

HDhuman: High-quality Human Novel-view Rendering from Sparse Views

Tiansong Zhou[†], Jing Huang[†], Tao Yu, Ruizhi Shao, Kun Li*

Abstract—In this paper, we aim to address the challenge of novel view rendering of human performers that wear clothes with complex texture patterns using a sparse set of camera views. Although some recent works have achieved remarkable rendering quality on humans with relatively uniform textures using sparse views, the rendering quality remains limited when dealing with complex texture patterns as they are unable to recover the high-frequency geometry details that are observed in the input views. To this end, we propose HDhuman, which uses a human reconstruction network with a pixel-aligned spatial transformer and a rendering network with geometry-guided pixel-wise feature integration to achieve high-quality human reconstruction and rendering. The designed pixel-aligned spatial transformer calculates the correlations between the input views and generates human reconstruction results with high-frequency details. Based on the surface reconstruction results, the geometry-guided pixel-wise visibility reasoning provides guidance for multi-view feature integration, enabling the rendering network to render high-quality images at 2k resolution on novel views. Unlike previous neural rendering works that always need to train or fine-tune an independent network for a different scene, our method is a general framework that is able to generalize to novel subjects. Experiments show that our approach outperforms all the prior generic or specific methods on both synthetic data and real-world data. Source code and test data will be made publicly available for research purposes.

Index Terms—Image-based rendering, neural rendering, human reconstruction, transformer, visibility reasoning

1 INTRODUCTION

Realistic free-viewpoint rendering of human performers is in increasing demand with the development of AR/VR. Neural radiance fields (NeRF) [1] is the most promising way that can render photo-realistic images on novel views. However, NeRF needs a lot of input views to produce photo-realistic novel-view images. When the input views are highly sparse, the rendering quality of NeRF will degrade dramatically.

To render novel-view images of human performers from sparse views, neural body [2] integrates SMPL model [3] to the NeRF framework, in which they anchor a latent code on each vertex of SMPL model to integrate the observations over video frames. However, as the SMPL model does not contain any geometry details such as clothing folds, neural body is difficult to capture high-frequency details of human performers, making their rendering quality remain limited when dealing with human performers that wear loose clothes with complex texture patterns. Moreover, neural body [2] needs to train an independent network for each human and the training procedure is extremely time-consuming (at least 10 hours for each subject), which further limits its applications.

In this work, we propose HDhuman, a general method that is able to render high-quality human images at 2k



Fig. 1. High-quality reconstruction and rendering of a challenging human performer that wears long dress with complex texture patterns from only 6 input views, without any fine-tuning.

resolution using sparse views. The used camera views are no more than 8 views in a uniformly distributed circle arrangement. As shown in Fig. 1, we are able to handle challenging human subjects wearing long dresses with complex texture patterns. HDHuman uses a reconstruction-rendering pipeline to achieve the goal of high-quality novel-view rendering from sparse views, demonstrating that recovering geometry details observed in the input views significantly helps to improve the rendering quality. Firstly, We design a reconstruction network to reconstruct highly-detailed human models, recovering the geometry details of human performers, such as clothing folds. Then, based on the reconstruction results, we propose a rendering network to render high-quality images on novel-views as 2k resolution.

For human reconstruction from sparse views, the key is to fuse the geometry details that are observed in the

- [†] Equal contribution.
- * Corresponding author.
- Kun Li, Tiansong Zhou and Jing Huang are with Tianjin University. Email: lik@tju.edu.cn, tiansong97@tju.edu.cn, hj00@tju.edu.cn.
- Tao Yu and Ruizhi Shao are with Tsinghua University. Email: ytrock@126.com, shaorz20@mails.tsinghua.edu.cn.
- This work was done when Tiansong Zhou was an intern at Tsinghua University.

input multi-view images. Instead of using a naive averaging pooling operation for multi-view feature fusion in PIFu [4], we present a pixel-aligned spatial transformer in the reconstruction network. The proposed transformer is able to automatically calculate different fusion weights for each input view. By this way, we are able to preserve the geometry details such as clothing folds that are observed in the input images, resulting in highly-detailed reconstruction.

To achieve the goal of high-quality rendering from sparse views, the core is to solve the severe occlusion problems caused by the sparsity of input views. If we warp all the pixels of source views to the novel view, the occlusion will cause artifacts and blurring in the rendering results, as many pixels are invisible in the novel view. To solve this problem, we propose geometry-guided pixel-wise feature integration in the rendering network. Based on the reconstruction results from the reconstruction network, the proposed integration method will perform visibility reasoning and find the visible source views for each pixel of the novel view, enabling us to solve the occlusion problems and get high-quality rendering images.

Unlike the neural-network-based rendering works that need to train or fine-tune an independent network for each scene or human [1], [2], [5], [6], the proposed HDhuman is a general rendering framework. Benefit from the generalization ability of pixel-aligned spatial transformer and geometry-guided feature integration, our method is able to perform high-quality reconstruction and rendering for novel subjects without any fine-tuning. Experiments show that our general approach significantly improves the rendering quality compared with all the existing generic and specific works.

Overall, thanks to the pixel-aligned spatial transformer for highly-detailed human reconstruction and the geometry-guided pixel-wise feature integration for solving the occlusion problems, we are able to achieve the goal of high-quality human novel-view rendering from sparse views. In summary, our main contributions are:

- We propose a new general neural rendering framework capable of rendering novel views of human performers with complex texture patterns at 2k image resolution from a sparse set of camera views.
- A pixel-aligned spatial transformer that is able to perform efficient multi-view feature fusion in human reconstruction, enabling us to reconstruct highly detailed human models from sparse views.
- A geometry-guided pixel-wise feature integration method for efficiently solving the severe occlusion problems that caused by the sparsity of input views.
- We demonstrate significant rendering quality improvements of our method compared to the prior works, especially for the humans with loose clothes or complex texture patterns.

2 RELATED WORK

2.1 Human Reconstruction

Human reconstruction has a long and overlapping history in both computer vision and graphics. In earlier years, multi-view stereo (MVS) is the most widely used method

for human reconstruction. Liu *et al.* [7] proposes a continuous depth map estimation method for improving the performances of MVS.

Over the past decade, benefit from the development of hardware, many works [8], [9], [10], [11] take single or multi RGBD views as input to get amazing real-time high fidelity human reconstruction and rendering. However, the relied depth sensors are only able to capture humans or objects that are near the sensors. When humans move far from the sensors (farther than 3m), the depth sensors will fail capturing depth streams, which greatly limits their applications.

In recent years, human reconstruction from only a single RGB camera [4], [12], [13], [14], [15], [16], [17], [18] draws great research attention. The most promising method for single-view human reconstruction is pixel-aligned implicit function (PIFu) [4]. PIFu takes a single color image as input and encode the image to generate pixel-aligned image features. Based on the pixel-aligned features that contain the geometry details in the input image, they learn an implicit function over the 3D space. Following PIFu, PIFuHD [15] introduces a multi-level framework for high fidelity 3D reconstruction of clothed human. It uses an additional image translation network to predict normal maps from original color images. The predicted normal maps enable it to reconstruct more geometry details, such as clothing folds. Moreover, a fine-level is used in PIFuHD to recover more subtle details from the high resolution input images. Video-based methods [16], [17], [19] reconstruct dense, space-time coherent deforming geometry of dynamic humans from monocular videos, but they need a pre-captured template mesh.

