Geometry-Guided Progressive NeRF for Generalizable and Efficient Neural Human Rendering

Mingfei Chen[1,2], Jianfeng Zhang[3], Xiangyu Xu[1], Lijuan Liu[1], Yujun Cai[1], Jiashi Feng[1], and Shuicheng Yan[1]

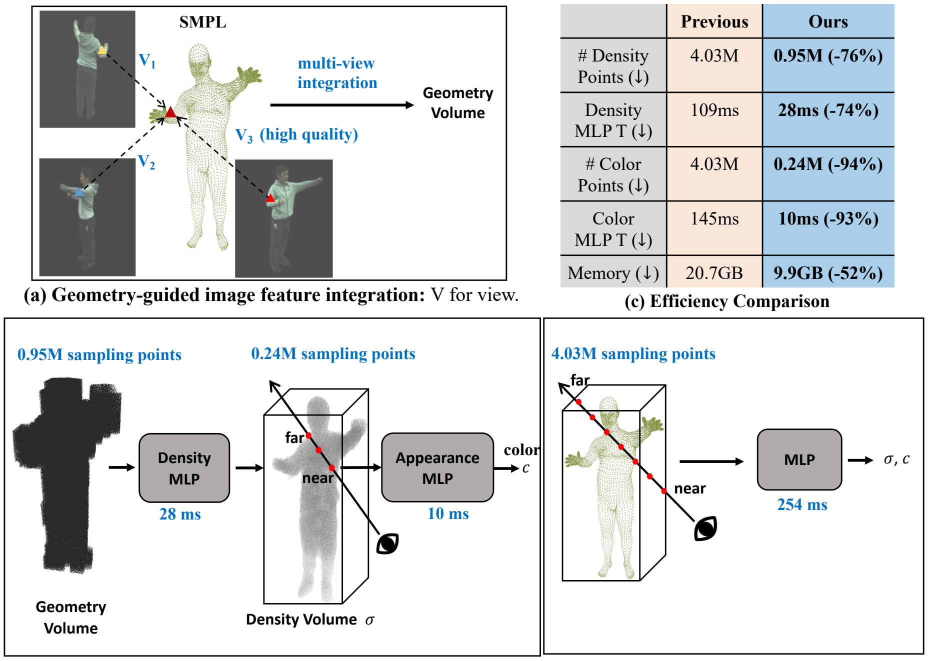
[1] Sea AI Lab
[2] University of Washington
[3] National University of Singapore

Abstract. In this work we develop a *generalizable* and *efficient* Neural Radiance Field (NeRF) pipeline for high-fidelity free-viewpoint human body synthesis under settings with *sparse* camera views. Though existing NeRF-based methods can synthesize rather realistic details for human body, they tend to produce poor results when the input has self-occlusion, especially for unseen humans under sparse views. Moreover, these methods often require a large number of sampling points for rendering, which leads to low efficiency and limits their real-world applicability. To address these challenges, we propose a Geometry-guided Progressive NeRF (GP-NeRF). In particular, to better tackle self-occlusion, we devise a geometry-guided multi-view feature integration approach that utilizes the estimated geometry prior to integrate the incomplete information from input views and construct a complete geometry volume for the target human body. Meanwhile, for achieving higher rendering efficiency, we introduce a progressive rendering pipeline through geometry guidance, which leverages the geometric feature volume and the predicted density values to progressively reduce the number of sampling points and speed up the rendering process. Experiments on the ZJU-MoCap and THUman datasets show that our method outperforms the state-of-the-arts significantly across multiple generalization settings, while the time cost is reduced $> 70\%$ via applying our efficient progressive rendering pipeline.

1 Introduction

High-fidelity free-viewpoint synthesis of human body is important for many applications such as virtual reality, telepresence and games. Some recent works [22,12,25,36] deploy a Neural Radiance Fields (NeRF) [18] pipeline, which achieved fairly realistic synthesis of human body. However, these works usually require dense-view capturing of human body, and have to train a separate model for each person to render new views. The limited generalization ability as well as demand for cost computation severely hinder their application in the real-world scenarios.

In this work, we aim at boosting high-fidelity free-viewpoint human body synthesis with a generalizable and efficient NeRF framework based on only single-frame images from sparse camera views. To pursue such a high-standard framework, there are mainly two challenges that need to be tackled. First, the human body is highly non-rigid and commonly has self-occlusions over body parts, which may lead to ambiguous results



(a) Geometry-guided image feature integration: V for view.

	Previous	Ours
# Density Points (↓)	4.03M	**0.95M (-76%)**
Density MLP T (↓)	109ms	**28ms (-74%)**
# Color Points (↓)	4.03M	**0.24M (-94%)**
Color MLP T (↓)	145ms	**10ms (-93%)**
Memory (↓)	20.7GB	**9.9GB (-52%)**

(c) Efficiency Comparison

(b) Rendering pipeline: our efficient geometry-guided progressive pipeline (left) vs. previous (right).
The amount of sampling points and forward time in blue are measured on the same data and model parameters.

Fig. 1. Our method can better handle self-occlusion (a) and high computational cost (b) issues than previous methods [10,24]. In (a), our multi-view integration can extract high-quality geometry information from V_3 for the red SMPL vertex. In (b), our progressive rendering pipeline leverages the geometric volume and the predicted density values to progressively reduce the number of sampling points and speed up the rendering, while previous methods [10,24] wastes large amount of computations at redundant empty regions. The efficiency comparison shown in (c) further verifies our high efficiency.

with only sparse-view captures. This ambiguity could drastically degrade the rendering quality without proper regularizations, which cannot be easily solved by simply sampling features from multi-view images as in [35,33,26]. This problem would become worse when using one model to synthesize unseen scenarios without specific per-scene training. Second, the high computation and memory cost of NeRF-based methods severely hinder human synthesis with accurate details in high-resolution. For example, when rendering one 512×512 image, existing methods need to process millions of sampling points through the neural network, even if using the bound of the geometry prior to remove empty regions.

