SAILOR: Synergizing Radiance and Occupancy Fields for Live Human Performance Capture

ZHENG DONG, State Key Laboratory of CAD&CG, Zhejiang University, China
KE XU, City University of Hong Kong, China
YAOAN GAO, State Key Laboratory of CAD&CG, Zhejiang University, China
QILIN SUN, The Chinese University of Hong Kong, Shenzhen and Point Spread Technology, China
HUJUN BAO, State Key Laboratory of CAD&CG, Zhejiang University, China
WEIWEI XU*, State Key Laboratory of CAD&CG, Zhejiang University, China
RYNSON W.H. LAU, City University of Hong Kong, China

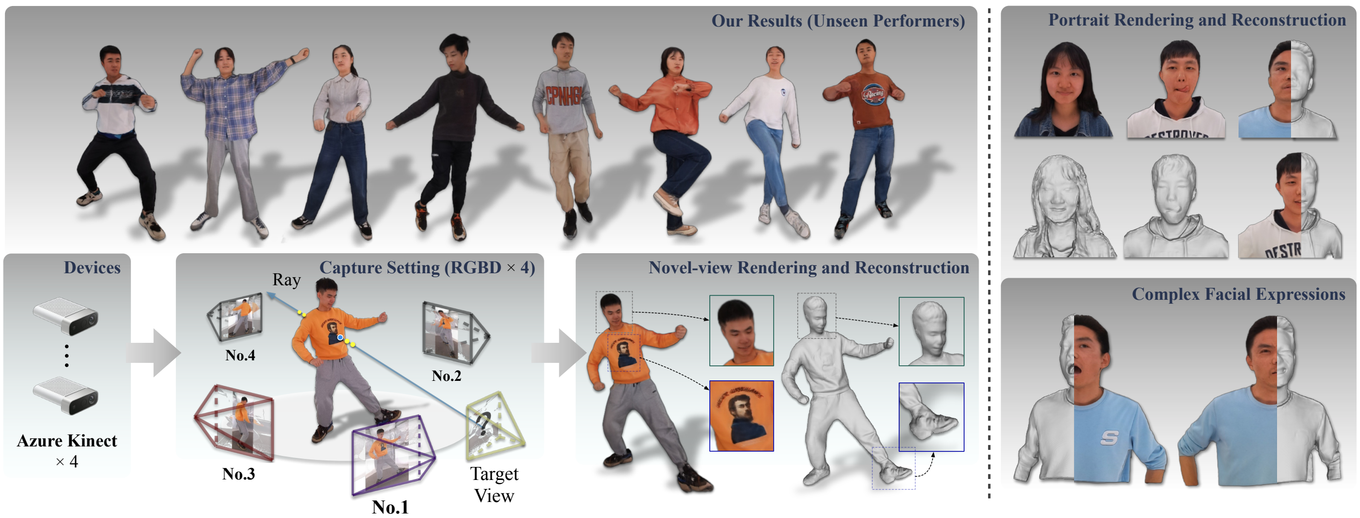

Fig. 1. We propose SAILOR, a novel method for human free-view rendering and reconstruction from very sparse (*e.g.*, 4) RGBD streams with low latency. Our approach learns a hybrid representation of radiance and occupancy fields, which can handle unseen performers without fine-tuning and generate high-quality appearance details in the novel view. In addition, it naturally supports portrait rendering and reconstruction without re-training on the corresponding datasets.

Immersive user experiences in live VR/AR performances require a fast and accurate free-view rendering of the performers. Existing methods are mainly based on Pixel-aligned Implicit Functions (PIFu) or Neural Radiance Fields (NeRF). However, while PIFu-based methods usually fail to produce photo-realistic view-dependent textures, NeRF-based methods typically lack local geometry accuracy and are computationally heavy (*e.g.*, dense sampling of 3D points, additional fine-tuning, or pose estimation). In this work, we propose a novel generalizable method, named SAILOR, to create high-quality human free-view videos from very sparse RGBD live streams. To produce view-dependent textures while preserving locally accurate geometry, we integrate PIFu and NeRF such that they work synergistically by conditioning the PIFu on depth and then rendering view-dependent textures through NeRF. Specifically, we propose a novel network, named SRONet, for this hybrid representation. SRONet can handle unseen performers without fine-tuning. Besides, a neural blending-based ray interpolation approach, a tree-based voxel-denoising scheme, and a parallel computing pipeline are incorporated to reconstruct and render live free-view videos at 10 fps on average. To evaluate the rendering performance, we construct a real-captured RGBD benchmark from 40 performers. Experimental results show that SAILOR outperforms existing human reconstruction and performance capture methods.