For multi-view human reconstruction, PIFu uses a naive average pooling operation for multi-view feature fusion, which is not efficient enough to fuse the geometry details that are observed in the multi-view input images. Zins *et al.* [20] use an attention-based fusion method to automatically weight the input views for multi-view human reconstruction. But they use an additional local 3D context encoding layer to make the network aware of the local 3D context around each querying point. This layer makes the reconstruction more robust but it increases the computation and memory consumption of the networks. In our observation, we found that directly predicting the SDF value of each point without any "local information" is also able to produce high-quality reconstruction results, helping us to strike a good balance between reconstruction quality and computation consumption. Zheng *et al.* [21] use SMPL as the geometry prior to solve the occlusion problem in multi-person reconstruction, but the usage of SMPL not only makes the reconstruction results might be pixel-misaligned, but also makes it unable to reconstruct the human cases with loose clothes, such as long dress. In contrast, benefit from the proposed pixel-aligned spatial transformer, we are able to produce highly detailed reconstruction results without any human template prior such as SMPL, and we are able to handle human performers that wear loose clothes with complex texture patterns.

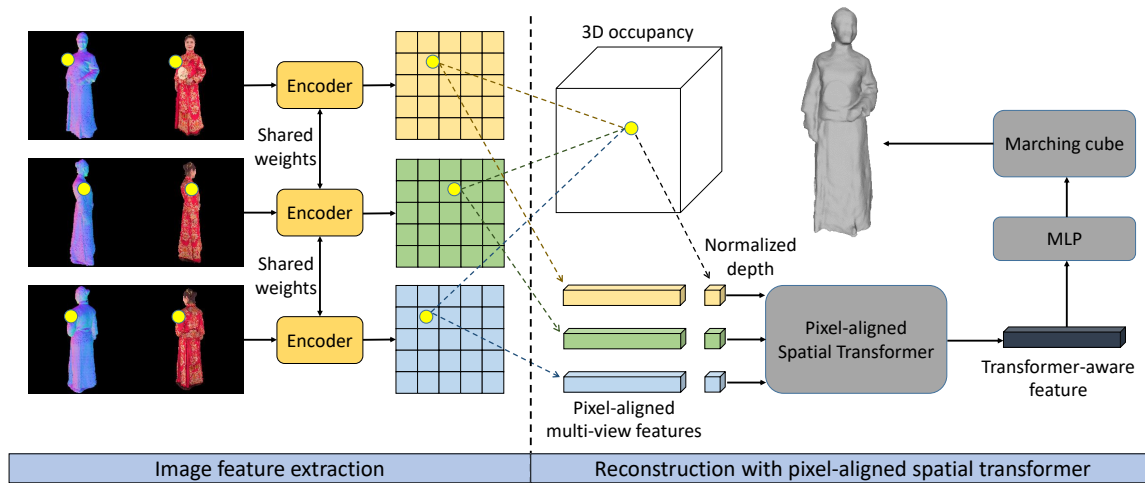


Fig. 2. **Architecture of our reconstruction network.** For memory efficiency, we only use a single level for reconstruction. The pixel-aligned spatial transformer enables us to preserve the geometry details that are observed in the input views, resulting in highly detailed reconstruction. Our framework doesn't use any geometry prior, and hence our reconstruction results are fully pixel-aligned.

2.2 Human Novel-view Rendering

Many works [22], [23], [24] use the traditional image-based rendering or view synthesis pipeline for novel view rendering, *i.e.*, proxy geometry reconstruction, warping and blending. To bridge the gap between reconstruction and rendering, Liu *et al.* [25] presents a point-cloud-based method to improve multi-view stereo algorithm for more realistic free view synthesis (FVV). Benefit from the significant research progress of deep learning in recent years, some works [22], [26] use neural networks to blend the novel views from the warped source views. However, these warping-based works need dense camera views as input to warp the source views to the novel view. If the input camera views are highly sparse, the performances of this classic method will degrade dramatically as the wide camera baselines will cause severe occlusion problems. To tackle this problem, Suo *et al.* [27] proposes a neural blending scheme with neural geometry prior for novel-view rendering of human performers.

More recently, volume rendering is another method to achieve photo-realistic rendering. It uses volumes to represent the shape and appearance of a scene. Multi-plane images (MPIs) [28] is one of the most promising ways for this volumetric representation, which computes a separate $RGB\alpha$ image for each depth plane. The α channel and the RGB channels represent the shape and appearance of the scene respectively. Based on the MPIs-representation, Soft3D [29] introduces a soft 3D representation of the scene geometry. This representation enables it to handle the depth uncertainty of the scene. Benefit from the significant research progress on neural networks, Flynn *et al.* and Mildenhall *et al.* [30], [31] use neural networks to predict the MPIs for each input views.

Another promising research direction for volume rendering is neural radiance fields (NeRF) [1], which uses a multi-layer perceptron (MLP) to represent the density and color fields of a scene and thus it is well-suited for differentiable rendering. Based on NeRF, Liu *et al.* [32] propose neural sparse voxel structure for faster training and inference. However, both MPIs-based and NeRF-based [1], [32] works

need dense camera views as input and always need to train or fine-tune an independent network of each scene.

For human rendering from sparse views, some works [33], [34], [35], [36], [37] use SMPL template as a prior, which helps to constrain the motion space and improve the rendering quality. However, as SMPL model does not contain any geometry details such as clothing folds, it is difficult for them to capture high-frequency details of human performers, which limits their rendering quality when dealing with human performers with complex texture patterns. In contrast, the proposed pixel-aligned spatial transformer in our work enables us to capture the geometry details that are observed in the input views, which helps improve the rendering quality on the areas with complex geometry patterns, such as loose clothes or clothing folds. Another line of works [38], [39] use a deformation field to render human performers from single or multi-view videos, in which they use a MLP to map all the frames of the video to a canonical space and perform free-view rendering using NeRF on the canonical space. However, as the capacity of a deformation field (MLP) is restricted, the deformation field is only able to handle a limited number of frames. When dealing with a long sequence with complex human motions, the deformation field will be “overwhelmed”, resulting in rendering failures. Instead of using a deformation field, we treat each frame as an independent subject to process dynamic cases, enabling us to handle videos with arbitrary length. Experiments show that we are able to produce spatial-temporal coherent rendering results naturally when dealing with dynamic human performers.

3 APPROACH

To achieve the goal of rendering high-quality human novel view images, we firstly propose a reconstruction network to reconstruct highly-detailed human models from sparse views (Sec. 3.1.2). Based on the reconstruction results, we propose our rendering network to render the high-quality images at 2k resolution on novel views (Sec. 3.2).