To address these challenges, we propose a geometry-guided progressive NeRF, called GP-NeRF, for generalizable and efficient free-view human synthesis. More specifically, to regularize the learned 3D human representation, we propose a geometry-guided multi-view feature integration approach to more effectively exploit the information in the sparse input views. For the geometry prior, we adopt a coarse 3D body model, *i.e.*, SMPL [17], which serves as a base estimate of our algorithm. We attach multi-view image features to the base geometry model using an adaptive multi-view aggregation layer. Then we can obtain an enhanced geometry volume by refining the base model with the attached image features, which substantially reduces the ambiguities in learning a high-fidelity 3D neural field. It is worth noting that our multi-view enhanced geometry prior differs significantly from related methods that also utilize human body priors [24,10,14,23]. NB [24] learns a per-scene geometry embedding, which is hard to generalize to unseen human bodies; NHP [10] relies on temporal information to complement the base geometry model, which is less effective for regions occluded throughout the input video. In contrast, our approach can adaptively combine the geometry prior and multi-view features to enhance the 3D estimation, and thus can better handle the self-occlusion problem and acquire lifted generalization capacity even without using videos (see Figure 1 (a)). By integrating the multi-view information and form a complete geometry volume adapting to the target human body, we can also compensate some limitations of the geometry prior (e.g., inaccurate body shape or lacks cloth information as in [14,23]), and support our following efficiency progressive pipeline.

Furthermore, to tackle the high computation and memory cost, we introduce a geometry-guided progressive rendering pipeline. As shown in Figure 1 (b), different from previous methods [24,10], our pipeline decouples the density and color prediction process, leveraging the geometry volume as well as the predicted density values to reduce the number of sampling points for rendering progressively. By simply deploying our progressive rendering pipeline with the same data and model parameters, we can remove 76.4% points for density prediction (with Density MLP in Figure 1 (b)) and 94% points for color prediction (with Appearance MLP in Figure 1 (b)), reducing the total forwarding time of this part for all points by 85%. Later experiments verify that our progressive pipeline causes no performance decline while requiring shorter training time, which is credited to focusing on the density and appearance learning separately.

Our main contributions are in three folds:

- We propose a novel geometry-guided progressive NeRF (GP-NeRF) for generalizable and efficient human body rendering, which reduces the computational cost of rendering significantly and also gains higher generalization capacity simply based on the single-frame sparse views.
- We propose an effective geometry-guided multi-view feature integration approach, where we let each view compensate the low-quality occluded information for other views with the guidance of the geometry prior.
- Our GP-NeRF has achieved state-of-the-art performance on the ZJU-MoCap dataset, taking only 175ms on RTX 3090 and reducing time for rendering per image by over 70%, which well verifies effectiveness and efficiency of our framework.

2 Related Work

Human Performance Capture. Previous works [20,2,5,8] apply traditional modeling and rendering pipelines for novel view synthesis of human performance, relying on either dense camera setup [4,8] or depth sensors [2,5,31] to ensure photo-realistic reconstruction. Follow-up improvements are made by introducing neural networks to the rendering pipeline to alleviate geometric artifacts. To enable human performance capture in a sparse multi-view setup, template-based methods [1,3,6,30] adopt pre-scanned human models to track human motion. However, these approaches require per-scene optimization and the pre-scanned human models are hard to collect in practice, which hinders them from real-world applications. Instead of performing per-scene optimization, recent methods [19,27,28,37] adopt neural networks to learn human priors from ground-truth 3D data, and hence can reconstruct detailed 3D human geometry and texture from a single image. However, due to the limited diversity of training data, it is difficult for them to generate photo-realistic view synthesis or generalize to human poses and appearances that are very different from the training ones. And some other methods [11,38] sample points from the generated 3D feature space, and then decide the human body opacity for later rendering, but they might generate results that violate the normal human body structures without involving human body geometry constraints.

Neural 3D Representations. Recently, researchers propose implicit function-based approaches [29,13,15,21] to learn a fully-connected network to translate a 3D positional feature into local feature representation. A very recent work NeRF [18] achieves high fidelity novel view synthesis by learning implicit fields of color and density along with a volume rendering technique. Later, several works extend NeRF to dynamic scenes modeling [22,25,36,12] by optimization NeRF and dynamic deformation fields jointly. Despite impressive performance, it is an extremely under-constrained problem to learn both NeRF and dynamic deformation fields together. NB [24] combines NeRF with a parametric human body model SMPL [17] to regularize the training process. It requires a lengthy optimization for each scene and hardly generalizes to unseen scenarios. To avoid such expensive per-scene optimization, Generalizable NeRFs [26,33,35,10] condition the network on the pixel-aligned image features. However, directly extending such methods to complex and dynamic 3D human modeling is highly non-trivial due to self-occlusion, especially when modeling unseen humans under sparse views. Besides, these approaches suffer low efficiency since they need to process a large number of sampling points for volumetric rendering, harming their real-world applicability. Different from existing methods, we carefully design a multi-view information aggregation approach and a progressive rendering technique to improve model robustness and generalization to unseen scenarios under sparse views and also speed up the rendering.

3 Methodology

Given a set of M sparse source views $\{\mathbf{I}_m | m = 1, 2, ..., M\}$ of an arbitrary human model, which are captured by M pre-calibrated cameras respectively, we aim to synthesize the novel view $\mathbf{I}_t$ of the human model from an arbitrary target camera.

To this end, we propose a geometry-guided progressive NeRF (GP-NeRF) framework for efficient and generalizable free-view human synthesis under very sparse views (e.g., $M = 3$). Figure 2 illustrates the overview of our framework. Firstly, a CNN backbone is used to extract image features $\mathbf{F}_m$ for each of the views $\mathbf{I}_m$. Then our GP-NeRF framework integrates these multi-view features to synthesize the novel-view image through three modules progressively, leveraging the geometry prior from SMPL [17] as guidance. The three modules are 1) geometry-guided multi-view feature integration (GMI) module (Section 3.1); 2) density network (Section 3.2); and 3) appearance network (Section 3.3). Details of the whole progressive human rendering pipeline are elaborated in Section 3.4, and the training method is described in Section 3.5.

