UV Volumes for Real-time Rendering of Editable Free-view Human Performance

Yue Chen[1,2*] Xuan Wang[3,4*] Xingyu Chen[1,2] Qi Zhang[4]
Xiaoyu Li[4] Yu Guo[1,2†] Jue Wang[4] Fei Wang[1,2]
[1] National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
[2] IAIR, Xi'an Jiaotong University [3]Ant Group [4]Tencent AI Lab

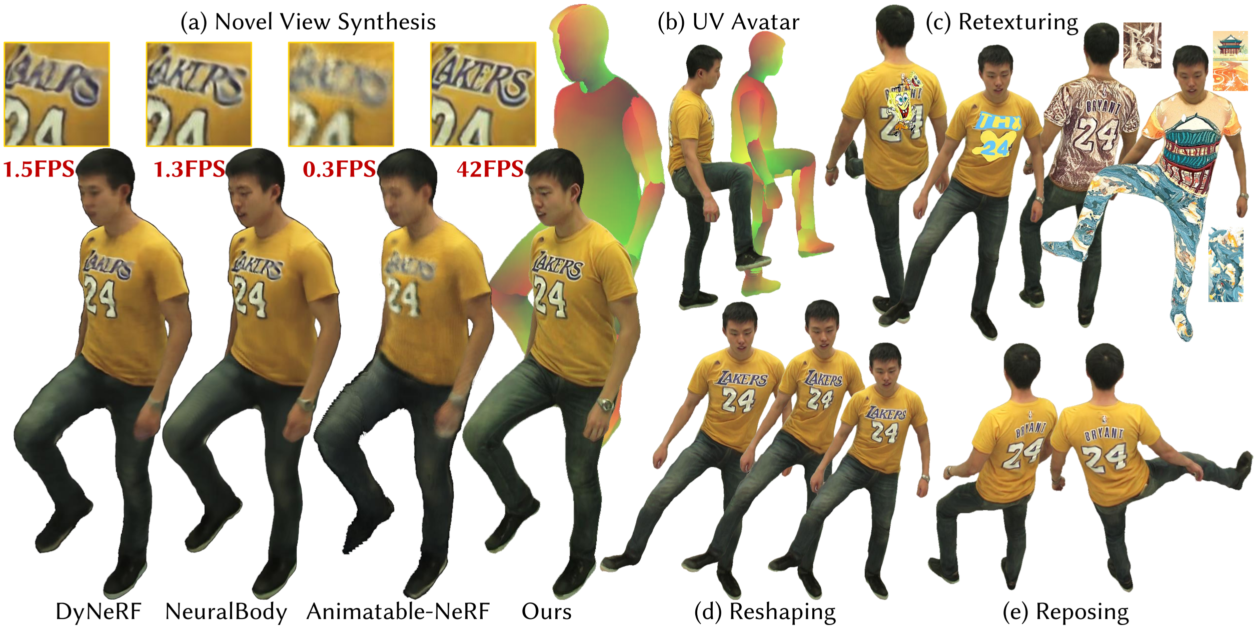


Figure 1. We decompose a dynamic human into 3D UV Volumes along with a 2D texture. The disentanglement of appearance and geometry enables us to achieve (a) high-fidelity real-time novel view synthesis guided by (b) a smooth UV avatar, (c) retexturing of a 3D human by editing a 2D texture, (d) reshaping and (e) reposing by changing the parameters of a human model while keeping the texture untouched.

Abstract

Neural volume rendering enables photo-realistic renderings of a human performer in free-view, a critical task in immersive VR/AR applications. But the practice is severely limited by high computational costs in the rendering process. To solve this problem, we propose the UV Volumes, *a new approach that can render an editable free-view video of a human performer in real-time. It separates the high-frequency (i.e., non-smooth) human appearance from the 3D volume, and encodes them into 2D neural texture stacks (NTS). The smooth UV volumes allow much smaller and shallower neural networks to obtain densities and texture coordinates in 3D while capturing detailed appearance in 2D NTS. For editability, the mapping between the parameterized human model and the smooth texture coordinates allows us a better generalization on novel poses and shapes. Furthermore, the use of NTS enables interesting applications, e.g., retexturing. Extensive experiments on CMU Panoptic, ZJU Mocap, and H36M datasets show that our model can render 960 × 540 images in 30FPS on average with comparable photo-realism to state-of-the-art methods. The project and supplementary materials are available at https://fanegg.github.io/UV-Volumes.*

1. Introduction

Synthesizing a free-view video of a human performer in motion is a long-standing problem in computer vision.

[*] Authors contributed equally to this work.

[†] Corresponding Author. This work is partly supported by the National Key Research and Development Program of China under Grant 2022YFB3303800 and National Key Projects of China, 2021XJTU0040.

Early approaches [4] rely on obtaining an accurate 3D mesh sequence through multi-view stereo. However, the computed 3D mesh often fails to depict the complex geometry structure, resulting in limited photorealism. In recent years, methods (e.g., NeRF [33]) that make use of volumetric representation and differentiable ray casting have shown promising results for novel view synthesis. These techniques have been further extended to tackle dynamic scenes.

Nonetheless, NeRF and its variants require a large number of queries against a deep Multi-Layer Perceptron (MLP). Such time-consuming computation prevents them from being applied to applications that require high rendering efficiency. In the case of static NeRF, a few methods [10, 42, 58] have already achieved real-time performance. However, for dynamic NeRF, solutions for real-time rendering of volumetric free-view video are still lacking.

In this work, we present *UV Volumes*, a novel framework that can produce an editable free-view video of a human performer in motion and render it in real-time. Specifically, we take advantage of a pre-defined UV-unwrapping (e.g., SMPL or dense pose) of the human body to tackle the geometry (with texture coordinates) and textures in two branches. We employ a sparse 3D Convolutional Neural Networks (CNN) to transform the voxelized and structured latent codes anchored with a posed SMPL model to a 3D feature volume, in which only smooth and view-independent densities and UV coordinates are encoded. For rendering efficiency, we use a shallow MLP to decode the density and integrate the feature into the image plane by volume rendering. Each feature in the image plane is then individually converted to the UV coordinates. Accordingly, we utilize the yielded UV coordinates to query the RGB value from a pose-dependent neural texture stack (NTS). This process greatly reduces the number of queries against MLPs and enables real-time rendering.