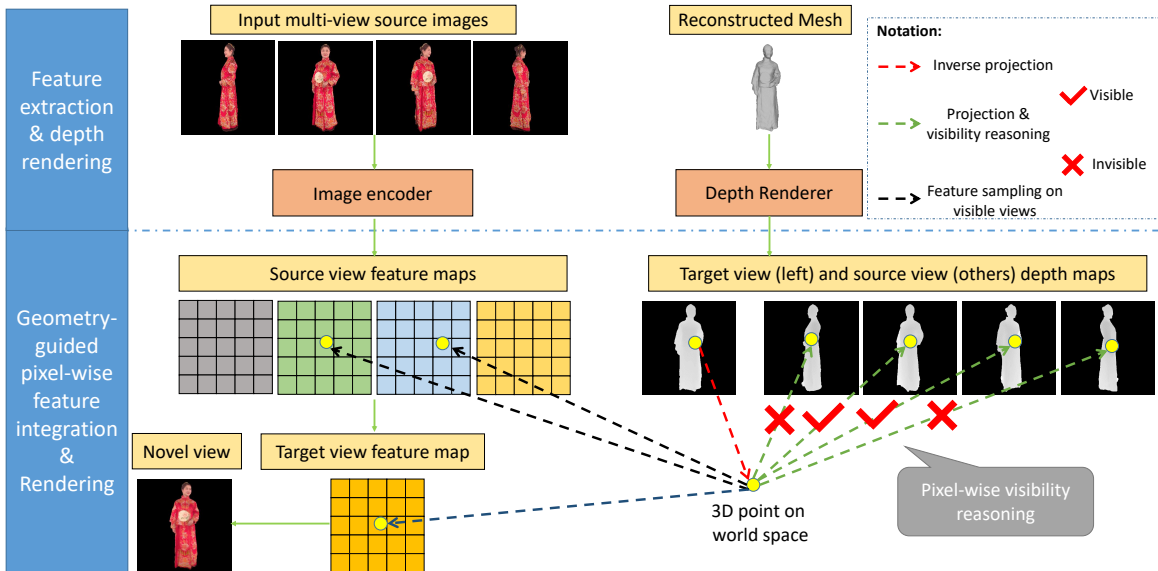


Fig. 3. **Architecture of our rendering network.** The geometry-guided pixel-wise feature integration enables to solve the severe occlusion problems caused by the sparsity of input views, resulting in high-quality novel view rendering.

3.1 Human Reconstruction

3.1.1 Preliminary

PIFu [4] introduces a pixel-aligned implicit function to reconstruct the underlying 3D human geometry from a single or multi-view images. The proposed pixel-aligned implicit function consists of a convolutional image encoder g and a continuous implicit function f that is represented by a multi-layer perceptron. Concretely, the human surface is defined as

$$f(g(I(x)), z(X)) = s : s \in \mathbb{R}, \quad (1)$$

where X is the given 3D point, $x = \pi(X)$ is its orthogonal 2D projection, $g(I(x))$ is the bilinear sampled image feature at x . During the inference, 3D space is uniformly sampled to infer the occupancy and the final iso-surface is extracted with a threshold of 0.5 using marching cube algorithm [40] (where the occupancy value s smaller than 0.5 is inside and s greater than 0.5 is outside).

To achieve high-fidelity 3D reconstruction of clothed human from a single image, recovering detailed information such as clothing folds, PIFuHD [15] proposes a coarse-to-fine framework for 3D clothed human reconstruction using images with resolution of 1024×1024 . It uses an additional network to predict normal maps from the input color images. Compared with PIFu, the predicted normal maps enable it to recover more details as the geometry information in normal maps is more explicit. Moreover, a fine-level is used in PIFuHD to recover details from high resolution images. Specifically, the fine level can be denoted as

$$f^H(g^H(I_H, F_H, B_H, x_H), \Omega(X)) = s : s \in \mathbb{R}, \quad (2)$$

where I_H, F_H, B_H are the input color images and the predicted front and back normal maps at resolution of 1024×1024 . $\Omega(X)$ is the feature extracted from the coarse level.

For multi-view reconstruction, PIFu [4] uses a naive average pooling operation to fuse multi-view features. The

TABLE 1
Quantitative reconstruction results on Twindom dataset. For each model, we use 6 views for reconstruction. Our reconstruction results outperform neural body by a large margin.

	P2S		Chamfer	
	Ours	Neural body [2]	Ours	Neural body [2]
Model1	0.1866	0.9299	0.1968	0.7030
Model2	0.2604	0.8902	0.3817	0.6753
Model3	0.1718	0.6639	0.1840	0.5410
Model4	0.3858	1.8421	0.4070	1.5791
Model5	0.1660	0.6104	0.1794	0.5164
Average	0.2698	0.8030	0.2341	0.9873

average pooling operation in PIFu is not efficient enough to aggregate geometry details that are observed in the input images, as it regards each view equally, resulting in losing subtle geometry details and limiting the rendering quality. Therefore, we need to find a way that is able to calculate the correlations between the input views, providing high-level information for feature aggregation and enabling us to preserve the subtle details.

3.1.2 Pixel-aligned Spatial Transformer

To solve the above limitations of the prior works, we propose a pixel-aligned spatial transformer in the reconstruction network. See Fig. 2 for an illustration.

Following PIFuHD [15], we firstly predict frontal normal maps from color images, and encode each view's color image and normal map using a Hourglass encoder g . Our observation is that in multi-view reconstruction, the coarse level network in PIFuHD [15] is enough to reconstruct high-frequency details with frontal normal maps as input. Therefore, for memory efficiency, we only use a single level for reconstruction and the input images of reconstruction network are at a resolution of 512×512 .

To efficiently preserve the geometry details that are observed in the input views, we use a spatial transformer to calculate the correlations between the input multi-view

TABLE 2

Quantitative rendering results on Twindom dataset. "NB" means neural body. "D+S3D" means Soft3D with our depth maps as input. "NT" means neural texture. "SA-NeRF" means Surface-Aligned NeRF. For each model, we use 6 views as input and 30 views for evaluation. \uparrow means higher is better and \downarrow means lower is better.

	LPIPS \downarrow					SSIM \uparrow					PSNR \uparrow				
	Ours	NB	D+S3D	NT	SA-NeRF	Ours	NB	D+S3D	NT	SA-NeRF	Ours	NB	D+S3D	NT	SA-NeRF
Model1	0.146	0.266	0.180	0.295	0.294	0.876	0.842	0.831	0.790	0.795	25.32	24.63	23.84	22.96	20.62
Model2	0.196	0.290	0.252	0.391	0.360	0.782	0.731	0.701	0.612	0.633	17.53	17.81	16.26	11.23	15.89
Model3	0.122	0.213	0.179	0.320	0.289	0.865	0.832	0.793	0.712	0.732	21.75	21.81	19.49	13.37	13.23
Model4	0.230	0.425	0.265	0.438	0.475	0.610	0.469	0.560	0.501	0.417	18.53	14.35	17.52	15.16	13.23
Model5	0.130	0.274	0.172	0.304	0.320	0.794	0.745	0.739	0.696	0.684	21.76	21.33	20.30	17.66	18.21
Average	0.1647	0.2931	0.2096	0.3493	0.348	0.785	0.724	0.725	0.662	0.652	20.98	20.03	19.48	16.08	16.48

TABLE 3

Quantitative rendering results on our multi-view video dataset. For each frame, we use 8 views as input and 16 views for evaluation. Each sequence has a length of 150 frames.

