The number of multiplications is up to 5.

We sample a set of points $\mathcal{X}_i$ in the observation space and apply the Eikonal term on these sampled points:

$$L_{\text{E}} = \sum_{\mathbf{x} \in \mathcal{X}_i} (\|\nabla F_s(T_i(\mathbf{x}))\|_2 - 1)^2. \tag{10}$$

To regularize the neural displacement fields, we additionally sample a set of points $\mathcal{X}_i'$ in the canonical space and then apply the regularization:

$$L_{\Delta \mathbf{x}} = \sum_{\mathbf{x} \in \mathcal{X}_i'} \|F_{\Delta \mathbf{x}}(\mathbf{x}, \boldsymbol{\psi}_i)\|_2. \tag{11}$$

The final loss function is defined as:

$$L = L_{\text{rgb}} + L_{\text{mask}} + \lambda_1 L_{\text{E}} + \lambda_2 L_{\Delta \mathbf{x}}, \tag{12}$$

where we set λ_1 as 0.1 and λ_2 as 0.01. We adopt the Adam optimizer [29] with a learning rate that starts from $5e^{-4}$ and decays exponentially to $5e^{-5}$ along the optimization.

3.3. Surface-guided rendering

After training, AniSDF can be used to synthesize images under input human poses based on the observation-to-canonical deformation, which first samples points along camera rays in the observation space and then transforms points to the canonical space using the inverse LBS. Previous methods [36, 56] adopt a similar rendering pipeline, but they are slow due to two factors. First, they require to sample a lot of points along the camera ray. Second, they calculate blend weights for sampled points by finding the nearest surface point on the SMPL mesh under the target human pose, which is time-consuming.

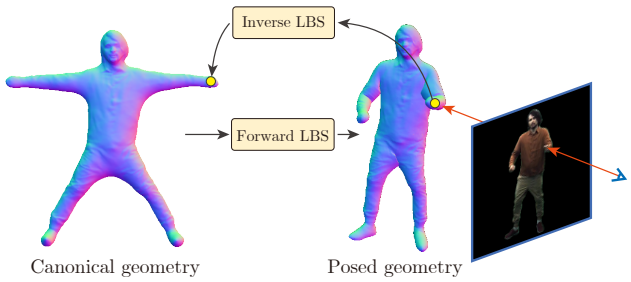

Figure 3. **Surface-guided rendering.** Given the extracted geometry, we use the forward LBS to deform it to the observation space. Then the posed geometry is used to guide the point sampling and establish the observation-to-canonical correspondences.

To improve the rendering speed, we utilize the learned SDF to efficiently establish observation-to-canonical correspondences, as shown in Figure 3. First, a human mesh is extracted by applying the Marching Cubes algorithm [42] to the signed distance field in the canonical space. To animate the human mesh, its vertices are assigned the blend weight of the closest point on the SMPL mesh under the canonical pose. Given a human pose, we perform the forward LBS to deform the canonical mesh to the observation space. Then, for each pixel, we obtain the surface point on the deformed mesh using the rasterization [60] and sample N'_k points near the surface along the camera ray. The sampled points are assigned the blend weight of corresponding surface point and are transformed to the canonical space using the inverse LBS. Finally, the transformed points are fed into the networks to predict the signed distances and colors, which are accumulated into the pixel color with volume rendering. In practice, the number of sampled points N'_k is set as 5 in all experiments. The sampled interval along the ray is set as 1 cm empirically.

3.4. Rendering with neural feature fields

We observe that rendering with color fields could produce blurry images under unseen human poses, as the observation-to-canonical deformation field may be inaccurate. To improve the rendering quality, AniSDF additionally represents the human appearance with a neural feature field and a 2D neural renderer, which performs the image-space rendering.

Specifically, the neural feature field is defined in the canonical space and is implemented as an MLP network that maps any 3D point to a feature vector. To render images, we first sample points along camera rays and transform them to the canonical space using the strategy in Section 3.3. Then, the networks map the points to signed distances and feature vectors, which are finally accumulated into a feature map using the volume rendering. A 2D CNN processes the feature map to output an image and a human mask, which is used to extract the foreground from the synthesized image.

As shown in [76, 80], 2D CNNs can be trained to correct for geometric errors. To ensure the inter-view and inter-frame consistency of rendered images, we adopt a shallow CNN as the neural renderer. The detailed network architecture is described in the supplementary material.

Following [80], we adopt the perceptual loss [26] and L1 loss to supervise the image generation. Denote the synthesized image as $\tilde{I}$. The image loss function is defined as:

$$L'_{\text{image}} = \|\tilde{I} - I\|_2 + \|F_{\text{vgg}}(\tilde{I}) - F_{\text{vgg}}(I)\|_2, \tag{13}$$

where I is the observed image, and F_{vgg} extracts feature maps from the second and fourth layer of the pretrained VGG-19 network [71]. To supervise the mask prediction, we use the binary cross entropy loss BCE:

$$L'_{mask} = \text{BCE}(\tilde{M}, M), \tag{14}$$

where $\tilde{M}$ and M are predicted and ground-truth masks. The coefficient weights of L'_{image} and L'_{mask} are both set to 1.

Note that the neural feature field and 2D CNN are not jointly trained with the signed distance field and color field, because we find that the joint optimization degrades the quality of our reconstruction results.

4. Experiments

4.1. Datasets and metrics

SyntheticHuman is a synthetic dataset that contains 7 animated 3D characters from RenderPeople [3] and Mixamo [2]. 4 characters perform rotation while holding A-pose, which are rendered into monocular videos. Another 3 characters perform random actions, which are rendered with four cameras. All video frames and camera views are used for training. This dataset is only used to evaluate the performance on 3D reconstruction. We describe more details of this dataset in the supplementary material.