It is worth noting that the 3D Volumes in the proposed framework only need to approximate relatively “smooth” signals. As shown in Figure 2, the magnitude spectrum of the RGB image and the corresponding UV image indicates that UV is much smoother than RGB. That is, we only model the low-frequency density and UV coordinate in the 3D volumes, and then detail the appearance in the 2D NTS, which is also spatially aligned across different poses. The disentanglement also enhances the generalization ability of such modules and supports various editing operations.

We perform extensive experiments on three widely-used datasets: CMU Panoptic, ZJU Mocap, and H36M datasets. The results show that the proposed approach can effectively generate an editable free-view video from both dense and sparse views. The produced free-view video can be rendered in real-time with comparable photorealism to the state-of-the-art methods that have much higher computational costs. In summary, our major contributions are:

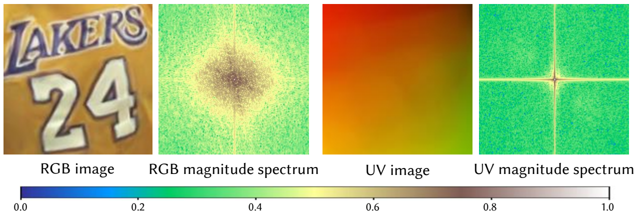


Figure 2. Discrete Fourier Transform (DFT) for RGB and UV image. In the magnitude spectrum, the distance from each point to the midpoint describes the frequency, the direction from each point to the midpoint describes the direction of the plane wave, and the value of the point describes its amplitude. The distribution of the UV magnitude spectrum is more concentrated in the center, which indicates that the frequency of the UV image is lower.

- A novel system for rendering editable human performance video in free-view and real-time.
- UV Volumes, a method that can accelerate the rendering process while preserving high-frequency details.
- Extended editing applications enabled by this framework, such as reposing, retexturing, and reshaping.

2. Related Work

Novel View Synthesis for Static Scenes. Novel view synthesis for static scenes is a well-explored problem. Early image-based rendering approaches [2, 6, 7, 14, 24] utilize densely sampled images to obtain novel views with light fields instead of explicit or accurate geometry estimation. The learning-based methods [8, 16, 23, 32, 47] apply neural networks to reuse input pixels from observed viewpoints. In recent years, dramatic improvements have been achieved by neural volume rendering techniques. For instance, NeRF [33] represents a static scene using a deep MLP, mapping 3D spatial locations and 2D viewing directions to volumetric density and radiance. For computation efficiency, rendering high-resolution scenes via NeRF is time-consuming since it requires millions of queries to obtain the density and radiance. Subsequent works [9,10,17,35,42,58] attempt to accelerate the inference of vanilla NeRF in various ways, some of which achieve real-time rendering performance, but only for static scenes. For editability, the generative models, FENeRF [49] and IDE-3D [48], exploit semantic masks to edit the synthesized free-view portraits, but they are not compatible with the free-view performance capture task. NeuTex [56] also employs the UV-texture to store the appearance and enables editing on the texture map. Unfortunately, it can only tackle static objects.

Free-View Video Synthesis. Early methods [5, 34] rely on accurate 3D reconstruction and texture rendering captured by dome-based multi-camera systems to synthesize

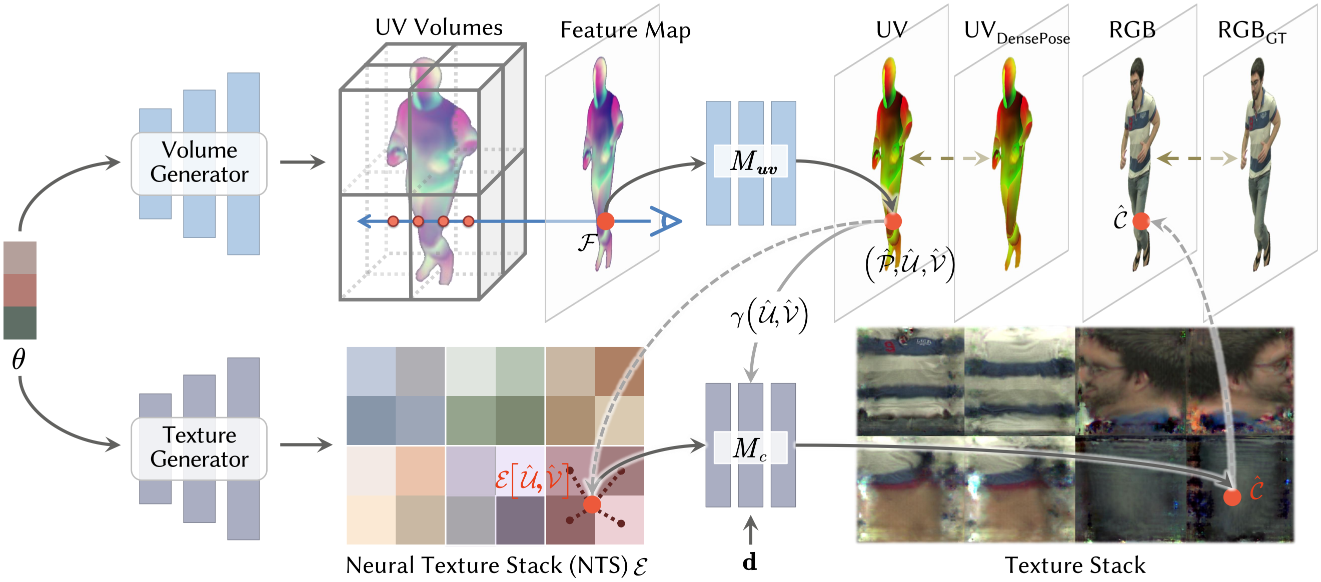

UV Volumes
Feature Map
UV
UV$_{DensePose}$
RGB
RGB$_{GT}$
Volume Generator
Texture Generator
θ
M_{uv}
M_c
$\mathcal{F}$
$(\hat{\mathcal{P}},\hat{\mathcal{U}},\hat{\mathcal{V}})$
$\gamma(\hat{\mathcal{U}},\hat{\mathcal{V}})$
$\hat{\mathcal{C}}$
$\mathcal{E}[\hat{\mathcal{U}},\hat{\mathcal{V}}]$
$\mathbf{d}$
Neural Texture Stack (NTS) $\mathcal{E}$
Texture Stack