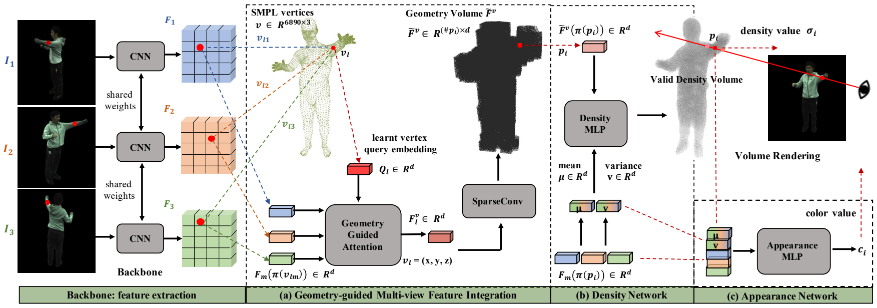


Fig. 2. Overview of our proposed framework. Our progressive pipeline mainly contains three parts. **(a) Geometry-guided multi-view feature integration.** We first learn query embedding Q_l for each SMPL vertex to adaptively integrate the multi-view pixel-aligned image features $\mathbf{F}_m(\pi(v_{lm}))$ through the geometry-guided attention module. Based on this, we utilize the SparseConvNet to construct a denser geometry feature volume $\tilde{\mathbf{F}}^v$. **(b) Density Network.** For point p_i within $\tilde{\mathbf{F}}^v$, we concatenate its geometry feature $\tilde{\mathbf{F}}_i^v$ with the mean ($\boldsymbol{\mu}$) and variance ($\mathbf{v}$) of its pixel-aligned image features $\mathbf{F}_m(\pi(p_i))$, and predict its density value σ_i through the density MLP. p_i with a positive σ_i form the valid density volume. **(c) Appearance Network.** For point p_i within the valid density volume, we utilize $\mathbf{F}_m(\pi(p_i))$ to predict its color value c_i through the appearance MLP. Finally, we conduct the volume rendering to render the target image.

3.1 Geometry-guided Multi-view Integration

The geometry-guided multi-view feature integration module, shown in Figure 2 (a), enhances the coarse geometry prior with multi-view image features by adaptively aggregating these features via a geometry-guided attention module. Then it constructs a complete geometry feature volume that adapts to the target human body.

Firstly, we use the SMPL model [17] as the geometry prior, and get the pixel-aligned image features for each of the 6890 SMPL vertices v_l from each source image $\mathbf{I}_m$. Specifically, we multiply the coordinate of v_l with each source camera pose $[\mathbf{R}_m|\mathbf{t}_m]$ to transform the original v_l to v_{lm} into the source camera coordinate system, and then utilize the intrinsic matrix $\mathbf{K}_m$ to obtain the projected coordinate $\pi(v_{lm})$ in the corresponding image plane. We denote the pixel-aligned features from the image features $\mathbf{F}_m$ that corresponds to the pixel location of $\pi(v_{lm})$ as $\mathbf{F}_m(\pi(v_{lm}))$. We use bilinear interpolation to obtain the corresponding features if the projected location is fractional.

After obtaining $\mathbf{F}_m(\pi(v_{lm}))$ from M source views, we integrate them to represent the geometry information at vertex v_l through a geometry-guided attention module. Concretely, we learn an embedding $\mathbf{Q}_l$ for each v_l, and then take $\mathbf{Q}_l$ as a query embedding to calculate the correspondence score s_{lm} with each $\mathbf{F}_m(\pi(v_{lm}))$ respectively:

$$s_{lm}^{v} = \frac{(\boldsymbol{W}_1\mathbf{Q}_l + b_1)(\boldsymbol{W}_{2m}\mathbf{F}_{lm}^{v} + b_{2m})^\top}{\sqrt{d}}, \tag{1}$$

where we denote $\mathbf{F}_m(\pi(v_{lm}))$ as $\mathbf{F}_{lm}^{v}$ for simplicity. d is the channel dimension of $\mathbf{F}_{lm}^{v}$. $\boldsymbol{W}$ represents linear projection layers. After that, we weighted sum the M pixel-aligned feature embeddings $\mathbf{F}_m(\pi(v_{lm}))$ based on the scores s_{lm} to obtain the aggregated geometry related feature $\mathbf{F}_l^v$ for vertex v_l:

$$\mathbf{F}_l^v = \sum_{m=1}^{M} s_{lm}^{v}\mathbf{F}_{lm}^{v}. \tag{2}$$

Considering the $6,890$ SMPL vertices with their corresponding features are not dense enough to represent the whole human body volume, we further learn to extend and fill the holes of the sparse geometry feature volume $\mathbf{F}^v = \{\mathbf{F}_l^v, l = 1, 2, ..., 6890\}$ through the SparseConvNet [7] and thus obtain a denser geometry feature volume, denoted as $\tilde{\mathbf{F}}^v$. In our method, we take the geometry volume $\tilde{\mathbf{F}}^v$ as a more reliable basis to indicate occupancy of the human body in the whole space volume. More advanced than the coarse model SMPL, $\tilde{\mathbf{F}}^v$ leverages the multi-view image-conditioned features to enhance the coarse geometry prior, which adapts to the shape of the target human body. $\tilde{\mathbf{F}}^v$ only preserves the effective volume regions with body contents, including clothes regions. Because the SparseConvNet can gain experience from training to extend the features towards the regions with contents, based on the image-conditioned features with some instructive context information at each feature point. Besides, the geometry volume will also benefit our progressive rendering pipeline, which will be detailed in Section 3.4.

3.2 Density Network

The density network predicts the opacity of each sampling point $\mathbf{p}_i$, which is highly related to the geometry of human body, like postures and shapes. Through the geometry-guided multi-view integration module in Section 3.1, we can construct a geometry feature volume $\tilde{\mathbf{F}}^v$ which can provide sufficient reliable geometry information of the target human body. As shown in Figure 2 (b), for each sampling point $\mathbf{p}_i$, we obtain its corresponding geometry related feature $\tilde{\mathbf{F}}_i^v$ from $\tilde{\mathbf{F}}^v$ based on its coordinate. Though the feature volume can provide the geometry information of human body, such geometry-related features are coarse and may lose some fine image-conditioned features that benefit the high-fidelity rendering. To compensate the information loss, we combine these two kinds of features at each sampling point to predict its density value more accurately. Therefore, we concatenate $\tilde{\mathbf{F}}_i^v$ with the mean ($\boldsymbol{\mu}$) and variance ($\mathbf{v}$) feature embedding of its corresponding pixel-aligned image features $\{\mathbf{F}_m(\pi(v_{lm})), m = 1, 2, ..., M\}$ that contain more detailed information, and process the concatenated feature through a density MLP to predict the density value at this point.