Human3.6M [24] captures human performers with 4 cameras and estimate the human poses with a marker-based motion capture system. We follow the experimental protocol in [56] to evaluate the image synthesis.

MonoCap consists of two videos from DeepCap dataset [20] and two videos from DynaCap dataset [19], which are captured by dense camera views and provide the human masks and 3D human poses. We use one camera view for training and select ten uniformly distributed cameras for test. We select a clip of each video to perform experiments. Each clip has 300 frames for training and 300 frames for evaluating novel pose synthesis, respectively. More details of this dataset can be found in the supplementary material.

Metrics. For 3D reconstruction, we follow [62] to use two metrics: point-to-surface Euclidean distance (P2S) and Chamfer distance (CD). Units for the two metrics are in cm. For image synthesis, we follow [45] to evaluate our method using three metrics: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and LPIPS [87].

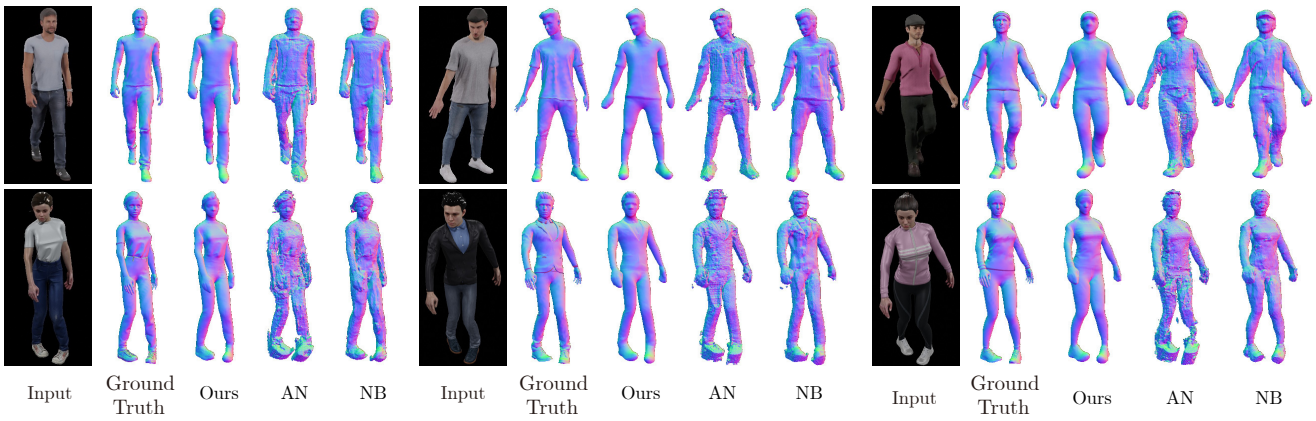

Figure 4. **3D reconstruction on the SyntheticHuman dataset.** The results in the first row are reconstructions from 4-view videos, and the results in second row are reconstructions from monocular videos. Our method significantly outperforms other methods.

	P2S↓				CD↓			
	D-NeRF [58]	NB [57]	AN [56]	Ours	D-NeRF [58]	NB [57]	AN [56]	Ours
S1	3.49	1.44	2.73	**0.64**	2.40	1.39	2.02	**0.81**
S2	3.38	1.68	3.12	**0.69**	2.45	1.48	2.11	**0.74**
S3	3.96	1.52	2.41	**0.58**	2.71	1.42	1.76	**0.74**
S4	4.18	1.20	3.29	**0.58**	2.85	1.23	2.28	**0.71**
S5	1.22	1.20	2.01	**0.45**	1.10	1.14	1.60	**0.49**
S6	1.76	1.31	2.44	**0.58**	1.43	1.28	1.83	**0.60**
S7	1.66	1.61	2.36	**1.21**	1.82	1.74	2.20	**1.47**
average	2.81	1.42	2.62	**0.67**	2.11	1.38	1.97	**0.79**

Table 1. **Results of 3D reconstruction on SyntheticHuman dataset.** The first four rows show the results on monocular videos, and the remaining rows present the results on 4-view videos.

4.2. Comparison with the state-of-the-art methods

We compare with other state-of-the-art methods that train a separate network for each human performer. We evaluate our method with three ways to render images:

- **Ours-V**: As described in Section 3.1, we sample points along camera rays, and transform them to the canonical space using the LBS model and displacement field.
- **Ours-S**: It means the surface-guided rendering, which is illustrated in Figure 3.
- **Ours-S***: As described in Section 3.4, we replace color fields with feature fields and use the surface-guided rendering to render feature maps, which are interpreted into images with a 2D neural renderer.

Performance on the SyntheticHuman dataset. Table 1 compares our method with [56, 57] in terms of the P2S and CD metrics. [56, 57] model the human geometry with the volume density field, while our method adopts the signed distance field. We empirically set the threshold of volume density to extract the geometry of [56, 57]. Our method significantly outperforms them by a margin of at least 0.75 in terms of P2S metric and 0.59 in terms of CD metric. Figure 4 presents the qualitative comparison.

Performance on the Human3.6M dataset. In Table 2, we compare our method with [56–58, 80] on image synthesis.