Figure 3. Overall pipeline of proposed framework. Our model has two main branches: 1) Based on a human pose θ, a volume generator constructs UV volumes involving the feature of UV information. Then a feature map can be rendered via differentiable raymarching and decoded to texture coordinates (UV) pixel-by-pixel. 2) A texture generator produces a pose-dependent Neural Texture Stack(NTS) $\mathcal{E}$ that encodes the highly-detailed appearance information. The UV coordinates and the texture embedding interpolated from NTS are passed into an MLP to predict the color $\hat{\mathcal{C}}$ at the desired ray direction $\mathbf{d}$.

novel views of a dynamic scene. Recently, various neural representations have been employed in differentiable rendering to depict dynamic scenes, such as voxels [29], point clouds [55], textured meshes [1, 31, 50], and implicit functions [25,27,36,37,39,40]. Particularly, DyNeRF [25] takes the latent code as the condition for time-varying scenes, while NeuralBody [39] employs structured latent codes anchored to a posed human model. Other deformation-based NeRF variants [26, 36, 37, 40, 51] take as input the monocular video, as a result, they fail to synthesize the free-view spatio-temporal visual effects. Besides, they also suffer from the high computational cost in inference and the lack of editing abilities. Geometric constraints and discrete space representation are exploited in methods [44, 59], and a hybrid scene representation is used for efficiency in [30, 43]. The method [53] employs Fourier PlenOctree to accelerate rendering, but the photorealism is harmed by the shared discrete representation across the time sequence. Furthermore, all these models are still non-editable.

Editable Free-View Videos. There exist previous works that focus on the problem of producing editable free-view videos or animatable avatars. ST-NeRF [60] exploits the layered neural representation in order to move, rotate and resize individual objects in free-view videos. Some methods [38, 54, 57] decompose a dynamic human into a canonical neural radiance field and a skeleton-driven warp field that backward maps observation-space points to canonical space. However, learning a backward warp field is highly under-constrained since the backward warp field is pose-dependent [3]. Textured Neural Avatars [45] proposes utilizing the texture map to improve novel pose generalization, whereas employing a 2D rendering neural network prevents it from consistent novel view synthesis. Neural Actor [28] takes the texture map as latent variables. Nevertheless, the requirement of the ground-truth texture map limits their application in many cases. In contrast, our approach estimates the texture map end-to-end, and can produce editable (including reposing, reshaping and retexturing) free-view videos in real-time from both dense and sparse views.

3. Method

Given multi-view videos of a performer, our model generates an editable free-view video that supports real-time rendering. We use the availability of an off-the-shelf SMPL model and the pre-defined UV unwrap in Densepose [15] to introduce proper priors into our framework. In this section, we describe the details of our framework, which is shown in Figure 3. The two main branches in our framework are presented in turn. One is to generate the UV volumes (Sec. 3.1), and the other is the generation of NTS (Sec. 3.2). Then we provide a more detailed description of the training process in Sec. 3.3.

3.1. UV Volumes

Neural radiance fields [33] have been proven to produce free-viewpoint images with view consistency and high fidelity. Nonetheless, capturing the high-fidelity appearance in a dynamic scene is time-consuming and difficult. To this end, we propose the UV volumes in which only the density and texture coordinate (i.e., UV coordinate) are encoded instead of human appearance. Given the UV image rendered by ray casting, we can use the UV coordinates to query the corresponding RGB values from the 2D NTS by employing the UV unwrap defined in Densepose.

We utilize the volume generator to construct UV volumes. First, the time-invariant latent codes anchored to a posed SMPL model are voxelized and taken as the input. Then we use the 3D sparse CNN to encode the voxelized latent codes to a 3D feature volume named UV volumes, which contains UV information.

Given a sample image $\mathcal{I}$ of multi-view videos, we provide a posed SMPL parameterized by human pose θ and a set of latent codes $\mathbf{z}$ anchored on its vertices and then query the feature vector $f(\mathbf{x}, \mathbf{z}, \theta)$ at point $\mathbf{x}$ from the generated UV volumes. The feature vector is fed into a shallow MLP M_σ to predict the volume density:

$$\sigma(\mathbf{x}) = M_\sigma(f(\mathbf{x}, \mathbf{z}, \theta)) \,. \tag{1}$$

We then apply the volume rendering [22] technique to render the UV feature volume into a 2D feature map. We sample N_i points $\{\mathbf{x}_i\}_{i=1}^{N_i}$ along the camera ray $\mathbf{r}$ between near and far bounds based on the posed SMPL model in 3D space. The feature at the pixel can be calculated as:

$$\begin{aligned} \mathcal{F}(\mathbf{r}) &= \sum_{i=1}^{N_i} T_i \left(1 - \exp(-\sigma(\mathbf{x}_i)\,\delta_i)\right) f(\mathbf{x}_i, \mathbf{z}, \theta) \,, \\ &\text{where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma(\mathbf{x}_j)\,\delta_j\right) , \end{aligned} \tag{2}$$

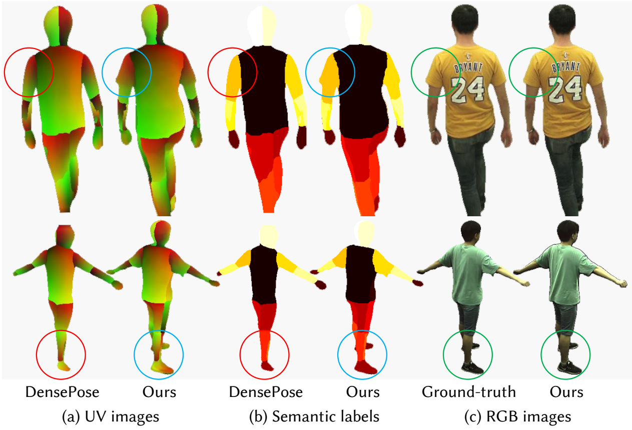
and $\delta_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2$ is the distance between adjacent sampled points. An MLP M_{uv} is then used to individually decode all the pixels in the yielded view-invariant feature map to their corresponding texture coordinates and generate the UV image. In specific, the texture coordinates can be represented as:

$$\left(\hat{\mathcal{P}}(\mathbf{r}), \hat{\mathcal{U}}(\mathbf{r}), \hat{\mathcal{V}}(\mathbf{r})\right) = M_{uv}(\mathcal{F}(\mathbf{r})) \,, \tag{3}$$

where $\hat{\mathcal{P}}$ and $\hat{\mathcal{U}}, \hat{\mathcal{V}}$ are the corresponding part assignments and UV coordinates, respectively.