	LPIPS \downarrow					SSIM \uparrow					PSNR \uparrow				
	Ours	NB	D+S3D	NT	SA-NeRF	Ours	NB	D+S3D	NT	SA-NeRF	Ours	NB	D+S3D	NT	SA-NeRF
Sequence1	0.122	0.278	0.177	0.267	0.281	0.823	0.732	0.722	0.687	0.720	24.14	21.05	21.11	17.61	19.50
Sequence2	0.162	0.350	0.206	0.329	0.354	0.787	0.659	0.671	0.649	0.650	22.36	20.60	19.13	17.11	19.51
Sequence3	0.119	0.236	0.1723	0.234	0.243	0.887	0.831	0.811	0.790	0.815	25.04	22.76	22.10	19.81	21.14
Sequence4	0.129	0.247	0.186	0.245	0.251	0.872	0.832	0.797	0.796	0.815	26.90	25.35	24.42	22.88	23.75
Sequence5	0.116	0.243	0.174	0.259	0.246	0.880	0.822	0.800	0.751	0.809	26.47	24.32	23.45	17.59	22.11
Average	0.129	0.271	0.183	0.267	0.275	0.850	0.775	0.760	0.735	0.761	24.98	22.82	22.04	19.00	21.20

pixel-aligned features. Specifically, for a 3D point X , given N views encoded features, we stack them together to get $\Phi_{mv} \in \mathbb{R}^{N \times D}$, where D is the feature dimensions. Then, we embed it with three learnable different linear layers: $\Phi_q = \Phi_{mv} W_q, \Phi_k = \Phi_{mv} W_k, \Phi_v = \Phi_{mv} W_v$, where $W_q, W_k, W_v \in \mathbb{R}^{D \times d_k}$ and $\Phi_q, \Phi_k, \Phi_v \in \mathbb{R}^{N \times d_k}$. d_k is the embedded feature dimension. After that, a spatial transformer will be applied:

$$\begin{aligned} \Phi_{mv_att} &= Transformer(\Phi_q, \Phi_k, \Phi_v) \\ &= softmax\left(\frac{\Phi_q \Phi_k^T}{\sqrt{d_k}}\right) \Phi_v, \end{aligned} \quad (3)$$

where $\Phi_{mv_att} \in \mathbb{R}^{d_k}$ is the transformer-aware multi-view features. To counter the gradient vanishing problem causing by the softmax operation, we scale the dot-product operation by $\frac{1}{\sqrt{d_k}}$. Different from the feature before transformer calculation, which only contains the geometry details that are observed in a single view, the transformer-aware feature contains all the geometry details that are observed in multi-view images. Compared with the average pooling operation in PIFu [4], the transformer-based fusion operation calculates the correlations between the multi-view features, which provides a high-level information for fusion, allowing the network to preserve more geometry details.

To query the depth value of point X with the encoded features, we need to normalize the depth value for each view. Given multi-view human images and the corresponding calibrated camera parameters as input, we will estimate each view's human 2D skeleton using Openpose [41], and then obtain 3D skeleton $S_w \in \mathbb{R}^{3 \times J}$ in world space through triangulation. J is the number of joints. We use the hip position $H_w \in \mathbb{R}^3$ and neck position $N_w \in \mathbb{R}^3$ to normalize the depth. Specifically,

$$H_c^i = R_i H_w + t_i = (H_{c_x}^i, H_{c_y}^i, H_{c_z}^i), \quad (4)$$

$$X_c^i = R_i X + t_i = (X_{c_x}^i, X_{c_y}^i, X_{c_z}^i), \quad (5)$$

$$z_i(X_{c_z}^i) = \frac{X_{c_z}^i - H_{c_z}^i}{\lambda \|H_w - N_w\|_2}, \quad (6)$$

where H_c^i, X_c^i are the hip position and the point X position in view i 's camera space respectively. R_i, t_i are camera i 's rotation and translation. λ is a constant value to make sure that the normalized depth always lies in $(0, 1)$, and we set it to $4\sqrt{3}$ in all our experiments.

Finally, we use a multi-layer perceptron to predict the 3D occupancy and use marching cube algorithm to extract the surface from the predicted occupancy, resulting in the final reconstructed mesh.

3.2 Novel View Rendering

To achieve the goal of high-quality novel view rendering from sparse views, the core problem is to solve the occlusion problems caused by the sparsity of input views. Based on the highly detailed reconstructed results from the reconstruction network, we propose geometry-guided pixel-wise feature integration to perform efficient visibility reasoning for solving the occlusion problems.

Concretely, given a sparse set of input images $\{I_0, \dots, I_N\}$, the corresponding calibrated camera parameters and the reconstructed meshes, we use a rendering network to render high-quality novel view images of human performers. The architecture of rendering network is shown in Fig. 3. In summary, we use an encode-integration-render framework for novel view rendering, which mainly consists of three parts: 1) encoding each input image to feature space using an encoder-net e , 2) pixel-wise visibility reasoning and feature integration using the reconstructed human geometry, and 3) rendering the novel view image using a render-net r .

Image encoding. For each input image $I_n \in \mathbb{R}^{H \times W \times 3}$, we firstly encode it to a high-dimensional feature space and get feature map $F_n = e(I_n) \in \mathbb{R}^{H \times W \times D}$ at the same resolution.

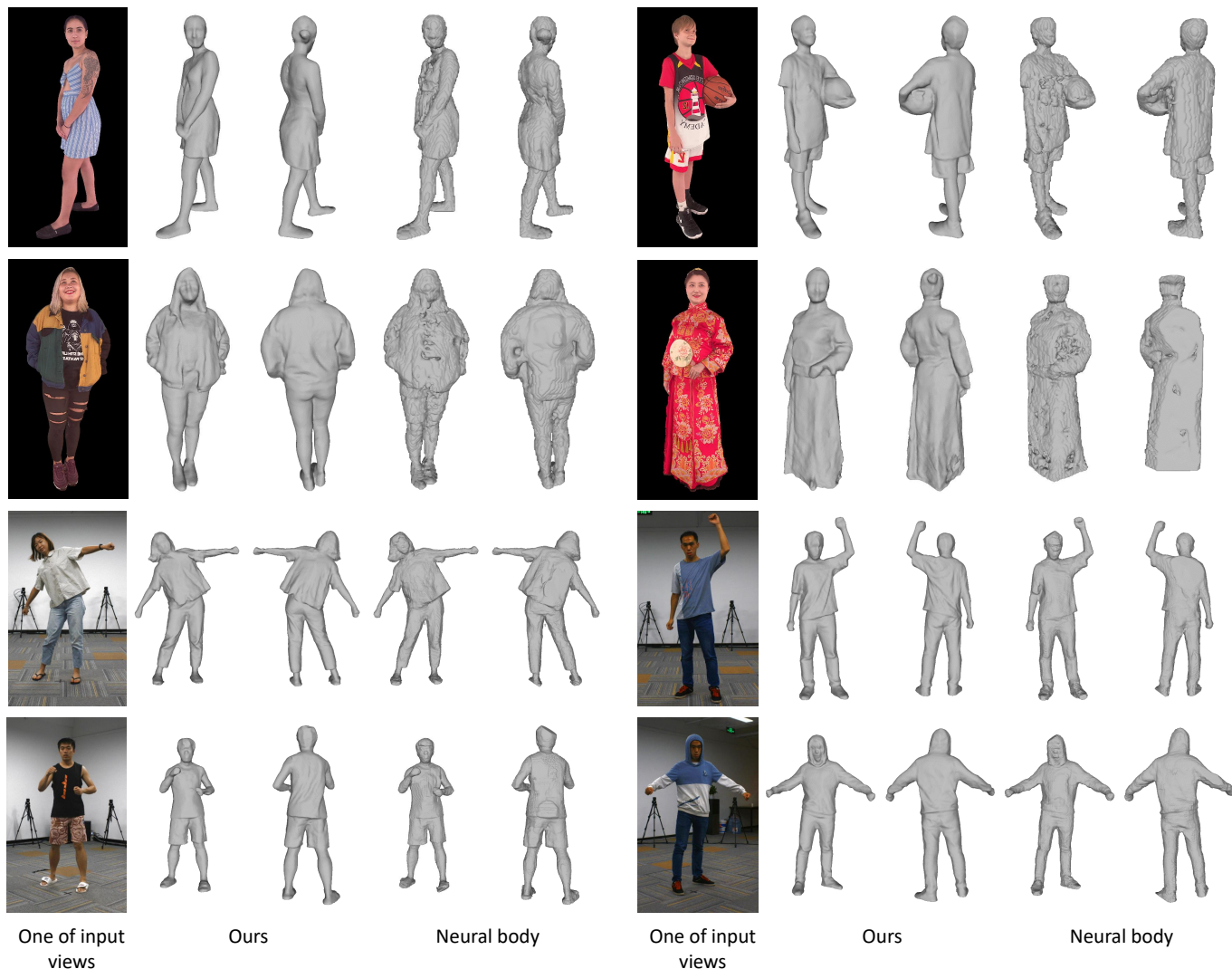


Fig. 4. **Qualitative results of reconstruction.** The top two rows are the results on twindom dataset, with 6 views as input. The bottom two rows are the results on real-world data, with 8 views as input. We strongly recommend the readers to zoom in the pages for a better visualization of the reconstructed details.