*Corresponding author

Authors' addresses: Zheng Dong, State Key Laboratory of CAD&CG, Zhejiang University, China, zhengdong@zju.edu.cn; Ke Xu, City University of Hong Kong, China, kkangwing@gmail.com; Yaoan Gao, State Key Laboratory of CAD&CG, Zhejiang University, China, yaoangao@zju.edu.cn; Qilin Sun, The Chinese University of Hong Kong, Shenzhen and Point Spread Technology, China, sunqilin@cuhk.edu.cn; Hujun Bao, State Key Laboratory of CAD&CG, Zhejiang University, China, bao@cad.zju.edu.cn; Weiwei Xu, State Key Laboratory of CAD&CG, Zhejiang University, China, xww@cad.zju.edu.cn; Rynson W.H. Lau, City University of Hong Kong, China, rynson.lau@cityu.edu.hk.

0730-0301/2023/12-ART205 $15.00
https://doi.org/10.1145/3618370

CCS Concepts: • **Computing methodologies → Image-based rendering**; *Mesh geometry models.*

Additional Key Words and Phrases: human performance capture, high-quality human free-view videos, occupancy and radiance fields, hybrid representation.

ACM Reference Format:
Zheng Dong, Ke Xu, Yaoan Gao, Qilin Sun, Hujun Bao, Weiwei Xu, and Rynson W.H. Lau. 2023. SAILOR: Synergizing Radiance and Occupancy Fields

for Live Human Performance Capture. *ACM Trans. Graph.* 42, 6, Article 205 (December 2023), 15 pages. https://doi.org/10.1145/3618370

1 INTRODUCTION

The creation of free-viewpoint videos featuring humans is an actively researched topic in the fields of computer graphics and vision. It serves as a critical component for a wide range of applications, including virtual and augmented reality, distance education, and telecommunications. To provide immersive experiences to the users, applications like remote presence and teleconferencing [Orts-Escolano et al. 2016; Zhang et al. 2022b] require capturing high-fidelity 3D human models from consumer-affordable capture rigs in real-time with low latency from live video streams.

Recently, neural implicit representations have been widely used in human performance capture. Pixel-aligned implicit functions (PIFu) can reconstruct dynamic 3D human body surface meshes with details and textures [Dong et al. 2022; Feng et al. 2022; Li et al. 2020a,b; Saito et al. 2019, 2020; Yu et al. 2021b], where the surface meshes are extracted from a reconstructed occupancy field, and the surface textures are obtained using a trained network for predicting the RGB colors of surface points. Neural radiance fields (NeRF) are another increasingly popular family of techniques that leverage coordinate-based networks to encode volumetric density and color fields. It may synthesize photorealistic novel-view images with highly detailed 3D space sampling [Gafni et al. 2021; Mildenhall et al. 2020; Pumarola et al. 2021; Tretschk et al. 2021]. However, Both two lines of methods still have weaknesses. First, the surface-texture-based rendering method of PIFu may lead to blurred rendering results in some cases, and PIFu cannot handle view-dependent effects or the transparency of human hairs. Second, NeRF suffers from slow rendering speed and weak generalization ability. Latest generalizable NeRF methods may fail to handle unseen subjects nor novel image rendering from sparsely captured views [Chen et al. 2021b; Gao et al. 2022; Jiang et al. 2022; Kwon et al. 2021; Peng et al. 2021b]. Typically, fine-tuning is necessary to achieve high-quality rendering results for a new subject [Gafni et al. 2021; Lin et al. 2022; Shao et al. 2022b; Wang et al. 2021a; Yu et al. 2021a]. Hence, developing a generalizable method that can create live and photorealistic human free-viewpoint videos with sparse capture rigs is still challenging.