	Training poses			Novel poses			Rendering
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	speed↑
NHR [80]	20.93	0.866	0.092	20.47	0.857	0.085	4.33
NHR* [80]	22.36	0.890	0.083	21.81	0.879	0.082	2.64
D-NeRF [58]	20.13	0.807	0.202	-	-	-	0.17
NB [57]	23.31	0.903	0.120	22.74	0.885	0.114	0.50
AN [56]	23.00	0.890	0.128	22.55	0.880	0.121	0.25
Ours-V	**24.40**	0.905	0.134	-	-	-	0.11
Ours-S	23.90	0.903	0.128	23.11	**0.894**	0.096	4.78
Ours-S*	23.92	**0.908**	**0.067**	**23.12**	**0.894**	**0.071**	3.95

Table 2. **Results of novel view synthesis of training poses and novel poses on Human3.6M dataset.** The methods are trained on 3-view videos and tested on one novel view. "Ours-V" does not have rendering results on novel poses, because the displacement fields cannot generalize to unseen human poses. "NHR*" uses our reconstructed geometries as input point clouds. The unit for rendering speed is in frames per second (fps). We report the speed of rendering an 1000×1000 image on an RTX 2080Ti GPU.

The results of "NHR" and "AN" are obtained from [56]. Here [80] renders SMPL vertices into images with a 2D CNN. Our method achieves the best performance among all methods, and the surface-guided rendering enables us to have a much faster rendering speed than NeRF-based methods [56–58]. "Ours-S*" significantly outperforms other rendering methods in terms of the LPIPS metric. As shown in [87], LPIPS metric better measures the image quality than PSNR and SSIM. Figures 5 and 6 present the qualitative results of image synthesis and 3D reconstruction.

"Ours-S*" has a similar rendering pipeline with NHR [80]. They both render 3D representations into feature maps and use image-space CNNs to synthesize images. To further validate the effectiveness of neural feature field, we use our reconstructed geometries as the input of NHR, denoted as "NHR*". This makes NHR and "Ours-S*" render images based on the same human geometries. The better input geometries improve the performance of NHR on image synthesis, as shown in Table 2. "Ours-S*" still outperforms "NHR*", indicating that neural feature fields improve the performance on image synthesis.

Figure 5. **Novel view synthesis of novel human poses on Human3.6M and MonoCap.** Our method produces photorealistic rendering results, and the synthesized images of "Ours-S*" achieve the best visual quality. More results can be found in the supplementary material.

	Training poses			Novel poses		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NHR [80]	21.29	0.875	0.110	20.45	0.866	0.123
NB [57]	**21.76**	0.872	0.119	20.83	0.854	0.133
AN [56]	21.06	0.859	0.143	19.58	0.830	0.166
Ours-V	**21.76**	**0.879**	0.117	-	-	-
Ours-S	21.13	0.872	0.097	20.51	0.867	0.107
Ours-S*	21.60	0.876	**0.078**	**20.73**	**0.869**	**0.097**

Table 3. **Results of novel view synthesis of training poses and novel poses on MonoCap dataset.** The methods are trained on monocular videos and tested on ten novel views.

Performance on the MonoCap dataset. Table 3 summarizes the quantitative comparison between our method with [56, 57, 80] on image synthesis. Our method gives the best performance, and "Ours-S*" significantly outperforms other methods in terms of the LPIPS metric. Figures 5 and 6 present the qualitative results of image synthesis and 3D reconstruction, which show that our model is able to reconstruct high-quality human models from monocular videos.

4.3. Ablation studies

We conduct ablation studies to analyze the impact of displacement field, canonical-space viewing direction, and neural feature field on one subject (S1) of the SyntheticHuman dataset and one subject (S9) of the Human3.6M dataset. Table 4 summarizes the results of ablation studies. The "Baseline" means that we represent the human geometry and appearance with implicit fields of signed distance and color, and use the LBS model to produce the deformation fields. Its color network takes the observation-space viewing direction as input. The results show that our adopted components increase the performance of 3D reconstruction and novel pose synthesis. We also present the qualitative results of non-rigid deformations captured by the displacement field in the supplementary material.

	SyntheticHuman S1		Human3.6M S9		
	P2S↓	CD↓	PSNR↑	SSIM↑	LPIPS↓
Baseline	0.96	1.01	23.08	0.884	0.088
+ Displacement field	0.68	0.85	23.16	0.889	0.079
+ Canonical-space viewdir	**0.64**	**0.81**	23.33	0.890	0.077
+ Feature field	-	-	**23.37**	**0.891**	**0.059**

Table 4. **Ablation studies on "S1" of SyntheticHuman and "S9" of Human3.6M.** The results show that the adopted displacement field and canonical-space viewing direction improve the performance of our model on both 3D reconstruction and novel pose synthesis. Rendering with neural feature field significantly improves the rendering quality in terms of the LPIPS metric.

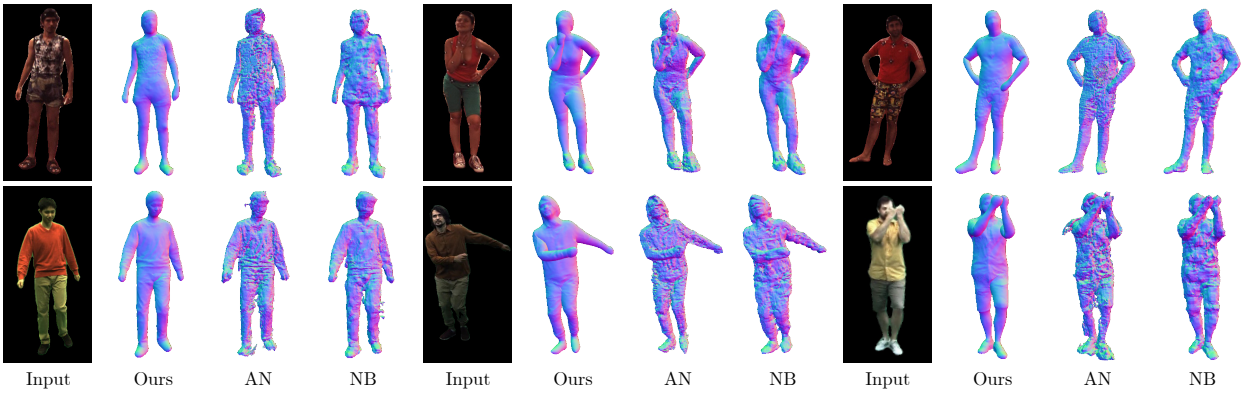

Figure 6. **3D reconstruction on the Human3.6M and MonoCap datasets.** The results in the first row are reconstructions from 3-view videos, and the results in second row are reconstructions from monocular videos. Our method produces better reconstruction results.