3.2. Neural Texture Stack

Given the generated UV image, we employ the continuous texture stack encoded in the implicit neural representation to recover the color image. To extract the local relation of the neural texture stack with respect to the human pose, we use a CNN texture generator G to produce the pose-dependent NTS:



(a) UV images (b) Semantic labels (c) RGB images

Figure 4. Given noisy UV and semantic labels (e.g., red circles), we can recover proper UV volumes (e.g., blue circles) under the intrinsic multi-view constraint of minimizing the photometric error between renderings and ground-truth (e.g., green circles).

$$\mathcal{E}_k = G(\theta, \mathbf{k}) \,, \tag{4}$$

where we subdivide the body surface into $N_k = 24$ parts, and $\mathbf{k}$ is a one-hot label vector representing the k-th body part. At a foreground pixel, the part assignments $\hat{\mathcal{P}}$ predicted from UV volumes (referred in Equation (3)) can be interpreted as the probability of the pixel belonging to the k-th body part, which is defined as $\sum_{k=1}^{N_k} \hat{\mathcal{P}}_k(\mathbf{r}) = 1$. For each human body part k, the texture generator generates the corresponding neural texture stack $\mathcal{E}_k$. We forward propagate the generator network G once to predict the neural textures with a batch size of 24. Let $\hat{\mathcal{U}}_k$ and $\hat{\mathcal{V}}_k$ denote the predicted UV coordinates of the k-th body part. We sample the texture embeddings at non-integer locations $(\hat{\mathcal{U}}_k(\mathbf{r}), \hat{\mathcal{V}}_k(\mathbf{r}))$ in a piecewise-differentiable manner using bilinear interpolation [19]:

$$\mathbf{e}_k(\mathbf{r}) = \mathcal{E}_k\left[\hat{\mathcal{U}}_k(\mathbf{r}), \hat{\mathcal{V}}_k(\mathbf{r})\right] . \tag{5}$$

To model the high-frequency color of human performances, we apply positional encoding $\gamma(\cdot)$ [41] to UV coordinates and the viewing direction, and pass the encoded UV map along with the sampled texture embedding into an MLP M_c to decode the view-dependent color $\hat{\mathcal{C}}_k(\mathbf{r})$ of camera ray $\mathbf{r}$ at the desired viewing direction $\mathbf{d}$:

$$\hat{\mathcal{C}}_k(\mathbf{r}) = M_c\left(\gamma(\hat{\mathcal{U}}_k(\mathbf{r}), \hat{\mathcal{V}}_k(\mathbf{r})), \mathbf{e}_k(\mathbf{r}), \mathbf{k}, \gamma(\mathbf{d})\right) . \tag{6}$$

Following that, the color $\hat{\mathcal{C}}(\mathbf{r})$ at each pixel is reconstructed via a weighted combination of decoded colors at N_k body parts, where the weights are prescribed by part assignments $\hat{\mathcal{P}}_k$:

$$\hat{\mathcal{C}}(\mathbf{r}) = \sum_{k=1}^{N_k} \hat{\mathcal{P}}_k(\mathbf{r})\, \hat{\mathcal{C}}_k(\mathbf{r}). \tag{7}$$

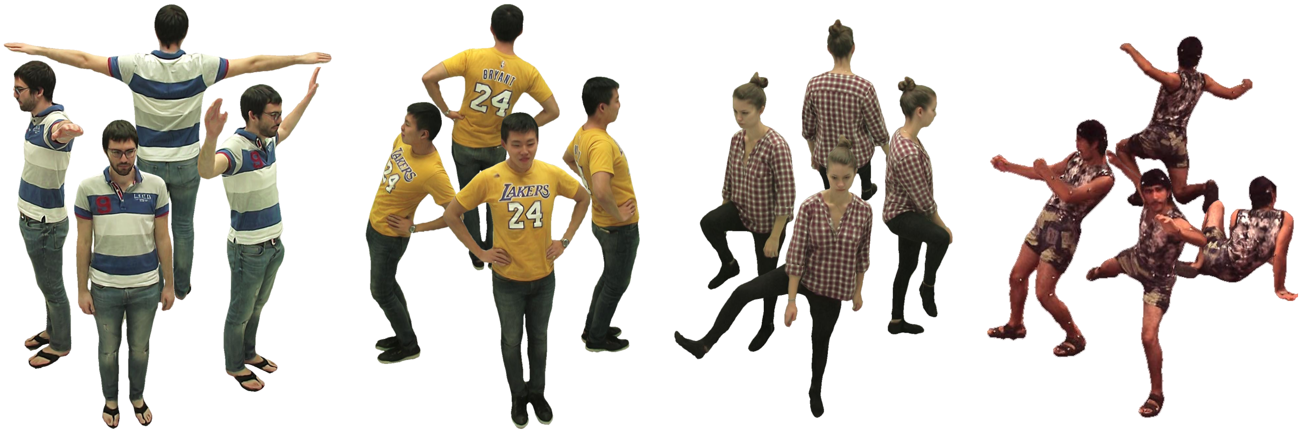


Figure 5. The novel view synthesis of our model on various human performances, which achieves high-fidelity renderings in real-time.