Following Riegler *et al.* [42], the encoder e uses a Res-UNet architecture, where the encoding part is an ImageNet-pretrained ResNet [43] and the decoding part upsamples the feature map using nearest-neighbor interpolation, concatenating it with the corresponding feature map (of the same resolution) from the encoding part.

Geometry-guided pixel-wise feature integration. Traditional image-based rendering methods always use warping operation to map the source views to the novel view, in which the warping operation will warp the pixels or areas that are not visible in the novel view, resulting in rendering artifacts or blurring. Therefore, we argue that this warping operation is not suitable for solving the occlusion problems and propose our geometry-guided pixel-wise feature integration. For each pixel in a novel view, the visible input views are different. Therefore, we perform visibility reasoning and get the visible source views for each pixel in the novel view. Then, we only integrate the features from visible source views to the novel view pixel, the source views that are not visible will be abandoned.

Concretely, given the calibrated camera parameters (intrinsic and extrinsic parameters) of input views $\{C_1, \dots, C_N\}$, the novel view camera C_{novel} and the reconstructed human mesh, we will render depth maps of the input views $\{D_1, \dots, D_N\}$ and novel views D_{novel} using OpenGL or Taichi. Then, for each pixel $p \in \mathbb{R}^2$ with valid depth value ($d_{render} > 0$) in the novel view, we will unproject it to the world space, getting point $P \in \mathbb{R}^3$, and then project it to the image space of each input view using the corresponding cameras, getting the reprojected pixel, reprojected depth and rendered depth in each view $\{(preproj_1, d_{reproj_1}, d_{render_1}), \dots, (preproj_N, d_{reproj_N}, d_{render_N})\}$:

$$(preproj_n, d_{reproj_n}, d_{render_n}) = C_n(C_{novel}^{-1}(p, d_{render})), \quad (7)$$

where $d_{render_n} = D_n(preproj_n)$. If the difference between the rendered depth and the reprojected depth is lower than a threshold, we will regard view n to be visible for pixel p :

$$|d_{render_n} - d_{reproj_n}| < \lambda \cdot \min(d_{render_n}, d_{reproj_n}), \quad (8)$$

where λ is a hyper-parameter and we set it to 0.01 in all our experiments.



Fig. 5. **Novel view rendering on Twindom dataset.** We use 6 views as input. Results show that our method significantly outperforms all other methods qualitatively, especially in the areas that have complex details.

After the visibility reasoning, we will sample features from all the visible views using bilinear sampling. *i.e.*, $f_n = F_n(\text{Preproj}_n)$, and getting $\{f_1, \dots, f_K\}$, where K is the number of visible source views for pixel p . Then, a direction average operation will be performed to integrate source view features to the novel view pixel:

$$f_{fusion} = \frac{1}{W} \sum_{k=1}^K \max(0, \cos(\langle dir_{novel}, dir_k \rangle)) \cdot f_k, \quad (9)$$

where dir_{novel}, dir_k are the novel view direction and the visible source view k 's direction respectively. The fused novel view feature is divided by $W = \sum_{k=1}^K \max(0, \cos(\langle dir_{novel}, dir_k \rangle))$ for normalization.

Novel view rendering. After getting the feature map F_{novel} of the novel view, we will use a convolutional render-net r to render the novel view's color image *i.e.*, $I_{novel} = r(F_{novel})$.

4 EXPERIMENTS

4.1 Training

Dataset. We collect 1700 high-quality textured human meshes from Twindom [44] as a large-scale dataset for the training and evaluation of our reconstruction network and rendering network. The collected models have a wide range

of clothing, poses and shapes. We randomly split the models into a training set of 1500 subjects and a testing set of 200 subjects.

Reconstruction network training. For each subject in the dataset, we generate 360 virtual perspective cameras in yaw axis (1 camera for each degree) and each camera has a random pitch angle. Then, we render 360 images at a resolution of 512×512 using Taichi. During training, we randomly pick 4 views over the 360 images as the input for each iteration. We use Adam [45] optimizer with a learning rate of 1×10^{-4} and train the reconstruction network for nearly 300000 iterations for convergence. The training procedure costs nearly 3 days with a batch size of 1 in a single Nvidia TitanXp GPU.

Rendering network training. The rendering network needs reconstructed meshes as input, but we didn't use the ground-truth geometry. Instead, we randomly pick 5 views over the 360 images and use the reconstruction network to generate a coarse mesh for each subject. With the coarse mesh as input, the rendering network will have greater generalization ability and have more robust performances. We use the generated perspective cameras to render 360 images and depth maps at a higher resolution of 1024×1024 for