3.3 Appearance Network

The appearance network aims to predict the RGB color value for each sampling point $\mathbf{p}_i$. Since the RGB value is more related to the appearance details of human body, we utilize the image-conditioned features as the input to the appearance network for more detailed information. As shown in Figure 2 (c), we first aggregate the pixel-aligned image features from input views for each color sampling point $\mathbf{p}_i^c$. Specifically, similar to obtaining the pixel-aligned image features for each SMPL vertex, we project the coordinate of $\mathbf{p}_i$ to the image plane of each source view, and obtain the pixel-aligned feature embedding, denoted as $\mathbf{F}_m(\pi(\mathbf{p}_i))$. We then concatenate $\mathbf{F}_m(\pi(\mathbf{p}_i))$ from M source views with their mean ($\boldsymbol{\mu}$) and variance ($\mathbf{v}$) feature embeddings together. Afterwards, based on the concatenated feature embeddings, an appearance MLP is deployed to predict the RGB value $\hat{\mathbf{c}}_i = (\hat{r}_i, \hat{g}_i, \hat{b}_i)$ for the corresponding point $\mathbf{p}_i$.

3.4 Geometry-guided Progressive Rendering

We render the human body in the target view through the volumetric rendering following previous NeRF-based methods [18,24,10]. Instead of sampling many redundant points for rendering, we introduce an efficient geometry-guided progressive rendering pipeline for the inference process. Our pipeline leverages the geometry volume in Section 3.1 as well as the predicted density values in Section 3.2 to reduce the number of points progressively.

Specifically, we first preserve the sampling points that occupy the geometry volume $\tilde{\mathbf{F}}^v$ as valid density sampling points $\mathbf{p}_i^d$. Compared to the smallest pillar that contains the human body that is used by previous methods [24,10], the geometry volume is closer to the human body shape and contains much fewer redundant void sampling points. Then we predict the density values for $\mathbf{p}_i^d$ through the density network, and the sampling points that have positive density values form a valid density volume. As shown in Figure 2, the valid density volume is very close to the 3D mesh of the target human body and we further remove many empty regions compared to the geometry volume. We take the sampling points in the valid density volume as the new valid sampling points $\mathbf{p}_i^c$, and further predict their color values through the appearance network in Section 3.3.

We conduct volume rendering based on the density and color predictions to synthesize the target view $\mathbf{I}_t$. Traditional volume rendering methods often march rays $\mathbf{r}$ from the target camera to the pixels of the target view image, and then sample N points on each $\mathbf{r}$. Denoting the distance of two adjacent sampling points on $\mathbf{r}$ as δ, we can formulate the color rendering process for each $\mathbf{r}$ as:

$$\begin{aligned} \hat{C}(\mathbf{r}) &= \sum_{i=1}^{N} T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right) \hat{\mathbf{c}}_i, \\ \text{where } T_i &= \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right). \end{aligned} \tag{3}$$

For our progressive rendering pipeline, we use projection to bind the sampling points to $\mathbf{r}$. Concretely, we project the points within the geometry volume to the target view, take the nearest four pixels of the projected points as valid pixels to march a ray, and then uniformly sample N points between its near and far bounds as [24,10]. We only process the sampling points within the valid volume regions and then conduct volume rendering based on the rays $\mathbf{r}$.

Experiments in Section 4.4 verify that our geometry-guided progressive rendering pipeline reduces the memory and time consumption during rendering significantly, and our performance can be even lifted by removing noisy unnecessary sampling points.

3.5 Training

During training, we do not deploy the progressive rendering pipeline in Section 3.4, because it is useful only when our density network is reliable. Instead, we march rays from the target camera to pixels randomly sampled on the image while ensuring no fewer than half of the pixels are on the human body. We uniformly sample points on the rays to predict the corresponding density and color values. By performing the volume rendering in Eq. (3), we obtain the predicted color $\hat{C}(\mathbf{r})$ for each $\mathbf{r}$. To supervise the network, we calculate the Mean Square Error loss between $\hat{C}(\mathbf{r})$ and the corresponding ground truth $C(\mathbf{r})$ color value as our training loss $\mathcal{L}_{rgb}$.

4 Experiments

We study four questions in experiments. 1) Is GP-NeRF able to improve the fitting and generalization performance of human synthesis on the seen and unseen scenarios (Section 4.3)? 2) Is GP-NeRF effective at reducing the time and memory cost for rendering (Section 4.4)? 3) How does each individual design choice affect model performance (Section 4.5) 4) Can GP-NeRF provide promising results, both for human rendering and 3D reconstruction (Section 4.6)? We describe the datasets and evaluation metrics in Section 4.1, and our default implementation setting in Section 4.2.

4.1 Datasets and Metrics

We train and evaluate our method on the ZJU-MoCap dataset [24] and THUman 1.0 dataset [37]. ZJU-MoCap contains 10 sequences with 21 synchronized cameras. We split the 10 sequences into a training set with 7 sequences and a test set with the remaining 3 sequences, following [10] for a fair comparison. THUman contains 202 human body 3D scans. 80% of the scans are taken as the training set, and the remaining are the test set. We render images for each scan from 24 virtual cameras, which are uniformly set on the horizontal plane.

To evaluate the rendering performance, we choose two metrics: peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) following [18,24]. For the 3D reconstruction, we only provide the qualitative results since the corresponding ground truth is not available.