3.3. Training

Collecting the results of all rays $\{\hat{\mathcal{C}}(\mathbf{r})\}^{H\times W}$, we denote the entire rendered image as $\hat{\mathcal{I}} \in \mathbb{R}^{H\times W\times 3}$. To learn the parameters of our model, we optimize the photometric error between the renderings $\hat{\mathcal{I}}$ and the ground-truth images $\mathcal{I}$:

$$\mathcal{L}_{\text{rgb}} = \left\|\hat{\mathcal{I}} - \mathcal{I}\right\|_2^2 . \tag{8}$$

Benefiting from our memory-saving framework that disentangles appearance and geometry, we can render an entire image during training instead of sampling image patches [33, 39]. Thus, we also compare the rendered images against the ground-truth using perceptual loss [11, 20, 52], which extracts feature maps by a pretrained fixed VGG network $\psi(\cdot)$ [46] from both images and minimizes the L1-norm between them:

$$\mathcal{L}_{\text{vgg}} = \left\|\psi(\hat{\mathcal{I}}) - \psi(\mathcal{I})\right\|_1 . \tag{9}$$

To warm-start the UV volumes and regularize its solution space, we leverage the pre-trained DensePose model as an auxiliary supervisor. In particular, we perform the DensePose network on the training data and utilize the outputs of Denspose as pseudo supervision, such that we can regularize UV volumes by semantic loss $\mathcal{L}_{\text{p}}$ and UV-metric loss $\mathcal{L}_{\text{uv}}$ between DensePose outputs and our UV images:

$$\begin{aligned} \mathcal{L}_{\text{p}} &= \sum_{k=1}^{N_k} \mathcal{P}_k \log(\hat{\mathcal{P}}_k), \\ \mathcal{L}_{\text{uv}} &= \sum_{k=1}^{N_k} \mathcal{P}_k \left(\left\|\hat{\mathcal{U}}_k - \mathcal{U}_k\right\|_2^2 + \left\|\hat{\mathcal{V}}_k - \mathcal{V}_k\right\|_2^2 \right), \end{aligned} \tag{10}$$

where N_k is the number of body parts, and $\mathcal{P}_k$ and $\hat{\mathcal{P}}_k$ are respectively the multi-class semantic probability at the k-th part of DensePose outputs and UV images. Similarly, $\mathcal{U}_k$,$\mathcal{V}_k$ and $\hat{\mathcal{U}}_k$, $\hat{\mathcal{V}}_k$ are the predicted UV coordinates at the k-th part of DensePose and UV images, respectively. $\mathcal{L}_{\text{p}}$ is chosen as a multi-class cross-entropy loss to encourage rendered part labels to be consistent with provided DensePose labels, and $\mathcal{L}_{\text{uv}}$ promotes to generate inter-frame consistent UV coordinates.

We present the UV images predicted by our UV volumes and the pseudo supervision of DensePose in Figure 4. Given noisy semantic and UV labels (e.g., the red circles), we can reconstruct proper UV volumes (e.g., the blue circles) under the intrinsic multi-view constraint of RGB loss (e.g., the green circles). As shown in the second row of Figure 4, it can be observed that UV volumes successfully recover the UV images even though the provided DensePose supervision is incorrect.

Given the binary human mask $\mathcal{S}$ for the observed image $\mathcal{I}$, we propose a silhouette loss to facilitate UV volumes modeling a more fine-grained geometry:

$$\begin{aligned} T(\mathbf{r}) &= \exp\left(-\sum_{j=1}^{N_i-1} \sigma(\mathbf{x}_j)\delta_j\right), \\ \mathcal{L}_{\text{s}} &= \sum_{\mathbf{r}\in\mathcal{R}} \left(\mathcal{S}(\mathbf{r})(1 - T(\mathbf{r})) + (1 - \mathcal{S}(\mathbf{r}))T(\mathbf{r})\right), \end{aligned} \tag{11}$$

where $T(\mathbf{r})$ is accumulated transmittance. Here we define the value of mask $\mathcal{S}(\mathbf{r})$ in the foreground as zero, and the background as one.

We combine the aforementioned losses and jointly train our model to optimize the full objective:

$$\mathcal{L} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{vgg}}\mathcal{L}_{\text{vgg}} + \lambda_{\text{p}}\mathcal{L}_{\text{p}} + \lambda_{\text{uv}}\mathcal{L}_{\text{uv}} + \lambda_{\text{s}}\mathcal{L}_{\text{s}} . \tag{12}$$

4. Experiments

To demonstrate the effectiveness and efficiency of our method, we perform extensive experiments. We report quantitative results using four standard metrics: PSNR, SSIM, LPIPS, and FPS[1]. And the qualitative experiments

[1] The sparse CNN output is pre-computed for the reported framerates.

Datasets		View synthesis quality															Efficiency			
		PSNR ↑					SSIM ↑					LPIPS ↓					FPS ↑			
		DN	NB	AN	w/o $\mathcal{L}_p$	Ours	DN	NB	AN	w/o $\mathcal{L}_p$	Ours	DN	NB	AN	w/o $\mathcal{L}_p$	Ours	DN	NB	AN	Ours
CMU (960×540)	p1	30.04	29.78	27.12	30.09	30.38	0.968	0.962	0.936	0.963	0.966	0.088	0.099	0.135	0.055	0.036	1.01	0.76	0.21	44.76
	p2	25.56	25.68	26.13	28.51	28.78	0.939	0.942	0.903	0.952	0.953	0.137	0.139	0.204	0.062	0.044	1.45	1.28	0.34	37.30
	p3	27.04	27.12	24.20	29.36	29.38	0.955	0.956	0.874	0.962	0.962	0.154	0.142	0.259	0.062	0.047	2.12	1.28	0.33	34.60
ZJU (512×512)	313	29.67	28.82	27.50	28.44	29.11	0.958	0.952	0.939	0.956	0.958	0.084	0.088	0.124	0.068	0.053	2.07	1.51	0.62	51.39
	377	27.13	28.12	25.71	26.18	26.28	0.933	0.949	0.923	0.931	0.930	0.112	0.088	0.152	0.094	0.085	2.41	2.02	0.76	38.70
	386	30.29	30.12	28.51	28.38	28.48	0.938	0.939	0.915	0.919	0.916	0.122	0.112	0.163	0.103	0.078	3.00	4.89	0.91	35.88
H36M (500×500)	s9p	21.53	25.11	26.08	26.03	26.19	0.824	0.912	0.917	0.915	0.916	0.242	0.136	0.139	0.085	0.084	1.06	2.19	0.30	40.00
	s11p	21.27	24.39	25.21	25.20	25.82	0.828	0.899	0.906	0.905	0.911	0.313	0.193	0.174	0.118	0.111	1.18	1.02	0.67	33.41
	s1p	18.91	23.24	23.43	23.83	23.98	0.781	0.909	0.901	0.911	0.911	0.332	0.149	0.162	0.094	0.093	1.38	0.97	0.50	41.43

Table 1. Quantitative results of **novel view synthesis**. We present competitive PSNR and SSIM while outperforming baselines on LPIPS (agrees well with human visual perception [61]) and achieve 30 FPS (pre-computed sparse CNN) available for real-time applications.