In this work, we aim to address the above challenge with two observations. First, we observe that the PIFu and NeRF representations can be synergized through depth information in such a way that while NeRF uses global radiance information to synthesize high-quality views, its shape ambiguity can be reduced by incorporating the occupancy field in PIFu, which helps guide surface reconstruction. Second, we observe that synergizing the PIFu and NeRF representations demands for accurate depth information, as relying on image features solely may still yield unreliable shape estimation results in the occupancy fields, especially under sparse capture settings. If we can obtain accurate depths, we may constrain the PIFu surface field and align image features better with surfaces to model colors in the radiance fields, resulting in a generalization of the learned human model to novel poses and appearances.

Based on the above two observations, we propose a novel human performance capture method, SAILOR, for creating high-quality free-view videos from sparse (*e.g.*, 4) RGBD video streams. It is a generalizable method that can handle unseen performers without fine-tuning (see Fig. 1). Our method has three main steps to process the RGBD video inputs, at the core of which is a novel neural 3D human representation that takes both advantages of PIFu and NeRF for high-quality geometry and photorealistic appearance reconstruction. First, we train a UNet-like [Ronneberger et al. 2015] depth denoising network to reduce noise and fill possible holes in the raw depth maps. We condition our method on depth denoising, as it provides readily robust geometry cues for correcting topological errors caused by unseen performers/gestures in the live streams. Second, we propose a novel network (called SRONet) to model neural performers through a combination of occupancy and neural radiance fields. Specifically, SRONet constructs 3D human surfaces in the soft occupancy field based on the pixel-aligned denoised depth features, and renders high-quality appearances in the color field conditioned on both image-aligned color features and geometry features. We also construct a tree structure [Liu et al. 2020; Lombardi et al. 2021] from denoised depths, based on which a novel voxel-denoising scheme is proposed to constrain the sampling points inside voxels falling on the body surface during inference. Third, a neural blending-based ray interpolation scheme is proposed to render novel-view images in 1K resolution with small computational overheads.

We note that the Unisurf [Oechsle et al. 2021] may be closely related to ours, as we both combine the occupancy and radiance fields in one model. Specifically, Unisurf [Oechsle et al. 2021] is proposed for solid object reconstruction, which essentially adopts a coarse-to-fine strategy to locate and refine solid object surfaces via volume rendering of NeRF and surface rendering in the occupancy field, respectively. However, as Unisurf uses the occupancy field for rendering, it does not incorporate the pixel-aligned features of PIFu and therefore is a non-generalizable method. In practice, our method runs two magnitudes faster than Unisurf in rendering and can handle unseen performers without any fine-tuning.

To summarize, this work makes the following contributions:

- A novel human performance capture method (called SAILOR) with a hybrid network (SRONet), which synergizes occupancy and radiance fields conditioned on a depth denoising process and its resulting pixel-aligned RGBD features. SAILOR is generalizable to handle unseen performers under a sparse RGBD camera setting without fine-tuning.
- An applicable system that incorporates a tree-based structure, a voxel denoising scheme, a neural blending-based ray interpolation approach, and a parallel computing pipeline. It creates free-view rendering results in 1K resolution at 10 fps on average.
- A real-captured human benchmark, which contains multi-view RGBD videos captured from 40 performers (with ~4,000 frames per person), covering various actions.

Extensive experiments on performers with diverse gestures, motions, and clothing, verify the effectiveness of SAILOR against existing human performance capture methods in terms of reconstruction and rendering accuracy.

2 RELATED WORK

2.1 Monocular Human Performance Capture

A line of methods is proposed to use monocular videos for human performance capture. Xu *et al.* [2018] propose the first markerless deep method, which computes a textured template mesh of static T-pose for each performer, and models the articulated motions and non-rigid surface deformations via a combination of 2D/3D pose estimations and silhouette-based surface refinement, respectively. The T-pose (or A-pose) mesh is then widely adopted [Dou et al. 2017, 2016; Habermann et al. 2021, 2019, 2020; Li et al. 2021; Newcombe et al. 2015a, 2011; Su et al. 2020; Yu et al. 2018], based on which motions are modeled by estimating the non-rigid deformations from the template mesh. Xiang *et al.* [2020] propose the statistical deformation models for clothing capturing. Zhao *et al.* [2022c] propose a dynamic surface network to predict dynamic offsets and texture maps based on the SMPL [Loper et al. 2015] template and a reference-based rendering network that combines the predicted offsets and texture maps to render novel human avatar images.