	1 point	3 points	5 points	7 points	9 points
PSNR	22.02	**23.48**	23.33	23.21	23.14
SSIM	0.866	0.888	**0.890**	0.889	0.888
LPIPS	0.085	0.081	0.077	0.075	**0.073**

Table 5. **Results of surface-guided rendering with different number of sampled points** on novel pose synthesis. Experiments are performed on "S9" of the Human3.6M dataset.

NeuS [78] proposes another way to perform volume rendering of SDF that ensures unbiased surface reconstruction based on the first-order approximation of SDF. To analyze the influence of the volume rendering scheme, we render AniSDF with the volume rendering technique proposed in NeuS [78], and perform experiments on all subjects of the SyntheticHuman dataset, which gives 0.91 P2S and 1.04 CD on average. AniSDF with the volume rendering technique in Section 3.2 has a better performance, which gives 0.67 P2S and 0.79 CD on average. The reason may be that deformed signed distance fields violate the first-order approximation of SDF in [78]. The per-subject quantitative comparison on the SyntheticHuman dataset is provided in the supplementary material.

Table 5 explores the impact of the number of sampled points in the surface-guided rendering scheme. The results show that sampling only one point degrades the quality of synthesized images. Increasing the number of sampled points improves the performance of our model on image synthesis in terms of the LPIPS metric.

4.4. Running time

We evaluate the speed of the proposed rendering strategies "Ours-S" and "Ours-S*" that render the performer "S9" of Human3.6M dataset on a desktop with an Intel i7 3.7GHz CPU and an RTX 2080 Ti GPU. 1) For 512×512 images, the proposed rendering strategy "Ours-S" runs at 12.2 fps. Specifically, sampling 3D points and transforming them to the canonical space takes 46.5 ms, predicting the signed distance and color fields takes 34.1 ms, and the volume rendering takes 1.4 ms. 2) The proposed rendering strategy "Ours-S*" runs at 10.3 fps for rendering a 512×512 image. Our implementation takes 16.5 ms for predicting the signed distance and feature fields, and 32.7 ms for the forward propagation of the 2D neural renderer.

5. Limitations

Our method has the following limitations. 1) The neural displacement fields cannot generalize to novel human poses, so that our model cannot express dynamic cloth deformations during animation. It would be interesting to learn a network that maps human poses to cloth deformations to produce high-fidelity animations. 2) It is difficult for our method to reconstruct human performers wearing loose clothes, i.e., long dresses, due to the complex deformations. Capturing highly non-rigid deformations from sparse views is left as future work. 3) The proposed model trains a separate network for each human performer, which costs a lot of time. A generalizable model across human subjects could be achieved by equipping our model with image encoders [31, 79, 86].

6. Conclusion

We introduced a novel human representation, named AniSDF, for reconstructing humans from sparse multi-view videos. AniSDF represents the human model with signed distance, color, and feature fields in the canonical space. By establishing observation-to-canonical deformation fields, we learned AniSDF over the video with volume rendering. We developed a surface-guided rendering strategy to improve the rendering speed and leverage the image-space rendering to synthesize high-quality images during animation. Experiments demonstrated that our approach exhibits state-of-the-art performance on the SyntheticHuman, Human3.6M and MonoCap datasets in terms of 3D reconstruction and image synthesis.

Societal impact. The positive side is that our approach makes a step toward immersive telepresence, which helps the communication of people that are physically apart. The negative side is that due to the high-quality reconstruction and animation of human characters, our approach may be used for generating personal videos without consent. We strongly oppose such an application of our method because it violates privacy and security.

References

[1] Blender. https://www.blender.org/. 13

[2] Mixamo. https://www.mixamo.com/. 5, 13

[3] Renderpeople. https://renderpeople.com/. 5, 13

[4] Badour AlBahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. In *SIGGRAPH Asia*, 2021. 2

[5] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *CVPR*, 2018. 14

[6] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *ECCV*, 2020. 2

[7] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *NeurIPS*, 2020. 4

[8] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, 2019. 2

[9] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *ICCV*, 2021. 2

[10] 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 TOG*, 2015. 1, 2

[11] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *SIGGRAPH*, 2000. 1, 2

[12] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 13

[13] Junting Dong, Qing Shuai, Yuanqing Zhang, Xian Liu, Xiaowei Zhou, and Hujun Bao. Motion capture from internet videos. In *ECCV*, 2020. 2

[14] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM TOG*, 2016. 1, 2

[15] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In *CVPR*, 2021. 2

[16] Graham Fyffe. Cosine lobe based relighting from gradient illumination photographs. In *SIGGRAPH*, 2009. 1

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020. 4

[18] 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 TOG*, 2019. 1, 2

[19] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. In *SIGGRAPH Asia*, 2021. 5, 13, 14

[20] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *CVPR*, 2020. 5, 13, 14

[21] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *ICCV*, 2021. 2

[22] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 1996. 2

[23] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, 2020. 2, 4

[24] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2013. 5, 13, 14

[25] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, 2020. 2

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5

[27] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 2

[28] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O'Sullivan. Skinning with dual quaternions. In *I3D*, 2007. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4, 12