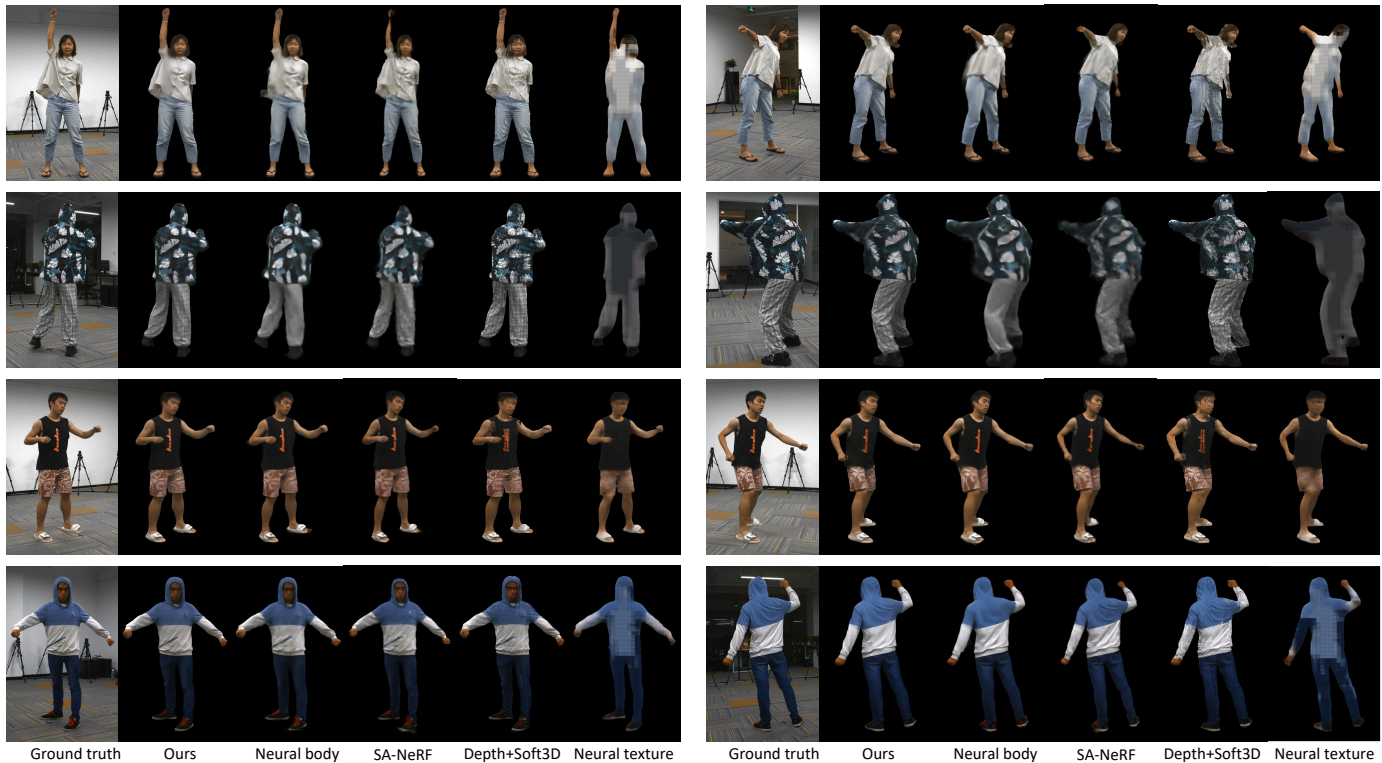


Fig. 6. **Novel view rendering on our captured multi-view videos dataset.** We use 8 views as input. We strongly recommend the readers to zoom in the pages for a better visualization of the rendered details.

TABLE 4
Quantitative evaluation for ablation study on Twindom dataset.

	Lpips ↓	SSIM ↑	PSNR ↑
Full model	0.1647	0.7853	20.98
Average pooling	0.1880	0.7610	19.82
No visibility reasoning	0.1865	0.7720	20.46

each subject in our dataset. We divide the yaw axis degrees into 6 parts, each parts has a range of 60 degrees. During training, we randomly pick one view in each part, resulting in 6 images as input. Then, we randomly select one view as the novel view and the other views are the source views. We use Adam optimizer with a learning rate of 1×10^{-4} and train for nearly 400000 iterations for convergence. The training procedure costs nearly 2 days with a batch size of 1 using a single Nvidia RTX3090 GPU.

4.2 Evaluation Settings

In this section, we will introduce our evaluation metrics, compared methods and the settings for each compared method. The evaluation will be performed both on static human performers (synthetic data) and dynamic human performers (real-world data). We highly recommend readers to refer to the supplemental video for a better visualization of the reconstruction and rendering results.

Metrics. For reconstruction, we use Chamfer distance and Point-to-Surface distance (P2S) for quantitative evaluation. For rendering, we quantitatively evaluate our method using

three most widely-used metrics: peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) and learned perceptual image patch similarity (LPIPS).

Compared methods. As our final goal is human novel-view rendering, we compare our method with both generic and specific rendering methods [2], [6], [29], [35]: **1)** Neural texture [6] introduces differentiable latent neural texture maps for novel view rendering. It needs to train an independent network for a different subject. Note that, to evaluate neural texture on dynamic humans, topologically consistent reconstruction of human performers is necessary. Therefore, we simply track the reconstructed results and get a topologically consistent mesh sequence for each dynamic human; **2)** Soft3D [29] is an MPI-based method for view synthesis, which uses multi-plane images (MPI) to handle the uncertainty of depth maps. It is a generic view synthesis method that doesn't need to train any network. Note that Soft3D doesn't use any geometry prior, and hence it needs to use MPI to estimate an initial depth map for each input image, which is extremely coarse, making the quality of synthetic images very low. For fairness, we use the rendered depth maps in our method as the initial depth maps for an additional input of Soft3D, which improves its performance; **3)** Neural body [2] proposes a NeRF-based implicit neural representation for human novel-view rendering, which needs to train a separate network for each human. Note that reconstruction results are only available on neural body, and hence we will perform comparisons with it for reconstruction evaluation. **4)** Surface-Aligned NeRF [35] builds neural radiance fields on the surfaces of SMPL meshes, in

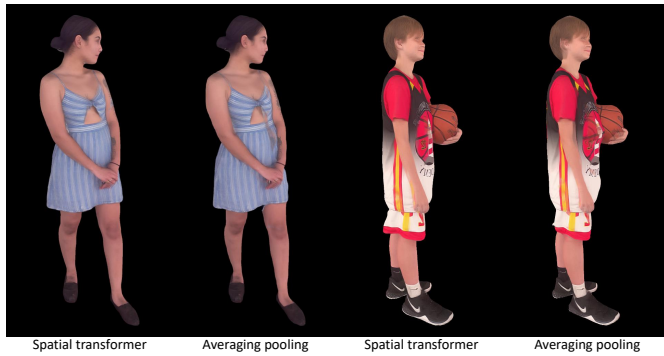


Fig. 7. **Comparisons with average pooling operation.** Compared with average pooling operation for multi-view feature fusion that introduced in PIFu, our proposed spatial transformer is able to generate higher-quality results.

which it injectively maps a spatial point to a surface-aligned representation that consists of a projected surface point and a signed height to the mesh surface.

4.3 Results on Synthetic Data

We select 5 models in the test set of Twindom for the evaluation on synthetic data. For each model, we render 36 images in a circle arrangement at a resolution of 2048×2048 . We use 6 uniformly distributed views as input and the remaining 30 views for evaluation. Table 1 shows the quantitative reconstruction comparisons of our method with neural body [2]. For both P2S distance and Chamfer distance, our method outperforms neural body by extremely large margins. Some qualitative reconstruction results are shown in the top two rows of Fig. 4. The reconstruction results are produced with only 6 views as input, showing that our method is able to perform highly detailed reconstruction of challenging humans using only a sparse set of camera views.

For novel view rendering, some qualitative results are illustrated on Fig. 5, showing that our method is able to render high-quality images on challenging humans with complex texture patterns or loose clothes, such as long dresses.

4.4 Results on Real-World Data

To evaluate our method's performance on real-world data, we create a multi-view dataset, which captures 5 dynamic human videos at a resolution of 2660×2300 using a multi-camera system that has 24 calibrated synchronized cameras in a circle arrangement. In contrast to the dataset of neural body [2], which only captures humans with relatively uniform textures, we capture human performers with complex texture patterns or loose clothes. We select 8 uniformly distributed cameras as the source input views and use the remaining 16 camera views for testing. All sequences have a length of 150 frames.

Some qualitative results are shown in the bottom two rows of Fig. 4. Compared with neural body, our reconstruction results have more high-frequency details, such as clothing folds. Achieving great performances on real-world data means that the proposed pixel-aligned spatial transformer is robust and efficient.



Fig. 8. **Impact of pixel-wise visibility reasoning.** If we warp all the source view pixels to the target view, the rendering results will suffer from severe occlusion problems caused by the wide camera baselines, resulting in blurring and artifacts.

The quantitative results are shown in Table 3. Our method outperforms all other methods on all the measured metrics. Fig. 6 illustrates some qualitative results. The novel views rendered by our method contain more texture details compared with other methods. Benefiting from the geometry-guided pixel-wise feature integration, we efficiently solve the occlusion problems caused by the sparsity of input views, resulting in the robust high-quality rendering on real-world data.