4.2 Implementation Details

In our implementation, we perform training and inference with an image size of 512×512 under $M = 3$ camera views, where the horizontal angle interval is around $120°$ (Uniform). We utilize a U-Net like architecture [33] as our backbone to extract the image features $\mathbf{F}$ in Section 3 with a dimension of 32. We sample $N = 64$ points uniformly between the near and far bound on each ray. For training, we utilize the Adam optimizer [9], and the learning rate decays exponentially from $1e-4$ for 180k steps. We use one RTX 3090 GPU with a batch size of 1 for both training and inference.

Table 1. Synthesis performance comparison. Our proposed method outperforms existing methods on all the settings.

	Dataset		Per-scene	Unseen		Results	
Method	Train	Test	training	Pose	Body	PSNR (↑)	SSIM (↑)
Performance on training frames							
NT [32]	ZJU-7	ZJU-7	✓	✗	✗	23.86	0.896
NHR [34]	ZJU-7	ZJU-7	✓	✗	✗	23.95	0.897
NB [24]	ZJU-7	ZJU-7	✓	✗	✗	28.51	**0.947**
NHP [10]	ZJU-7	ZJU-7	✗	✗	✗	28.73	0.936
GP-NeRF (Ours)	ZJU-7	ZJU-7	✗	✗	✗	**28.91**	0.944
Performance on unseen frames from training data							
NV [16]	ZJU-7	ZJU-7	✓	✓	✗	22.00	0.818
NT [32]	ZJU-7	ZJU-7	✓	✓	✗	22.28	0.872
NHR [34]	ZJU-7	ZJU-7	✓	✓	✗	22.31	0.871
NB [24]	ZJU-7	ZJU-7	✓	✓	✗	23.79	0.887
NHP [10]	ZJU-7	ZJU-7	✗	✓	✗	26.94	0.929
GP-NeRF (Ours)	ZJU-7	ZJU-7	✗	✓	✗	**27.92**	**0.934**
Performance on test frames from test data							
NV [16]	ZJU-3	ZJU-3	✓	✓	✗	20.84	0.827
NT [32]	ZJU-3	ZJU-3	✓	✓	✗	21.92	0.873
NHR [34]	ZJU-3	ZJU-3	✓	✓	✗	22.03	0.875
NB [24]	ZJU-3	ZJU-3	✓	✓	✗	22.88	0.880
PVA [26]	ZJU-7	ZJU-3	✗	✓	✓	23.15	0.866
Pixel-NeRF [35]	ZJU-7	ZJU-3	✗	✓	✓	23.17	0.869
NHP [10]	ZJU-7	ZJU-3	✗	✓	✓	24.75	0.906
GP-NeRF (Ours)	ZJU-7	ZJU-3	✗	✓	✓	**25.96**	**0.921**
Generalization performance across datasets							
NHP [10]	AIST	ZJU-3	✗	✓	✓	17.05	0.771
GP-NeRF (Ours)	THUman-7	ZJU-3	✗	✓	✓	24.74	0.907
GP-NeRF (Ours)	THUman-all	ZJU-3	✗	✓	✓	**25.60**	**0.917**

4.3 Synthesis Performance Analysis

In Table 4, we compare our human rendering results to previous state-of-the-art methods. To evaluate the capacity of fitting and generalization on different levels, we train our framework on the first 300 frames of 7 training video sequences of ZJU-MoCap (ZJU-7), and test on 1) the training frames, 2) unseen frames of ZJU-7, and 3) test frames from the 3 test sequences (ZJU-3), respectively. The results in Table 4 verify our advanced generalization capacity on the unseen scenarios. We also achieve competitive fitting performance on the training frames, even comparable to the per-scene optimization methods [32,34,24].

Notably, our method outperforms the state-of-the-art NHP [10] which utilizes the geometry prior with features of multi-view videos. Specifically, for the unseen poses and the unseen bodies, we outperform NHP by 0.98 and 1.21 dB on PSNR, and also by 0.5% and 1.5% on SSIM respectively, using only single-frame input. We also conduct generalization experiments across two datasets with different domains. We train our model on 7 random human bodies from the THUman dataset (THUman-7) and all 202 human bodies (THUman-all) separately, and test the synthesis performance on the test frames of ZJU-3. From Table 4, we observe our method outperforms NHP by a large margin under cross-dataset evaluation setup, i.e., around 7.7 dB and 13.6% improvements on PSNR and SSIM respectively. All these results demonstrate the effectiveness of our geometry-guided multi-view information integration approach.

4.4 Efficiency Analysis

In Table 2, we analyze the efficiency improvements [4] gained from our progressive pipeline on the first 300 frames of the 315 (Taichi) sequence in ZJU-MoCap dataset.

Table 2. Computation and memory cost comparison. GP-NeRF† has the same structure as our GP-NeRF but adopts vanilla rendering technique. $\times N$ indicates the sampling points are split into N chunks to be processed. #$\mathbf{r}$ means the number of sampling rays; #$\mathbf{p}^d$ and #$\mathbf{p}^c$ mean sampling points through the density network and appearance network, respectively. T^d-total indicates the total time cost from backbone output to the density volume, including T^d-MLP which means the forwarding time of the density MLP. T^c-total means the time from density volume to the color prediction, and T^c-MLP is the time for the appearance MLP.

Method	#$\mathbf{r}$ (M) (↓)	#$\mathbf{p}^d$ (M) (↓)	#$\mathbf{p}^c$ (M) (↓)	Time (ms) (↓)	Mem (GB) (↓)
NHP [10]	0.063	4.03	4.03	1160	14.20
NHR [34]	0.063	4.03	4.03	636	10.20
NB [24]	0.063	4.03	4.03	611	21.80
GP-NeRF† 3×	0.063 (-0.0%)	4.03 (-0.0%)	4.03 (-0.0%)	589 (-3.6%)	14.53 (-33.3%)
GP-NeRF† 2×	0.063 (-0.0%)	4.03 (-0.0%)	4.03 (-0.0%)	567 (-7.2%)	20.74 (-4.9%)
GP-NeRF 2×	**0.039 (-38.1%)**	**0.95 (-76.4%)**	**0.24 (-94.0%)**	243 (-60.2%)	**9.88 (-54.7%)**
GP-NeRF 1×	**0.039 (-38.1%)**	**0.95 (-76.4%)**	**0.24 (-94.0%)**	**175 (-71.4%)**	14.25 (-34.6%)