[30] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 2

[31] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. In *NeurIPS*, 2021. 8

[32] YoungJoong Kwon, Stefano Petrangeli, Dahun Kim, Haoliang Wang, Eunbyung Park, Viswanathan Swaminathan, and Henry Fuchs. Rotationally-temporally consistent novel view synthesis of human performance video. In *ECCV*, 2020. 2

[33] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH*, 2000. 2, 3

[34] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. Crowdsampling the plenoptic function. In *ECCV*, 2020. 2

[35] Yariv Lior, Kasten Yoni, Moran Dror, Galun Meirav, Atzmon Matan, Basri Ronen, and Lipman Yaron. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020. 2, 3, 4

[36] 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. In *SIGGRAPH Asia*, 2021. 1, 2, 4

[37] Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhöfer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt. Neural human video rendering by learning dynamic textures and rendering-to-video translation. *TVCG*, 2020. 2

[38] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *ACM TOG*, 2019. 2

[39] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, 2020. 2

[40] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. In *SIGGRAPH*, 2019. 2

[41] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM TOG*, 2015. 2, 4

[42] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, 1987. 5, 12

[43] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *CVPR*, 2020. 2

[44] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. Leap: Learning articulated occupancy of people. In *CVPR*, 2021. 2

[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. In *ECCV*, 2020. 1, 2, 4, 5, 12

[46] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *CVPR*, 2019. 2

[47] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015. 2

[48] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[49] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 2

[50] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021. 2

[51] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021. 2

[52] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3

[53] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021. 3

[54] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019. 4

[55] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, 2018. 2

[56] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021. 1, 2, 3, 4, 5, 6, 7, 13, 14

[57] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 1, 2, 3, 6, 7, 13, 14

[58] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 1, 2, 3, 6

[59] Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. Anr: Articulated neural rendering for virtual avatars. In *CVPR*, 2021. 2

[60] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

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

[62] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 2, 5

[63] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, 2020. 2

[64] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. Scanimate: Weakly supervised learning of skinned clothed avatar networks. In *CVPR*, 2021. 2

[65] Soubhik Sanyal, Alex Vorobiov, Timo Bolkart, Matthew Loper, Betty Mohler, Larry S Davis, Javier Romero, and

Michael J Black. Learning realistic human reposing using cyclic self-supervision with 3d shape, pose, and appearance consistency. In *ICCV*, 2021. 2

[66] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. In *ECCV*, 2020. 2

[67] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[68] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1

[69] Dario Seyb, Alec Jacobson, Derek Nowrouzezahrai, and Wojciech Jarosz. Non-linear sphere tracing for rendering deformed signed distance fields. *SIGGRAPH Asia*, 2019. 2

[70] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *CVPR*, 2019. 2

[71] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[72] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

[73] Carsten Stoll, Juergen Gall, Edilson De Aguiar, Sebastian Thrun, and Christian Theobalt. Video-based reconstruction of animatable human characters. *ACM TOG*, 2010. 2

[74] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. In *NeurIPS*, 2021. 2

[75] Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, and Lu Fang. Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. In *ECCV*, 2020. 2

[76] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM TOG*, 2019. 2, 5

[77] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-gif: Neural generalized implicit functions for animating people in clothing. In *ICCV*, 2021. 2

[78] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 2, 4, 8, 13

[79] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 8

[80] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *CVPR*, 2020. 2, 5, 6, 7, 14

[81] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. In *NeurIPS*, 2021. 2

[82] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *CVPR*, 2020. 4

[83] Ze Yang, Shenlong Wang, Sivabalan Manivasagam, Zeng Huang, Wei-Chiu Ma, Xinchen Yan, Ersin Yumer, and Raquel Urtasun. S3: Neural shape, skeleton, and skinning fields for 3d human modeling. In *CVPR*, 2021. 2

[84] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv preprint arXiv:2106.12052*, 2021. 2, 4, 13

[85] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *CVPR*, 2021. 2

[86] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 8

[87] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 6

[88] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *ICCV*, 2019. 2

Supplementary Material

In the supplementary material, we describe how to animate the learned human model. For reproducibility, we provide implementation details, dataset details, and evaluation details. To show the effectiveness of our approach, we present more results of 3D reconstruction and image synthesis. In addition, we provide a video to describe our approach and present the qualitative results.

1. Animation

3D reconstruction. After training, AniSDF can be used to generate 3D human shapes under given human poses. For training human poses, we first construct a set of grid points by discretizing the 3D human bounding box in the observation space with a voxel size of $5mm \times 5mm \times 5mm$. Then, the grid points are transformed to the canonical space using the inverse LBS model and the displacement field, which are fed into the geometry model F_s to compute signed distances. We extract the human mesh from the signed distances with the Marching Cubes algorithm [42].

For novel human poses, we adopt another way to generate 3D human shapes, as the displacement field cannot generalize to unseen human poses. We first discretize the human bounding box in the canonical space with a voxel size of $5mm \times 5mm \times 5mm$ and evaluate the signed distances for the grid points. Then, the canonical human mesh is extracted from the signed distances based on the Marching Cubes algorithm. Blend weights of mesh vertices are obtained by retrieving blend weights of the closest surface points on the SMPL mesh under the canonical space. Given a human pose, we use the forward LBS model to deform the canonical mesh to the observation space.

Image synthesis. For training human poses, we can use "Ours-V", "Ours-S" and "Ours-S*" to render images. The color and feature fields infer the colors and feature vectors based on appearance codes of corresponding video frames. For novel human poses, we can use "Ours-S" and "Ours-S*" to render images. The color and feature fields take the appearance code at the first frame as input.