4.5 Ablation Studies

We compare two main different factors that contribute to our method: pixel-aligned spatial transformer and pixel-wise visibility reasoning.

Impact of pixel-aligned spatial transformer. For comparison, we train a reconstruction network using average pooling operation for multi-view feature fusion. Then, we use the reconstruction results generated by average pooling operation and our proposed transformer as the input of rendering network separately. Note that the reconstruction network with average pooling operation can be regraded as a multi-view PIFuHD method. Quantitative evaluation results on Twindom dataset are illustrated in the second and third rows of Table 4, showing that spatial transformer used in our method outperforms average pooling operation on all the three metrics. Fig. 7 shows that, benefiting from the inner-product operation, our proposed spatial transformer is able to calculate the correlations between input views, resulting in rendering results with higher quality compared with average pooling operation.

Impact of pixel-wise visibility reasoning. We design a pixel-wise visibility reasoning module to solve the occlusion problems caused by the sparsity of input views. To figure out the strength of this module, we evaluate our method's performance without visibility reasoning. Quantitative comparisons on Twindom dataset are shown in the second and fourth rows of Table 4, showing that without visibility reasoning, rendering results will degrade on all the three metrics. Qualitative comparisons are illustrated in Fig. 8, showing that without pixel-wise visibility reasoning, occluded pixels in the source views will be warped to the target view, resulting in rendering artifacts and blurring.

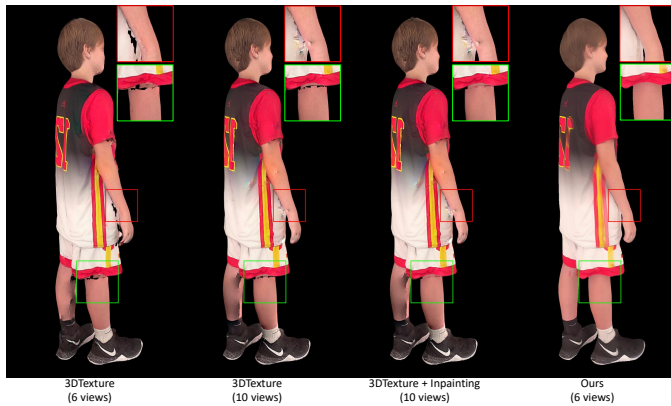


Fig. 9. **Our rendering network vs. 3DTexture [46].** Compared with 3DTexture that directly “colorizes” the reconstructed mesh, our rendering network is able to render the occluded areas that are invisible in the input views.

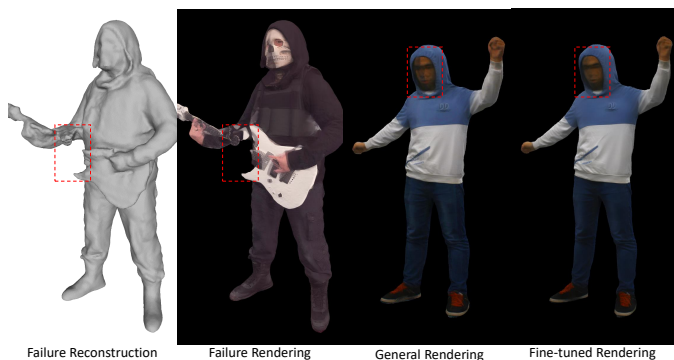


Fig. 10. **Failure cases of our proposed method.** The first and second columns show that failure reconstruction results will cause rendering artifacts. The third and fourth columns show that fine-tuning is not an efficient way to improve the rendering quality as we are unable to refine the geometry in the process of fine-tuning.

Impact of the number of input views. The quantitative evaluation on Twindom dataset with different number of input views is illustrated in Table 5, showing that increasing the number of input views only has slight improvement on the rendering results. Therefore, we can draw a conclusion that our method works well when the input views are highly sparse.

Rendering network vs. texture mapping. To compare with the traditional texture mapping methods and figure out why the rendering network is needed, we use the state-of-the-art texture mapping method 3DTexture [46] to “colorize” the reconstructed mesh and use OpenGL to render novel-view images. Qualitative comparison is illustrated in Fig. 9, showing that 3DTexture fails to render the occluded areas that are invisible in the input views, resulting in “black holes” in the invisible areas. We also compare an inpainted version for 3DTexture by using an inpainting technique [47] to fill the holes. It can be seen that the hole filling cannot deal with all the artifact problems, while our method is able to produce more reasonable rendering results. Our rendering network can be regarded as a texture prior, which helps to complete these invisible areas.

Quantitative evaluation on Twindom dataset is reported

TABLE 5
Quantitative evaluation of different number of input views.

	SSIM \uparrow			PSNR \uparrow		
	6 view	8 view	10 view	6 view	8 view	10 view
Subject1	0.876	0.884	0.885	25.32	25.58	25.63
Subject2	0.782	0.794	0.796	17.53	17.77	17.82
Subject3	0.865	0.873	0.874	21.75	21.97	22.03
Subject4	0.610	0.626	0.625	18.53	18.70	18.70
Subject5	0.794	0.804	0.803	21.76	21.90	21.88
Average	0.785	0.796	0.797	20.98	21.18	21.21

TABLE 6
Quantitative comparisons with 3DTexture [46].

	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
Ours (6 views)	20.78	0.856	0.141
3DTexture (6 views)	19.60	0.822	0.160
3DTexture + Inpainting (6 views)	19.70	0.824	0.158
3DTexture (10 views)	19.77	0.829	0.149
3DTexture + Inpainting (10 views)	19.80	0.832	0.145

in Table 6, showing that even with less input views, our rendering network is able to outperform the state-of-the-art texture mapping method. Besides, when dealing with dynamic cases (real-world dataset), as shown in the supplementary video, 3DTexture seems to be sensitive to the reconstruction results. Experiments show that slight flicking of the reconstruction results will lead to incoherence of the rendering results. Moreover, in the areas that suffer from reconstruction errors, the flicking problems will get worse. In contrast, our method is able to produce more stable results on 4D rendering.

4.6 Limitations

As our rendering network needs to use the reconstruction results to perform geometry-guided feature integration, the rendering quality heavily relies on the reconstruction results. If the reconstruction network fails in some parts of the human performer, it will result in rendering artifacts in these parts. The first and second columns of Fig. 10 demonstrate that failure reconstruction will cause rendering artifacts in the corresponding parts.

To improve the rendering quality in the parts where reconstruction network fails, a straightforward way is to fine-tune the rendering network to a specific human performer. However, the geometry representation we used is mesh, and hence the geometry-guided pixel-wise feature integration used in our method is indifferentiable with the reconstructed geometry, making us unable to refine the geometry in the process of fine-tuning. Thus, fine-tuning will cause blurring in the parts that suffer from rendering artifacts. The third and fourth columns of Fig. 10 show the rendering results that are rendered with general model and fine-tuned model, respectively. It can be seen that fine-tuning is unable to solve the rendering artifacts caused by reconstruction failure and will result in blurring in these parts.

5 CONCLUSION AND DISCUSSION

In this paper, we introduce HDhuman, a generic method that uses pixel-aligned spatial transformer and geometry-guided pixel-wise feature integration for high-quality human reconstruction and rendering using sparse views. Experiments show that our rendering quality significantly outperforms both the prior generic and specific methods. Our work demonstrates that high-quality human rendering in sparse views is possible without any network fine-tuning, which can serve as an important baseline in the area of human rendering.