Method	T^d-MLP (ms) (↓)	T^d-total (ms) (↓)	T^c-MLP (ms) (↓)	T^c-total (ms) (↓)	PSNR (↑)
GP-NeRF† 2×	108.58	226.56	145.38	146.39	26.56
GP-NeRF 2×	28.08 (-74.1%)	83.65 (-63.1%)	10.02 (-93.1%)	11.4 (-92.2%)	**26.67 (+0.4%)**
GP-NeRF 1×	**23.55 (-78.3%)**	**74.07 (-67.3%)**	**9.50 (-93.5%)**	**10.27 (-93.0%)**	**26.67 (+0.4%)**

Considering the limited GPU memory, our final GP-NeRF can process all the sampling points in one run, but GP-NeRF† and NB [24] requires at least twice. As shown in the upper panel of Table 2, compared to NB, NHR and NHP which also use the SMPL bounds to remove redundant marched rays, our GP-NeRF can further remove 38.1% rays and 76.4% #$\mathbf{p}^d$ by referring to the constructed geometry volume, and remove 94.0% #$\mathbf{p}^c$ based on the valid density volume. Comparing to NB, NHR and NHP, our GP-NeRF $2\times$ costs $60\% - 79\%$ less time with lower memory. For fair comparison to GP-NeRF† $2\times$, we also test the speed on GP-NeRF for 2 chunks, and our progressive pipeline still reduces the time cost by 57% and the memory cost by 52.4%, which verifies the significant efficiency improvement from the proposed rendering pipeline.

In the bottom panel of Table 2, we compare the time cost of each component in GP-NeRF to GP-NeRF† without progressive points reduction. The results show that we can

[4] We count averaged per-sample inference time in milliseconds. For all methods, the time is counted on NVIDIA GeForce RTX 3090 and CPU Intel i7-11700 @ 2.50GHz, PyTorch 1.8, CUDA 11.4.

reduce over 74% and 63% time cost for density MLP forwarding and the total density related time T^d-total respectively, by simply using our progressive rendering pipeline on the same network structures. Our pipeline can also reduce over 92% time cost for the appearance MLP forwarding. Moreover, our progressive pipeline improves the efficiency significantly while even improving the PSNR metric by 0.4%, as it can ignore some noisy sampling points during rendering that might degrade the performance.

Table 3. Ablations: feature integration. G, Q, P are different approaches to obtain input features for the shared density and appearance network. G: geometry feature volume; Q: integrate multi-view information at each geometry vertex; P: pixel-aligned image features.

Variants	G	Q	P	PSNR (↑)	SSIM (↑)
G	✓	✗	✗	23.47	0.880
QG	✓	✓	✗	23.68	0.885
P	✗	✗	✓	26.09	0.915
QG+P	✓	✓	✓	**26.69**	**0.924**

Table 4. Ablations: progressive structure. G, Q, P have the same meanings as Table 3. Disentangle indicates whether the density and appearance networks are in a progressive pipeline. Steps mean the number of training steps. The columns of Density and Appearance demonstrate components of the input features.

Disentangle	Density	Appearance	Steps	PSNR (↑)	SSIM (↑)
✗	QG+P	QG+P	5000	26.05	0.912
✓	QG+P	QG+P	5000	26.13	0.917
✓	QG+P	P	5000	26.16	0.920
✓	QG	P	5000	25.71	0.904
✗	QG+P	QG+P	35000	26.69	0.924
✓	QG+P	QG+P	35000	26.65	0.925
✓	QG+P	P	35000	26.67	0.923
✓	QG	P	35000	26.40	0.918

4.5 Ablation Studies

We conduct ablation studies under the *uniform camera setting* in Section 4.2 to verify effectiveness of our main designed components on generalization capacity. We train our model on 7 training sequences of the ZJU-MoCap dataset for 35k steps and validate it on remaining 3 sequences.

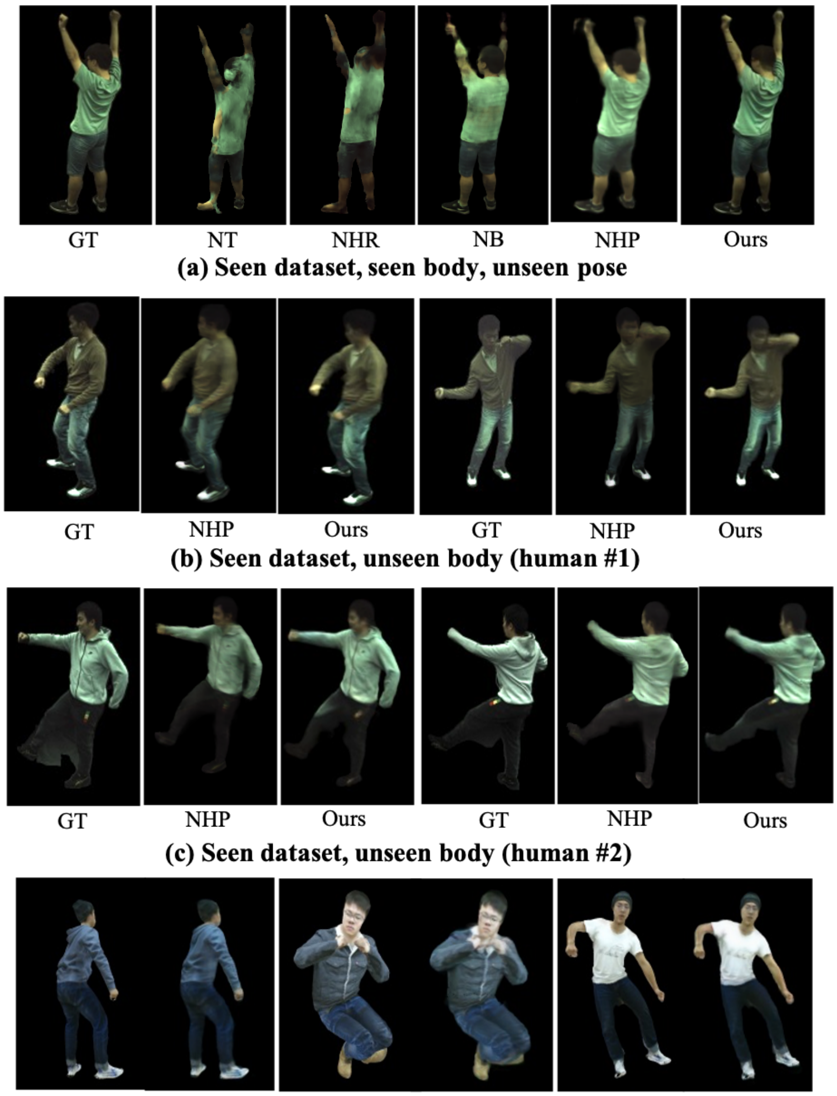
Feature Integration. In Table 3, we explore the effectiveness of the proposed geometry-guided feature integration mechanism on the baseline GP-NeRF, i.e., GP-NeRF without adopting progressive rendering pipeline. As shown in Table 3, adaptively aggregating multi-view image features with the guidance of the geometry prior to construct the geometry feature volume (QG) achieves better performance (i.e., 0.21 dB and 0.5% improvements on PSNR and SSIM respectively) than baseline that simply uses the mean of multi-view image features (G), as the proposed geometry-guided attention module helps focus more on the views corresponding to the geometry prior. We also observe baseline using only pixel-aligned image features (P) gains 2.41 dB PSNR and 3% SSIM over baseline using only geometry feature (G), as it captures more detailed appearance features from images for high-fidelity rendering. Moreover, by combining the geometry feature and its corresponding detailed image features (QG+P), we can improve upon P by 0.6 dB PSNR and 0.9% SSIM respectively. This indicates that both the geometry and the pixel-aligned image features can compensate each other for better generalization performance on unseen scenarios.