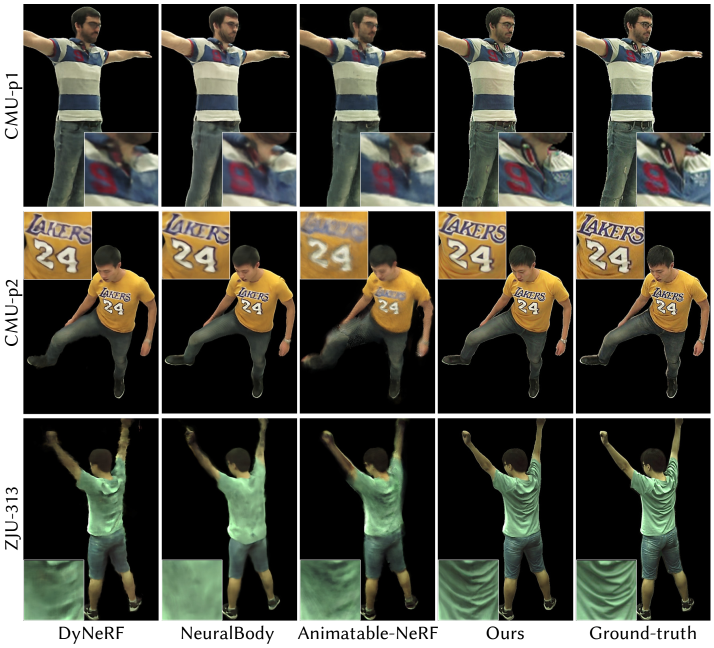


Figure 6. Qualitative results of **novel view synthesis** on CMU Panoptic and ZJU Mocap. Benefiting from spatially aligning the appearance across different poses in a 2D texture, our method produces high-fidelity novel view synthesis, while baselines suffer from blurs (at letters and wrinkles).

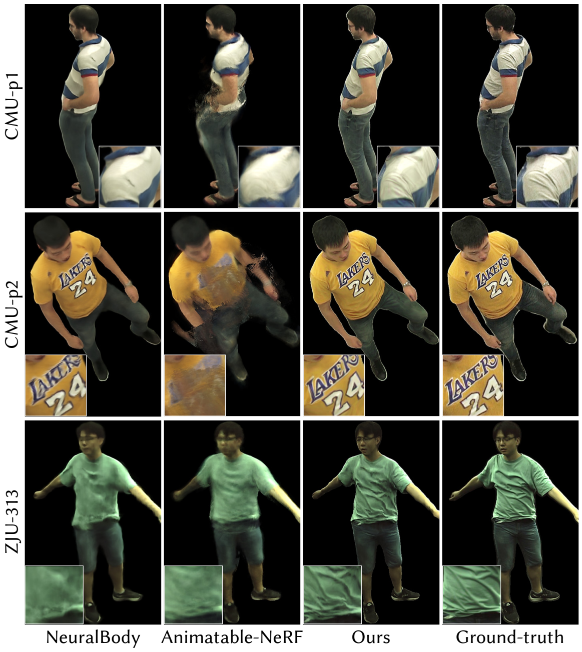


Figure 7. Qualitative results of **novel pose synthesis** on CMU Panoptic, ZJU Mocap. Benefiting from the disentanglement of appearance and geometry, our method performs better on novel poses, especially for preserving sharp details.

further illustrate that our method produces photo-realistic images in different tasks, e.g., novel view synthesis, reposing, reshaping, and retexturing.

Dataset. We perform experiments on several types of datasets which consist of calibrated and synchronized multi-view videos. We use 26 and 20 training views on CMU Panoptic dataset [21] with 960 × 540 resolution and ZJU Mocap dataset [39] with 512 × 512 resolution, respectively. The most challenging one is the H36M dataset [18] with 500 × 500 resolution, where only three cameras are available for training. We obtain the binary human mask by [13]. The evaluation is done on the hold-out cameras (novel views) or hold-out segments of the sequence (novel poses).

Baselines. To validate our method, we compare it against several state-of-the-art free-view video synthesis techniques: 1) DN: DyNeRF [25], which takes time-varying latent codes as the conditions for dynamic scenes; and 2) NB: NeuralBody [39], which takes as input the posed human model with structured time-invariant latent codes and generates a pose-conditioned neural radiance field; 3) AN: Animatable-NeRF [38], which uses neural blend weight fields to generate correspondences between observation and canonical space.

Novel View Synthesis. For comparison, we synthesize images of training poses in hold-out test views. Table 1 shows the comparison of our method against baselines, which demonstrates that our method performs best LPIPS and FPS among all methods. Specifically, we achieve rendering free-

Method	CMU (960×540)			ZJU (512×512)			H36M (500×500)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NB	25.94	0.918	0.146	24.51	0.918	0.120	25.54	0.884	0.170
AN	23.65	0.883	0.208	24.55	0.911	0.153	25.00	0.873	0.170
Ours	26.20	0.927	0.073	23.69	0.910	0.104	25.04	0.874	0.141

Table 2. Quantitative results of **novel pose synthesis**. We achieve competitive PSNR and SSIM while outperforming baselines on LPIPS, which agrees well with humans [61].

view videos of human performances in 30FPS with the help of UV volumes. Note that LPIPS agrees surprisingly well with human visual perception [61], which indicates that our synthesis is more visually similar to ground-truth.

Figure 6 presents the qualitative comparison of our method with baselines. Baselines fail to preserve the sharp image details, whose rendering is blurry and even split. In contrast, our method can accurately capture high-frequency details like letters, numbers and wrinkles on shirts and the belt on pants benefiting from our NTS model. Furthermore, we show the view synthesis results of dynamic humans in Figure 5, which indicate that our method generates high-quality appearance results even with rich textures and challenging motions. Note that the rightmost example is from the H36M dataset with only 4 views. Please refer to the supplementary material for more results.