Pixel-aligned implicit functions (PIFu) [Saito et al. 2019, 2020] show promising high-resolution reconstruction results of textured objects compared with previous 3D representations (*e.g.*, SMPL [Loper et al. 2015; Pavlakos et al. 2019], voxels [Zheng et al. 2019], points [Yifan et al. 2019], and meshes [Alldieck et al. 2019; Zhu et al. 2022]). Li *et al.* [2020a] propose an octree-based surface localization method and a mesh-free rendering method to apply PIFu for monocular human performance capture. Later methods incorporate pre-computed template meshes [Li et al. 2020b], human parsing maps [Chan et al. 2022b], 3DMM [Cao et al. 2022], and SMPL [Chan et al. 2022a; Feng et al. 2022; Xiu et al. 2022; Zheng et al. 2021] with implicit functions to represent 3D humans with motions.

Neural radiance field (NeRF) [Mildenhall et al. 2020] is another popular 3D implicit representation that utilizes classic volumetric rendering to produce free-view images. To handle dynamic scenes, some methods [Park et al. 2021a; Peng et al. 2023; Pumarola et al. 2021; Tretschk et al. 2021] extend NeRF by constructing continuous deformation fields. The deformation fields typically map the observed coordinates to canonical coordinates of a template of the target, following the non-rigid reconstruction-and-tracking scheme [Newcombe et al. 2015b]. Peng *et al.* [2021b] construct the deformation field by leveraging a set of structured latent codes to represent the performer's local geometry and appearance. In [Chen et al. 2021b], 3D positions, shapes, and poses are incorporated to guide the construction of the deformation field. These methods [Chen et al. 2021b; Jiang et al. 2022; Peng et al. 2021b] rely on parametric human models [Joo et al. 2018; Kocabas et al. 2020; Loper et al. 2015] to handle human topology changes under motions. Recently, Weng *et al.* [2022] proposed to model skeletal rigid and non-rigid motions via a discrete grid and a continuous field, respectively.

Some other methods [Gafni et al. 2021; Hu et al. 2023; Su et al. 2022, 2021; Xian et al. 2021] handle dynamic scenes by conditioning the NeRF on additional inputs to change the radiance field of the scene directly. Xian *et al.* [2021] condition the NeRF on the timestamps of the input RGBD video (where D is estimated by a video depth estimation method), and use depth as supervision to refine the scene geometry. Gafni *et al.* [2021] condition the NeRF on a set of latent codes (computed from video frames and the background image) and a 3D morphable model (for tracking facial expressions and poses). The HyperNeRF method [Park et al. 2021b] combines the deformation field and the conditioning networks on latent deformation and appearance codes. Recently, Kim *et al.* [2023] extended the HumanNeRF [Weng et al. 2022] to support rendering of multiple performers, by introducing a set of latent identity codes and pose-conditioned codes.

While monocular videos are convenient and of lower cost, a fundamental limitation of monocular methods is the shape-radiance ambiguity caused by partial occlusions. The difference is that our method utilizes a sparse (*e.g.*, 4) set of RGBD cameras for full-body human performance capture and can effectively reduce this ambiguity by integrating PIFu and NeRF representations.

2.2 Volumetric Human Performance Capture

Volumetric capture methods [De Aguiar et al. 2008; Vlasic et al. 2008] typically leverage multiple cameras to cover the whole capture volume of performers. A group of methods [Collet et al. 2015; Guo et al. 2019; Işık et al. 2023; Jiakai et al. 2021; Liu et al. 2009; Vlasic et al. 2009; Wang et al. 2021b, 2022; Zhang et al. 2022a; Zhao et al. 2022a] leverage high-end studio (tens up to hundreds of) cameras for accurate 3D reconstruction. Multi-view RGB stereo information is used in [Işık et al. 2023; Wang et al. 2021b, 2022], while RGB and infrared (IR) are combined with silhouette [Collet et al. 2015] and depth [Guo et al. 2019]. These methods are typically unaffordable for novice users.