Progressive Structure. Our progressive rendering pipeline in Section 3.4 requires a progressive structure of the density and appearance network. Based on the same experimental settings, we further decouple the density and appearance networks to form a progressive pipeline as in Figure 2 and evaluate the performance. As shown in Table 4, the progressive structure does not harm the performance and even reaches relatively high performance faster. This is because it allows these two networks to lean their different focus, thus improving the performance more quickly during training. For the density network, involving more detailed image features P can enhance the relatively coarse geometry feature QG, and bring around 0.5% improvements on SSIM. The results also show that the geometry feature QG is much more impactful on the geometry-related density prediction than on the appearance-related color value prediction.

4.6 Visualization

We visualize our human rendering results under three uniform camera views in different experimental settings (Figure 3). As Figure 3 (a), (b) and (c) show, compared with other approaches, our method achieves better quality on unseen poses or bodies by synthesizing more high-fidelity details like the clothes wrinkles and reconstructing the body shape more accurately. From Figure 3 (d), we demonstrate some rendering results on the unseen bodies of the THUman dataset after training on it. Our method generalizes well on the same THUman dataset and can synthesize accurate details.

In Figure 4 , we visualize the density volume from the density MLP in Section 3.2 as the mesh results of our 3D reconstruction. Different from previous methods that densely sample points within bounds of the geometry prior to determine the inside points through the density network for mesh construction, our progressive pipeline directly determines the sampling points from the geometry volume in Section 3.1, which contains much fewer redundant points and thus is more efficient for 3D reconstruction. Then we construct the mesh based on the points with higher density values. As Figure 4 (b) shows, on the unseen human bodies, previous image based 3D construction method like PIFuHD [28] can not generalize well. Besides their lower efficiency on making predictions for a lot of redundant sampling points, they are more likely to predict body parts that do not conform to a normal human body structure, because they can not integrate and adapt the given geometry information as well as we do. As shown in Figure 4, by integrating multi-view information to form a complete geometry volume adapting to the target human body, our method can generally reconstruct very close human body shape and even clothes details like folds on even unseen human bodies (Figure 4 (b)).



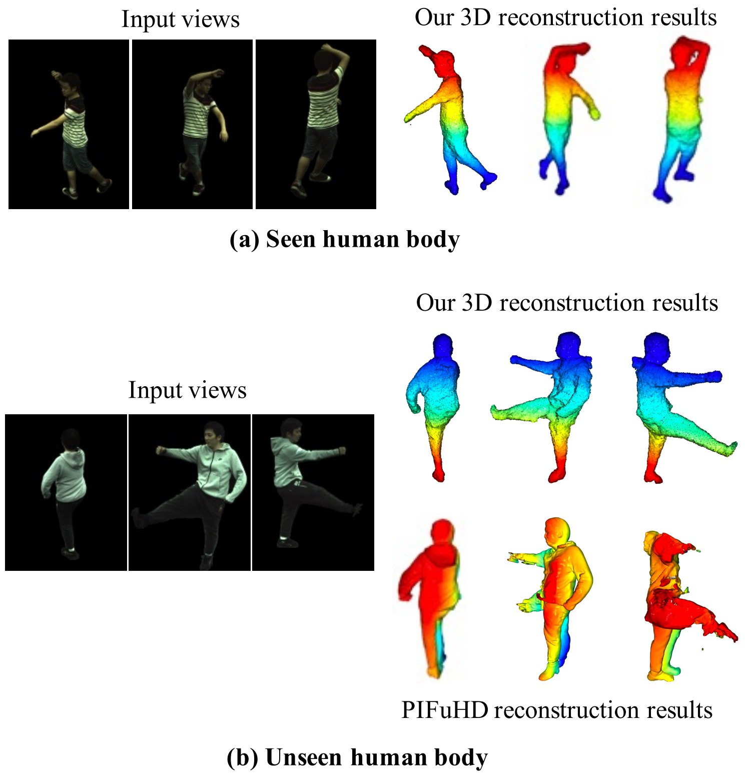
(a) Seen dataset, seen body, unseen pose

(b) Seen dataset, unseen body (human #1)

(c) Seen dataset, unseen body (human #2)

(d) Seen dataset, unseen body on THUman dataset
(for each image pair, GT in the left, our results in the right)

Fig. 3. Visualization comparisons on human rendering. Comparing to other methods, ours can synthesize more high-fidelity details like the clothes wrinkles and reconstructing the body shape more accurately. Our synthesis can stick to the normal human body geometry better than methods without geometry priors like NT and NHR. We can also recover more accurate lighting conditions than the previous video-based generalizable method NHP on unseen bodies (as (b) and (c)).