2. Implementation details

Figures 7, 8, 9, 10 and 11 illustrate network architectures of signed distance field F_s, color field $F_{\mathbf{c}}$, displacement field $F_{\Delta\mathbf{x}}$, feature field $F_{\mathbf{f}}$, and 2D neural renderer, respectively. We perform positional encoding [45] to the spatial point and viewing direction. 6 frequencies are used when encoding spatial position, and 4 frequencies are used when encoding viewing direction. The dimensions of appearance code $\boldsymbol{\ell}_i$ and displacement field code $\boldsymbol{\psi}_i$ are 128.

Training. We take a two-stage training pipeline. First, the parameters of F_s, $F_{\mathbf{c}}$, $F_{\Delta\mathbf{x}}$, $\{\boldsymbol{\ell}_i\}$ and $\{\boldsymbol{\psi}_i\}$ are jointly optimized over the input video. Second, we fix the parameters of F_s and $\{\boldsymbol{\ell}_i\}$, and train the feature field $F_{\mathbf{f}}$ and 2D neural renderer. We use the Adam optimizer [29] for the training and set the learning rate as $5e^{-4}$, which decays exponentially to $5e^{-5}$ during the optimization. The training is conducted on one 2080 Ti GPU. We sample 1024 rays at each iteration. For a monocular video of 300 frames, both stages take around 100k iterations to converge.

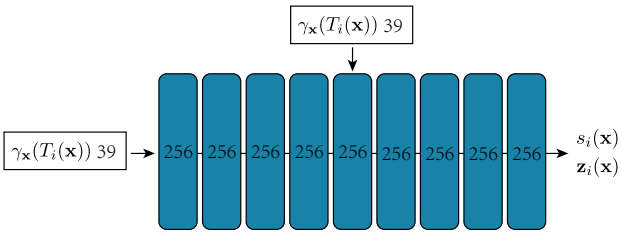

Figure 7. **Signed distance field.** All layers are linear layers with softplus activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$ as input and output the signed distance $s_i(\mathbf{x})$ and geometry feature $\mathbf{z}_i(\mathbf{x})$. The dimension of the input is shown in each block.

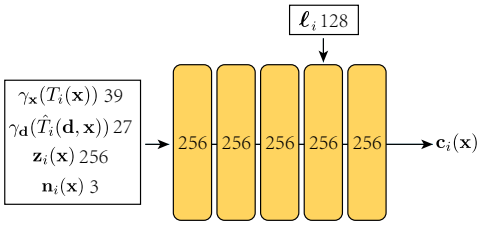

Figure 8. **Color field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$, the positional encoding of view direction $\gamma_{\mathbf{d}}(\hat{T}_i(\mathbf{d},\mathbf{x}))$, normal $\mathbf{n}_i(\mathbf{x})$, and geometry feature $\mathbf{z}_i(\mathbf{x})$ as inputs. We introduce the appearance code $\boldsymbol{\ell}_i$ in the fourth layer. The dimension of the input is shown in each block.

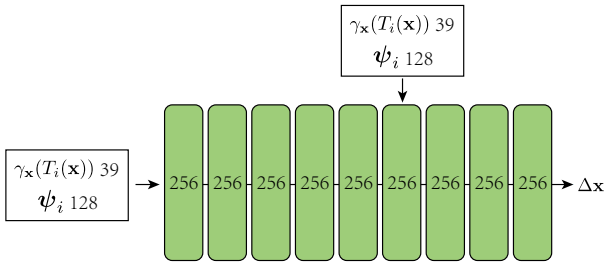

Figure 9. **Displacement field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$ and the per-frame latent code for ψ_i as input.

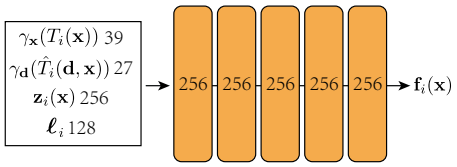

Figure 10. **Feature field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$, the positional encoding of view direction $\gamma_{\mathbf{d}}(\hat{T}_i(\mathbf{d},\mathbf{x}))$, and geometry feature $\mathbf{z}_i(\mathbf{x})$ as inputs, and output the feature vector $\mathbf{f}_i(\mathbf{x})$.

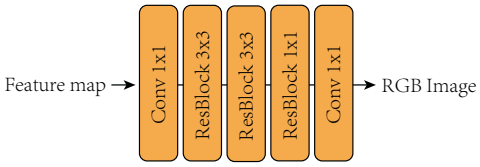

Figure 11. **2D neural renderer.** The network consists of standard ConvBlocks and ResBlocks. ALL blocks adopt the ReLU activations. The kernel size is shown in each block. The dimension of feature in the convolution operator is 256.

subject	S1	S5	S6	S7	S8	S9	S11
training	150	250	150	300	250	260	200
test	49	127	83	200	87	133	82

Table 6. **The number of training frames and test frames of the Human3.6M dataset.**

subject	S1	S2	S3	S4	S5	S6	S7
training	69	300	70	100	100	100	70

Table 7. **The number of video frames for each subject in the SyntheticHuman dataset.**

3. Dataset details

Human3.6M [24] Following [56], we use three camera views for training and test on the remaining view. [56] select video clips from the action "Posing" of S1, S5, S6, S7, S8, S9, and S11. The number of training frames and test frames is described in Table 6.

SyntheticHuman It contains 7 animated human characters obtained from RenderPeople [3] and Mixamo [2]. We render these subjects using Blender [1, 12]. Subjects S1, S2, S3, and S4 perform rotation with A-pose, which are rendered into monocular videos. Subjects S5, S6 and S7 perform random actions, which are rendered into 4-view videos. The number of video frames is listed in Table 7.