Reposing. We perform reposing on the human performer with novel motions. As DyNeRF is not designed for editing tasks, we compare our method against NeuralBody and Animatable-NeRF. As shown in Table 2, quantitative results demonstrate that our method achieves competitive PSNR and SSIM while outperforming others on LPIPS.

The qualitative results are shown in Figure 7. For novel human poses, NeuralBody gives blurry and distorted rendering results, while Animatable-NeRF even produces split humans due to a highly under-constrained backward warp field from observation to canonical space. In contrast, synthesized images of our method exhibit better visual quality with reasonable high-definition dynamic textures. The results indicate that using smooth UV volumes in 3D and encoding texture in 2D has better controllability on the novel pose generalization than directly modeling a pose-conditioned neural radiance field.

Reshaping. We demonstrate that our approach can edit the shape of reconstructed human performance by changing the shape parameters of the SMPL model. We illustrate the qualitative results in Figure 1 and Figure 8. NeuralBody fails to infer the reasonable changes of the cloth, while our method generalizes well on novel shapes.

Retexturing. With the learned dense correspondence of UV volumes and neural texture, we can edit the 3D cloth with a user-provided 2D texture, as shown in Figure 9. Visually inspected, the rich texture patterns are well preserved and transferred to correct semantic areas in different poses. Moreover, our model supports changing textures' style and appearance, which are presented in Figure 10. Thanks to the style transfer network [12], we can perform arbitrary artistic stylizations on 3D human performance. Given any fabric texture, we can even dress the performer in various appearances, which enables 3D virtual try-on in real-time.

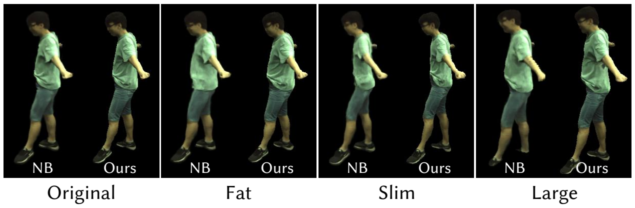


Figure 8. Qualitative results of **reshaping**. By changing the SMPL parameters β, we can conveniently make the human performer fatter, slimmer, or larger. The result of NeuralBody is shown on the left of each image pair, while ours is on the right. Obviously, more details and consistency are preserved by ours in varying shapes.

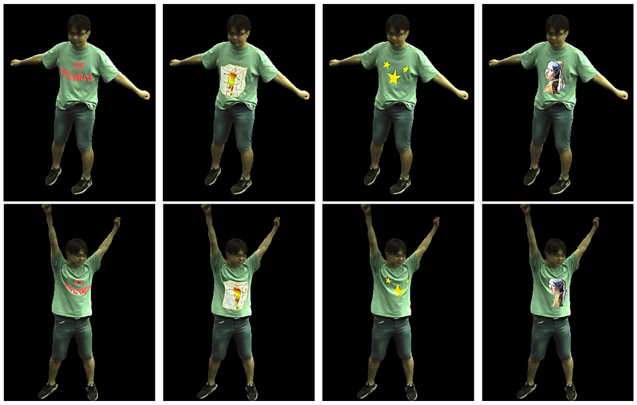
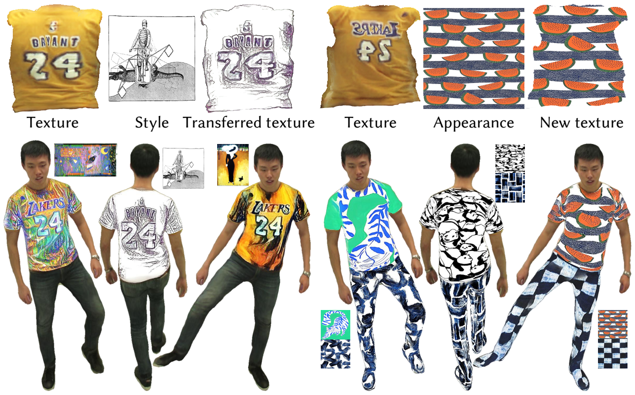


Figure 9. Qualitative results of **retexturing**. The disentanglement of appearance and geometry allows us to conveniently edit the texture by drawing patterns on the NTS. The rich texture patterns are well preserved and transferred to correct semantic areas in different poses, which demonstrates that the texture is not only changed as expected under the edited frame, but also transferred to a novel frame with the modeled dynamics.

4.1. Ablation Studies

We conduct ablation studies on performer p1 of the CMU dataset. As shown in Table 3, we analyze the effects of different losses for the proposed approach by removing warm-start loss, perceptual loss and silhouette loss, respectively. Then, we analyze the time consumption of each module. We encourage the reader to see the supplement for additional ablations, discussion of model design, and other experimental results.

Impact of warm-start loss. We present using semantic and UV-metric loss to warm-start the UV volumes and constrain its solution space. To prove the effectiveness of this process, we train an ablation (No Warm-start Loss) built upon our full model by eliminating the warm-start loss. It gives a lower performance in all metrics, especially the LPIPS in-



(a) Texture transfer (b) Try-on

Figure 10. Given any arbitrary artistic style or cloth appearance, we can render (a) a 3D dynamic human with the transferred texture or perform (b) a 3D virtual try-on in real-time.

Ablations	Novel View Synthesis			Novel Pose Generation		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
No Warm-start Loss	30.37	0.964	0.060	26.14	0.917	0.076
No Perceptual Loss	30.09	0.963	0.055	26.05	0.919	0.079
No Silhouette Loss	17.47	0.874	0.207	16.95	0.860	0.218
Complete Model	30.38	0.966	0.036	26.20	0.927	0.073

Table 3. Ablation study about different objective functions.

creased a lot when rendering novel views. This comparison indicates that the warm-start loss yields better information reuse of different frames by transforming the observation XYZ coordinates to canonical UV coordinates defined by the consistent semantic and UV-metric loss.