Recently, a set of methods has been proposed to capture human performance from a sparse set (less than ten views) of RGB(D) cameras. Wu *et al.* [2020] use PointNet++ [Qi et al. 2017] to extract 3D point cloud features and design a CNN to render novel images, while the newly rendered images are used to help further improve the visual hull reconstruction [Matusik et al. 2000]. This method is limited by the low-resolution noisy point cloud representation. A few methods [Dong et al. 2022; Saito et al. 2019, 2020; Shao et al. 2022a; Yu et al. 2021b] combine multi-view RGB(D) information with the PIFu representation to produce accurate geometry reconstruction results. However, learning the colors of surface points from image features of sparse views makes these PIFu-based methods difficult to produce novel view-dependent and photorealistic appearances.

Another line of methods is built upon the generalizable NeRF [Chen et al. 2021a; Yu et al. 2021a], which conditions the NeRF on pixel-aligned image features. Some methods [Gao et al. 2022; Kwon et al. 2021] factorize NeRF into a canonical NeRF and a deformation field, and model the deformation field through learning mappings from the surfaces of 3D body parametric models to the 3D volume. In [Peng et al. 2021a], a neural blend weight field in canonical space is combined with the skeleton-driven deformation [Lewis et al. 2000] to generate deformation fields. Wang *et al.* [2021a] combine NeRF with image-based rendering [Chen and Williams 1993; Debevec et al. 1998; Hedman et al. 2018], in which the colors and densities of target view are computed by aggregating the image features of neighboring source views. Mihajlovic *et al.* [2022] condition the NeRF on 3D keypoints to encode robust spatial 3D information. Some methods also condition NeRF with SMPL or pose [Liu et al. 2021], and texel-aligned (pose, image, and camera position) features [Remelli et al. 2022] for drivable volumetric avatar rendering.

Recent endeavors are made to mitigate the geometry ambiguities of NeRF. Zhao *et al.* [2022b] construct a pose-based deformation field for modeling geometry under motions. Shao *et al.* [2022b] propose to regress both occupancy and densities from multi-view RGB features,