MonoCap It consists of two videos "Lan" and "Marc" from DeepCap dataset [20], and two videos "Olek" and "Vlad" from DynaCap dataset [19]. "Lan" is selected from 620-th frame to 1220-th frame in the original video. "Marc" is selected from 35000-th frame to 35600-th frame. "Olek" is selected from 12300-th frame to 12900-th frame. "Vlad" is selected from 15275-th frame to 15875-th frame. Each clip has 300 frames for training and 300 frames for evaluating novel pose synthesis, respectively. We use the 0-th camera as the training view for "Lan" and "Marc". The 44-th camera is selected as the training view for "Olek". The training view of "Vlad" is the 66-th camera. We uniformly select ten cameras from the remaining cameras for test.

	P2S↓				CD↓			
	NB [57]	AN [56]	Ours + Neus	Ours	NB [57]	AN [56]	Ours + Neus	Ours
S1	1.44	2.73	0.91	**0.64**	1.39	2.02	1.09	**0.81**
S2	1.68	3.12	0.87	**0.69**	1.48	2.11	0.93	**0.74**
S3	1.52	2.41	0.80	**0.58**	1.42	1.76	0.96	**0.74**
S4	1.20	3.29	0.84	**0.58**	1.23	2.28	0.99	**0.71**
S5	1.20	2.01	0.65	**0.45**	1.14	1.60	0.68	**0.49**
S6	1.31	2.44	0.89	**0.58**	1.28	1.83	0.90	**0.60**
S7	1.61	2.36	1.42	**1.21**	1.74	2.20	1.71	**1.47**
average	1.42	2.62	0.91	**0.67**	1.38	1.97	1.04	**0.79**

Table 8. **Results of 3D reconstruction on SyntheticHuman dataset.** "Ours + Neus" means that we render AniSDF with the volume rendering scheme in Neus [78].

4. Evaluation details

We follow [57] to calculate the metrics of image synthesis. Specifically, the 3D human bounding box is first projected to produce a 2D mask. Then, we calculate the PSNR metric based on the pixels inside the 2D mask. Since the SSIM and LPIPS metrics require the image input, we compute the 2D box that bounds the 2D mask and crop the image within the box, which is used to calculate the SSIM and LPIPS metrics. For the SyntheticHuman dataset, we calculate the reconstruction metrics every 10-th frame. For the Human3.6M and MonoCap datasets, we calculate the metrics of image synthesis every 30-th frame.

5. Results of 3D reconstruction

Table 8 lists the per-subject comparison on the SyntheticHuman dataset, which shows that AniSDF with the volume rendering technique in VolSDF [84] achieves the best performance of 3D reconstruction in terms of the P2S and CD metrics.

Figure 12 visualizes the non-rigid deformations captured by the displacement fields. To obtain the geometry deformed by the displacement field, we first construct a set of grid points by discretizing the human bounding box in the canonical space and transform the grid points using the displacement field. Then, we evaluate the signed distances for the transformed points and extract the human mesh with the Marching Cubes algorithm.

Figure 13 presents the qualitative results of our reconstructed geometries. We additionally reconstruct humans in

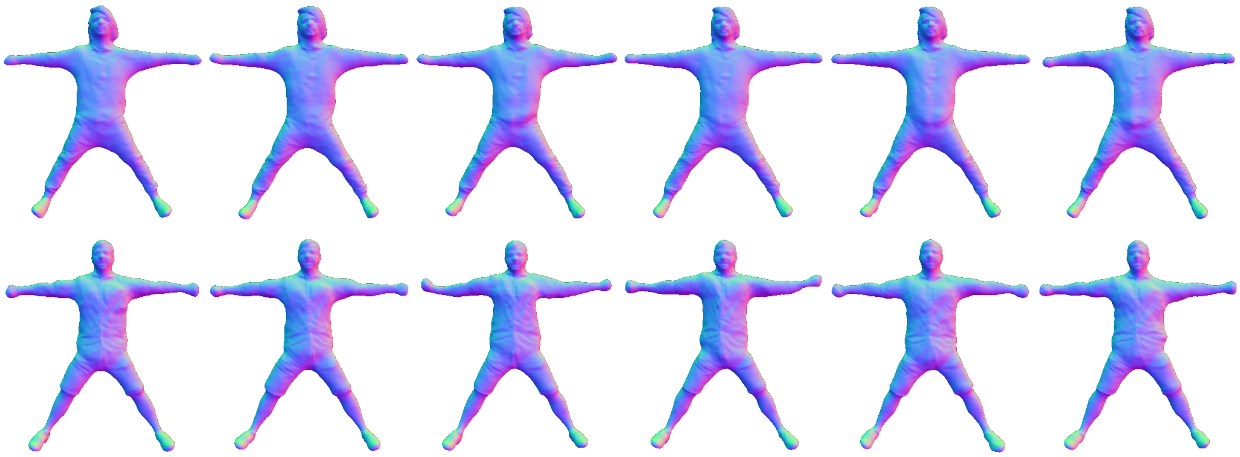

Figure 12. **Canonical geometries deformed by the displacement field on the MonoCap dataset.** We deform the canonical geometries using the learned displacement fields in different video frames. For different frames, the displacement fields produce different non-rigid deformations, including the cloth deformations and body deformations. We provide more results in the supplementary video.

Figure 13. **3D reconstruction results in the Human3.6M, MonoCap, and People-Snapshot datasets.** We reconstruct humans of Human3.6M [24] from 3-view videos. The geometries in MonoCap [19, 20] and People-Snapshot [5] are reconstructed from monocular videos.

the People-Snapshot dataset [5], which captures performers rotating while holing the A-pose. The results demonstrate that our method can reconstruct high-quality geometries from monocular videos.