Impact of perceptual loss. In contrast to sampling image patches as baselines, we can render an entire image during training, allowing us to use perceptual loss. Table 3 shows that using the same model but training without the perceptual loss (No Perceptual Loss) gives a lower performance in all metrics, especially the PSNR and LPIPS. It demonstrates that the perceptual loss is of critical importance to improving the visual quality of synthesized images, which is also reflected in Table 1 (w/o $\mathcal{L}_p$).

Impact of silhouette loss. To facilitate the UV volumes modeling a more fine-grained geometry, we employ a silhouette loss by using the 2D binary mask of the human performer. We present an ablation (No Silhouette Loss) built upon our full model by eliminating the silhouette loss, as shown in Table 3. It is obvious that No Silhouette Loss gives the worst performance in all metrics among all ablations. This comparison shows that our geometry does benefit from the silhouette loss, which can be seen in the supplement to get an intuitive visual impression.

4.2. Time Consumption

We analyze the time consumption of each module in our framework and the corresponding module in NeuralBody [39] on ZJU Mocap performer 313, as shown in Table 4. On average, it takes 48.78 ms for us to obtain the UV volumes from the posed human model. Then, our method takes only 19.46 ms (51FPS) to access the free-view renderings, which benefits from the smooth UV volumes that allow using much smaller and shallower MLP to obtain densities and texture coordinates in 3D while capturing detailed appearance in 2D NTS. On the contrary, NeuralBody spends 663.84 ms (1.5FPS) to synthesize novel views, which prevents it from being used in applications that require running in real-time. Even on the novel pose generalization task, our method can reach 68.23 ms per frame (14FPS) as well. All experiments are run on a single NVIDIA A100 GPU.

Method	Novel Pose Generation					
	Sparse CNN	Novel View Synthesis				
		Density	Color Model			Rendering
			UV	NTS	RGB	
Ours	48.78	7.08	1.53	7.52	1.60	1.73
			9.12			
		19.46				
	68.23					
NB	52.04	84.38	546.81			32.65
		663.84				
	715.88					

Table 4. Time consumption of each module in milliseconds(ms).

4.3. Limitation

Our method leverages the SMPL model as a scaffold and DensePose as supervision. Consequently, our method can handle clothing types that roughly fit the human body, but fails to correct the prediction from DensePose when handling long hair, loose clothing, accessories, and photorealistic hands. Therefore, the future work is to utilize explicit cloth models and extra hand tracking. While we use time-invariant structured latent codes to encourage temporally consistent UV, a little perturbation caused by the volume generator may occur in the dynamic human (e.g., some unnatural sliding on the trouser when retexturing the performer). It might be improved by adding temporal consistency loss. Replacing the volume representation with other sparse structures for efficiency is also promising.

5. Conclusions

We present the UV volumes for free-view video synthesis of a human performer. It is the first method to generate a real-time free-view video with editing ability. The key is to employ the smooth UV volumes and highly-detailed textures in an implicit neural texture stack. Extensive experiments demonstrate both the effectiveness and efficiency of our method. In addition to improving efficiency, our approach can also support editing, e.g., reposing, reshaping, or retexturing the human performer in the free-view videos.

References

[1] Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabian Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. Driving-signal aware full-body avatars. *ACM Transactions on Graphics (TOG)*, 40(4):1–17, 2021. 3

[2] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001. 2

[3] 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 International Conference on Computer Vision*, 2021. 3

[4] 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. 2

[5] 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. 2

[6] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314, 2012. 2

[7] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. In *Computer graphics forum*, volume 27, pages 409–418, 2008. 2

[8] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2

[9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 2

[10] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 2

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 5

[12] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. In *Proceedings of the British Machine Vision Conference*, pages 114.1–114.12, 2017. 7

[13] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *Proceedings of the European Conference on Computer Vision*, pages 770–785, 2018. 6

[14] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 2

[15] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 3

[16] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2

[17] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 2

[18] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 6

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015. 4

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[21] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):190–204, 2017. 6

[22] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 18(3):165–174, 1984. 4

[23] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016. 2

[24] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2

[25] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vi-*

sion and Pattern Recognition, pages 5521–5531, 2022. 3, 6

[26] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 3

[27] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 3

[28] 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. 3

[29] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3

[30] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 3

[31] Shugao Ma, Tomas Simon, Jason Saragih, Dawei Wang, Yuecheng Li, Fernando De La Torre, and Yaser Sheikh. Pixel codec avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 64–73, 2021. 3

[32] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[33] 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 *European conference on computer vision*, pages 405–421. Springer, 2020. 2, 3, 5

[34] Armin Mustafa, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton. Temporally coherent 4d reconstruction of complex dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4660–4669, 2016. 2

[35] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[36] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 3

[37] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021. 3

[38] 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 International Conference on Computer Vision*, 2021. 3, 6

[39] 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. 3, 5, 6, 8

[40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 3

[41] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310, 2019. 4

[42] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14335–14345, 2021. 2

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

[44] Ruizhi Shao, Hongwen Zhang, He Zhang, Mingjia Chen, Yan-Pei Cao, Tao Yu, and Yebin Liu. Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15872–15882, 2022. 3

[45] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019. 3

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[47] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 2

[48] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled edit-

ing for high-resolution 3d-aware portrait synthesis. *ACM Transactions on Graphics (TOG)*, 41(6):1–10, 2022. 2

[49] Jingxiang Sun, Xuan Wang, Yong Zhang, Xiaoyu Li, Qi Zhang, Yebin Liu, and Jue Wang. Fenerf: Face editing in neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7672–7682, 2022. 2

[50] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[51] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Nonrigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 3

[52] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the Conference on International Conference on Machine Learning*, volume 1, page 4, 2016. 5

[53] Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13524–13534, 2022. 3

[54] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16210–16220, 2022. 3

[55] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1682–1691, 2020. 3

[56] Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7128, 2021. 2

[57] 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, 2021. 3

[58] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2

[59] Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5746–5756, 2021. 3

[60] Jiakai Zhang, Xinhang Liu, Xinyi Ye, Fuqiang Zhao, Yanshun Zhang, Minye Wu, Yingliang Zhang, Lan Xu, and Jingyi Yu. Editable free-viewpoint video using a layered neural representation. *ACM Transactions on Graphics (TOG)*, 40(4):1–18, 2021. 3

[61] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 6, 7