In future, a research direction that is worthy to explore is to use a geometry representation that is differentiable in our framework. For example, we could encode the reconstructed mesh to a volume, and then perform differentiable geometry-guided visibility reasoning for each voxel and finally use volume rendering techniques to render novel views. In this way, the geometry will be differentiable, enabling us to refine the geometry in the process of fine-tuning.

6 ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (62122058, 62171255, 62171317) and Guoqiang Institute of Tsinghua University (2021GQG0001).

REFERENCES

- [1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *Proc. European Conference on Computer Vision*, 2020, pp. 405–421. 1, 1, 2.2
- [2] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 1, 4.2, 4.3, 4.4
- [3] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–16, 2015. 1
- [4] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2019. 1, 2.1, 3.1.1, 3.1.1, 3.1.2
- [5] L. Liu, W. Xu, M. Habermann, M. Zollhöfer, F. Bernard, H. Kim, W. Wang, and C. Theobalt, "Learning dynamic textures for neural rendering of human actors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 10, pp. 4009–4022, 2021. 1
- [6] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, 2019. 1, 4.2
- [7] Y. Liu, X. Cao, Q. Dai, and W. Xu, "Continuous depth estimation for multi-view stereo," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2121–2128. 2.1
- [8] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–13, 2015. 2.1
- [9] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor *et al.*, "Fusion4D: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–13, 2016. 2.1
- [10] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, and Y. Liu, "Function4D: Real-time human volumetric capture from very sparse consumer RGBD sensors," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5746–5756. 2.1
- [11] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu, "Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera," *ACM Transactions on Graphics*, vol. 36, no. 4, p. 1, 2017. 2.1
- [12] Q. Feng, Y. Liu, Y.-K. Lai, J. Yang, and K. Li, "FOF: Learning fourier occupancy field for monocular real-time human reconstruction," in *NeurIPS*, 2022. 2.1
- [13] K. Li, H. Wen, Q. Feng, Y. Zhang, X. Li, J. Huang, C. Yuan, Y.-K. Lai, and Y. Liu, "Image-guided human reconstruction via multi-scale graph transformation networks," *IEEE Transactions on Image Processing*, vol. 30, pp. 5239–5251, 2021. 2.1
- [14] X. Li, J. Huang, J. Zhang, X. Sun, H. Xuan, Y.-K. Lai, Y. Xie, J. Yang, and K. Li, "Learning to infer inner-body under clothing from monocular video," *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2.1
- [15] S. Saito, T. Simon, J. Saragih, and H. Joo, "PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 84–93. 2.1, 3.1.1, 3.1.2
- [16] M. Habermann, W. Xu, M. Zollhöfer, G. Pons-Moll, and C. Theobalt, "LiveCap: Real-time human performance capture from monocular video," *ACM Transactions on Graphics*, vol. 38, no. 2, pp. 1–17, 2019. 2.1
- [17] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt, "MonoPerfCap: Human performance capture from monocular video," *ACM Transactions on Graphics*, vol. 37, no. 2, pp. 1–15, 2018. 2.1
- [18] E. Corona, A. Pumarola, G. Alenya, G. Pons-Moll, and F. Moreno-Noguer, "SMPLicit: Topology-aware generative model for clothed people," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11875–11885. 2.1
- [19] D. Xiang, F. Prada, C. Wu, and J. Hodgins, "MonoClothCap: Towards temporally coherent clothing capture from monocular video," in *Proc. International Conference on 3D Vision*, 2020, pp. 322–332. 2.1
- [20] P. Zins, Y. Xu, E. Boyer, S. Wuhrer, and T. Tung, "Data-driven 3D reconstruction of dressed humans from sparse views," in *Proc. International Conference on 3D Vision*. IEEE, 2021, pp. 494–504. 2.1
- [21] Y. Zheng, R. Shao, Y. Zhang, T. Yu, Z. Zheng, Q. Dai, and Y. Liu, "DeepMultiCap: Performance capture of multiple characters using sparse multiview cameras," *arXiv preprint arXiv:2105.00261*, 2021. 2.1
- [22] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–15, 2018. 2.2
- [23] G. Chaurasia, S. Duchene, O. Sorkine-Hornung, and G. Drettakis, "Depth synthesis and local warps for plausible image-based navigation," *ACM Transactions on Graphics*, vol. 32, no. 3, pp. 1–12, 2013. 2.2
- [24] J. Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Nießner, "IGNOR: Image-guided neural object rendering," *arXiv preprint arXiv:1811.10720*, 2018. 2.2
- [25] Y. Liu, Q. Dai, and W. Xu, "A point-cloud-based multiview stereo algorithm for free-viewpoint video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 407–418, 2009. 2.2
- [26] G. Riegler and V. Koltun, "Free view synthesis," in *Proc. European Conference on Computer Vision*, 2020. 2.2
- [27] X. Suo, Y. Jiang, P. Lin, Y. Zhang, M. Wu, K. Guo, and L. Xu, "Neuralhumanfvv: Real-time neural volumetric human performance rendering using RGB cameras," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6226–6237. 2.2
- [28] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *arXiv preprint arXiv:1805.09817*, 2018. 2.2
- [29] E. Penner and L. Zhang, "Soft 3D reconstruction for view synthesis," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–11, 2017. 2.2, 4.2
- [30] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–14, 2019. 2.2
- [31] J. Flynn, M. Broxton, P. Debevec, M. DuVall, G. Fyffe, R. Overbeck, N. Snavely, and R. Tucker, "DeepView: View synthesis with learned gradient descent," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2367–2376. 2.2

- [32] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," in *Proc. International Conference on Neural Information Processing Systems*, 2020. 2.2
- [33] Z. Zheng, H. Huang, T. Yu, H. Zhang, Y. Guo, and Y. Liu, "Structured local radiance fields for human avatar modeling," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 15 893–15 903. 2.2
- [34] F. Zhao, W. Yang, J. Zhang, P. Lin, Y. Zhang, J. Yu, and L. Xu, "HumanNeRF: Efficiently generated human radiance field from sparse inputs," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7743–7753. 2.2
- [35] T. Xu, Y. Fujita, and E. Matsumoto, "Surface-aligned neural radiance fields for controllable 3d human synthesis," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 15 883–15 892. 2.2, 4.2
- [36] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 314–14 323. 2.2
- [37] Q. Shuai, C. Geng, Q. Fang, S. Peng, W. Shen, X. Zhou, and H. Bao, "Novel view synthesis of human interactions from sparse multi-view videos," in *Proc. ACM SIGGRAIPH*, 2022, p. 10. 2.2
- [38] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "HumanNeRF: Free-viewpoint rendering of moving people from monocular video," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 210–16 220. 2.2
- [39] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327. 2.2
- [40] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Proc. ACM SIGGRAIPH*, vol. 21, no. 4, pp. 163–169, 1987. 3.1.1
- [41] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2019. 3.1.2
- [42] G. Riegler and V. Koltun, "Stable view synthesis," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 216–12 225. 3.2
- [43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 3.2
- [44] [Online]. Available: <https://web.twindom.com/> 4.1
- [45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 4.1
- [46] M. Waechter, N. Moehrle, and M. Goesele, "Let There Be Color! large-scale texturing of 3D reconstructions," in *Proc. European Conference on Computer Vision*, 2014, pp. 836–850. 9, 4.5, 6
- [47] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM SIGGRAPH Asia*, 2003, p. 313–318. 4.5