6. Results of image synthesis

Figure 14 presents the rendering results of subjects "Olek" and "Vlad" in the MonoCap dataset driven by complex human poses, which show that our method can synthesize photorealistic images under complex human poses. We provide more results in the supplementary video.

Figures 15 and 16 show the qualitative comparisons between our method and [56,57,80] on novel view synthesis of training human poses and novel human poses, respectively. Our method recovers detailed human appearance and produces more photorealistic images than other methods.

Figure 14. **Results of image synthesis under complex human poses in the MonoCap dataset.** Our model is trained on short video clips and can generalize to complex human poses. More results can be found in the supplementary video.

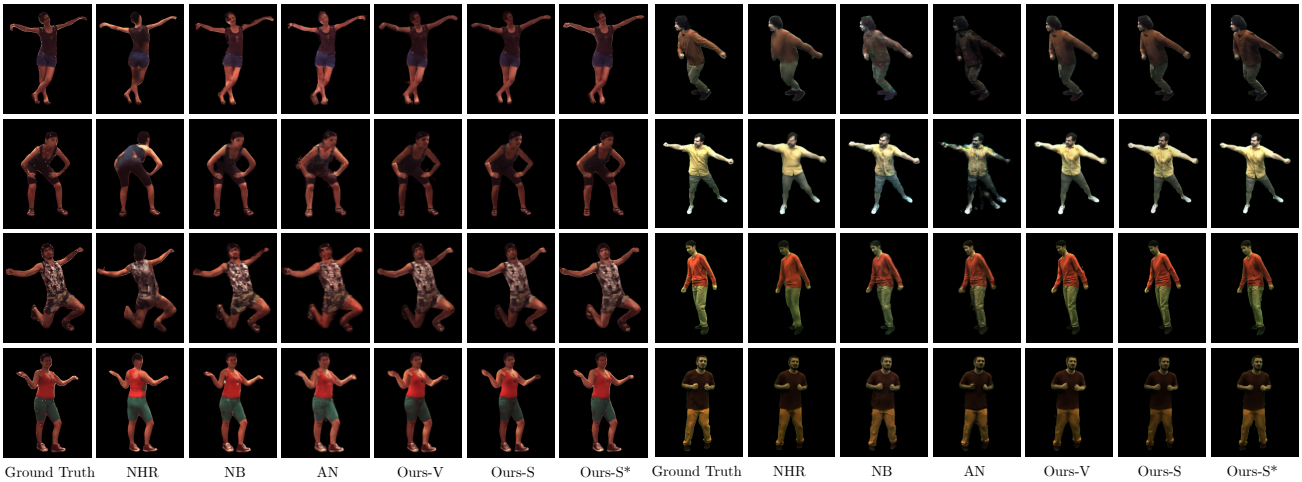

Figure 15. **Novel view synthesis of training human poses in the Human3.6M and MonoCap datasets.** Our method has better performance on image synthesis. "Ours-S*" renders more appearance details. Zoom in for details.

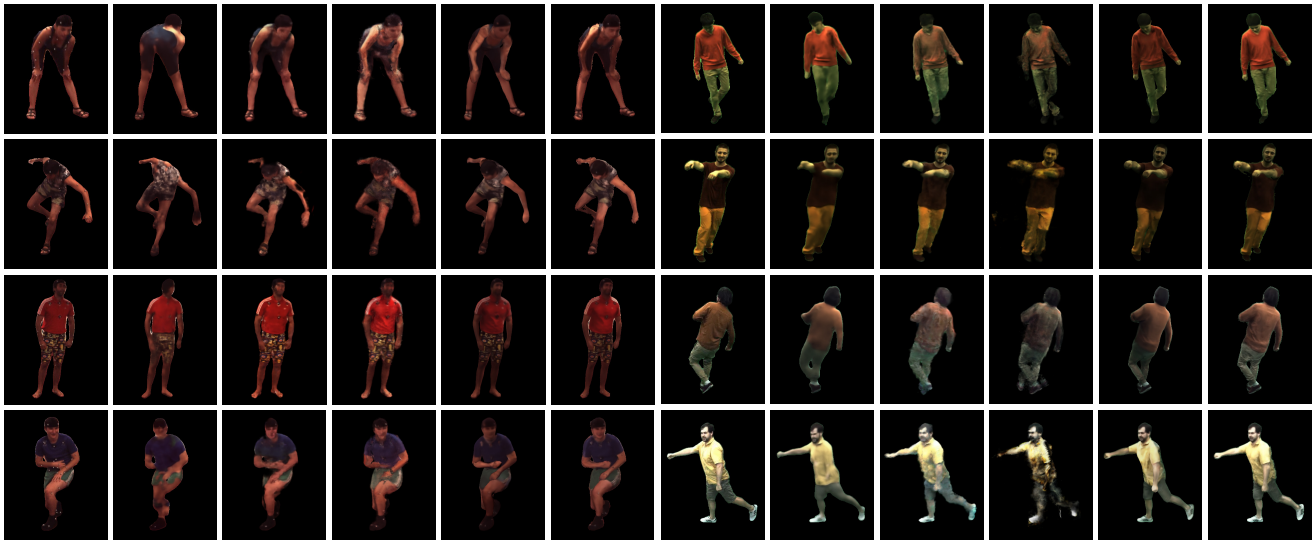

Figure 16. **Novel view synthesis of novel human poses in the Human3.6M and MonoCap datasets.** The rendered images of our method have a higher visual quality. Zoom in for details.