Input views
Our 3D reconstruction results
Our 3D reconstruction results
Input views
PIFuHD reconstruction results

(a) Seen human body

(b) Unseen human body

Fig. 4. Visualization of our 3D reconstruction results. The color in the mesh is only for clearer visualization. By integrating multi-view information to form a complete geometry volume adapting to the target human body, our method can compensate some limitations of SMPL (e.g., not accurate or lack cloth information), and can generally reconstruct very close human body shape and even clothes details like hoods and folds on unseen human bodies (as (b)). We can generalize better on the unseen human bodies than previous image based 3D construction method like PIFuHD, which predicts incomplete or redundant body parts in its reconstruction results (as (b)).

5 Conclusion

We propose a geometry-guided progressive NeRF model for generalizable and efficient free-viewpoint human rendering under sparse camera settings. Using our geometry-guided multi-view feature aggregation approach, the geometry prior can be effectively enhanced with the integrated multi-view information and form a complete geometry volume adapting to the target human body. The geometry feature volume combined with the detailed image-conditioned features can benefit the generalization performance on unseen scenarios. We also introduce a progressive rendering pipeline for higher efficiency, which reduces over 70% rendering time cost without performance degradation. Experimental results on two datasets verify our model can outperform previous methods significantly on generalization capacity and efficiency.

References

1. Carranza, J., Theobalt, C., Magnor, M.A., Seidel, H.P.: Free-viewpoint video of human actors. In: ACM Trans. on Graphics (2003) 4
2. Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., Sullivan, S.: High-quality streamable free-viewpoint video. In: ACM Trans. on Graphics (2015) 3
3. De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. In: ACM Transactions on Graphics (2008) 4
4. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques (2000) 3
5. Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S.R., Kowdle, A., Escolano, S.O., Rhemann, C., Kim, D., Taylor, J., et al.: Fusion4d: Real-time performance capture of challenging scenes. In: ACM Trans. on Graphics (2016) 3
6. Gall, J., Stoll, C., De Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: CVPR (2009) 4
7. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: CVPR (2018) 6
8. Guo, K., Lincoln, P., Davidson, P., Busch, J., Yu, X., Whalen, M., Harvey, G., Orts-Escolano, S., Pandey, R., Dourgarian, J., et al.: The relightables: Volumetric performance capture of humans with realistic relighting. In: ACM Trans. on Graphics (2019) 3
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICCV (2015) 8
10. Kwon, Y., Kim, D., Ceylan, D., Fuchs, H.: Neural human performer: Learning generalizable radiance fields for human performance rendering. In: NeurIPS (2021) 2, 3, 4, 7, 8, 9, 10
11. Li, R., Xiu, Y., Saito, S., Huang, Z., Olszewski, K., Li, H.: Monocular real-time volumetric performance capture. In: ECCV (2020) 4
12. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Lv, Z.: Neural 3d video synthesis. arXiv (2021) 1, 4
13. Liu, L., Gu, J., Lin, K.Z., Chua, T.S., Theobalt, C.: Neural sparse voxel fields. arXiv (2020) 4
14. Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: Neural free-view synthesis of human actors with pose control. ACM Trans. Graph. **40**(6) (dec 2021) 3
15. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In: CVPR (2020) 4
16. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. In: ACM Transactions on Graphics (2019) 9
17. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: ACM Trans. on Graphics (2015) 2, 4, 5
18. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020) 1, 4, 7, 8
19. Natsume, R., Saito, S., Huang, Z., Chen, W., Ma, C., Li, H., Morishima, S.: Siclope: Silhouette-based clothed people. In: CVPR (2019) 4
20. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: CVPR (2015) 3
21. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: CVPR (2020) 4

22. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Deformable neural radiance fields. arXiv (2020) 1, 4
23. Peng, S., Zhang, S., Xu, Z., Geng, C., Jiang, B., Bao, H., Zhou, X.: Animatable neural implicit surfaces for creating avatars from videos (2022) 3
24. Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: CVPR (2021) 2, 3, 4, 7, 8, 9, 10
25. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: CVPR (2021) 1, 4
26. Raj, A., Zollhoefer, M., Simon, T., Saragih, J., Saito, S., Hays, J., Lombardi, S.: Pva: Pixel-aligned volumetric avatars. arXiv (2021) 2, 4, 9
27. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: ICCV (2019) 4
28. Saito, S., Simon, T., Saragih, J., Joo, H.: Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In: CVPR (2020) 4, 12
29. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. arXiv (2019) 4
30. Stoll, C., Gall, J., De Aguiar, E., Thrun, S., Theobalt, C.: Video-based reconstruction of animatable human characters. In: ACM Trans. on Graphics (2010) 4
31. Su, Z., Xu, L., Zheng, Z., Yu, T., Liu, Y., Fang, L.: Robustfusion: Human volumetric capture with data-driven visual cues using a rgbd camera. In: ECCV (2020) 3
32. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. In: ACM Trans. on Graphics (2019) 9
33. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR (2021) 2, 4, 8
34. Wu, M., Wang, Y., Hu, Q., Yu, J.: Multi-view neural human rendering. In: CVPR (2020) 9, 10
35. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. arXiv (2020) 2, 4, 9
36. Yuan, W., Lv, Z., Schmidt, T., Lovegrove, S.: Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In: CVPR (2021) 1, 4
37. Zheng, Z., Yu, T., Wei, Y., Dai, Q., Liu, Y.: Deephuman: 3d human reconstruction from a single image. In: ICCV (2019) 4, 8
38. Zins, P., Xu, Y., Boyer, E., Wuhrer, S., Tung, T.: Data-driven 3d reconstruction of dressed humans from sparse views. In: 2021 International Conference on 3D Vision (3DV). pp. 494–504 (2021) 4