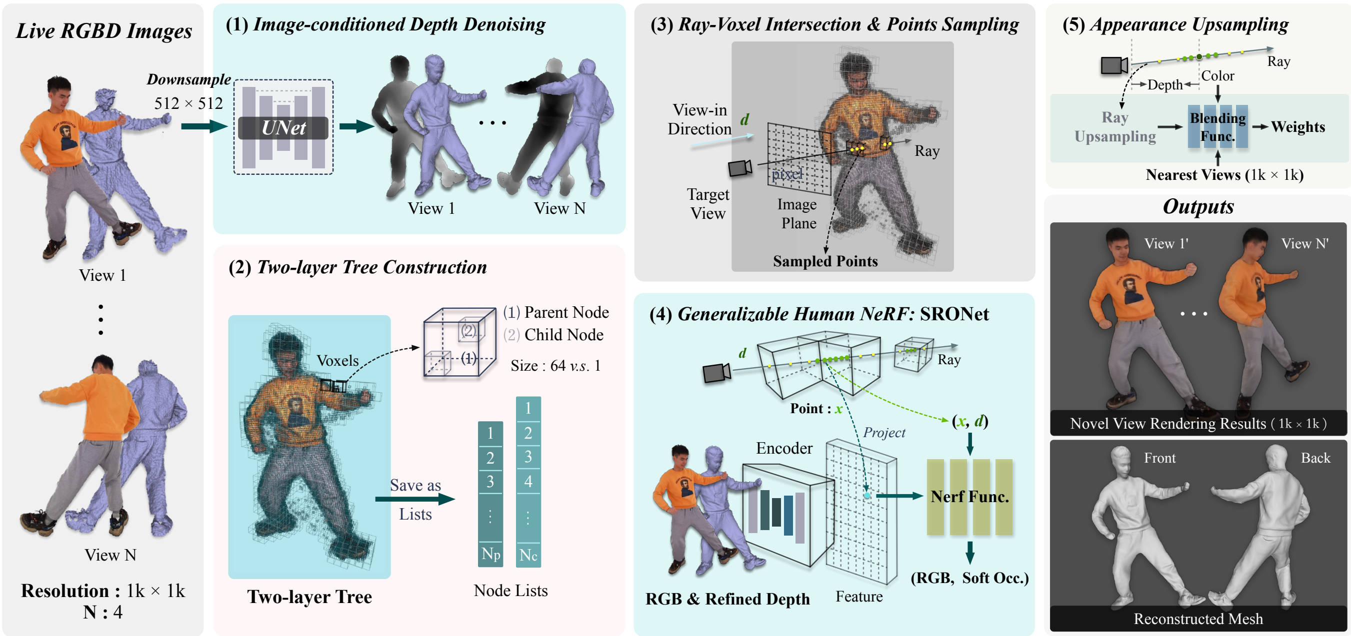

Fig. 2. Given RGBD streams captured by 4 Azure Kinect sensors as inputs, **(1)** a *Depth Denoising Module* first removes noise and fills the holes of raw depths conditioned on the RGB images. **(2)** With the denoised depths, a *Two-Layer Tree Structure* is constructed to store the global geometry in discretization. **(3)** Efficient *Ray-Voxel Intersection* and *Points Sampling* are performed for rendering. **(4)** A novel *SRONet* network is proposed to synergize the radiance and occupancy fields, for 3D reconstruction and free-view rendering. **(5)** The outputs of SRONet are then upsampled via *Ray Upsampling* and *Neural Blending* to produce the final results in 1k resolution.

in which ground-truth occupancy can be involved for geometry supervision. Lin *et al.* [2022] propose to estimate the depth probability distribution (*i.e.*, depth and confidence maps) for constraining the spatial sampling of NeRF near the surfaces. Nonetheless, deriving geometry proxies (*i.e.*, body parametric models, surface occupancy, and depth distribution in [Lin et al. 2022; Shao et al. 2022b,c; Zhao et al. 2022b]) based on RGB information is often not reliable. The inaccurate local geometry further results in visual blurriness and artifacts on the appearances, which makes the fine-tuning for unseen performers inevitable in these methods.

In this work, we propose a depth-conditioned hybrid representation of PIFu and NeRF to address this geometry/appearance ambiguity problem of unseen performers. By incorporating accurate depth, we show that pixel-aligned RGBD features enable accurate and generalizable surface reconstructions and can guide NeRF to produce high-fidelity appearances in near-real-time.

Fig. 3. Visualization of our depth denoising results and their fused point clouds on our real captured RGBD images. Our depth denoising network can reduce noise and fill in the missing regions (*e.g.*, hair and hand regions marked in black dashed boxes) for the full-body and portrait inputs.

3 OUR METHOD

Our method aims to generate high-quality, and high-resolution free-view videos in near-real time, given $\mathcal{N}$ RGB-D streams of $\{\mathbf{I}^i, \mathbf{D}^i\}_{i=1,\ldots,\mathcal{N}}$ captured by a sparse set of Kinect-V4 sensors, where $\mathbf{I}$ and $\mathbf{D}$ represent the RGB and depth images, respectively, and $\mathcal{N}$ is set to 4 in our implementation.

As illustrated in Fig. 2, our method contains five steps: **(1)** *Image-conditioned Depth Denoising* $\mathcal{F}_d$ removes noise and completes the holes in the noisy multi-view depth images. **(2)** *Two-layer Tree Construction* $\mathcal{T}$ divides the human-body volume into two levels, and stores parent-child voxels of the volume as two sequences of nodes in the GPU, based on the denoised depths of $\mathcal{F}_d$. **(3)** For a ray emitted from the target view, *Ray-Voxel Intersection* records the indexes of the two-level voxels that intersected with the ray, along with the depths of the intersected points. *Points Sampling* then records the depths in the target view of the sampled points within each voxel. These sampled points along the ray are located near the 3D human surface for efficient appearance rendering. **(4)** For each sampled point on the ray, our *SRONet* predicts its RGB and soft occupancy values, based on the denoised depths and RGB images. The color and depth of the sampled pixel in the target view are computed via a blending function. This step processes the rays for an image at 1/4

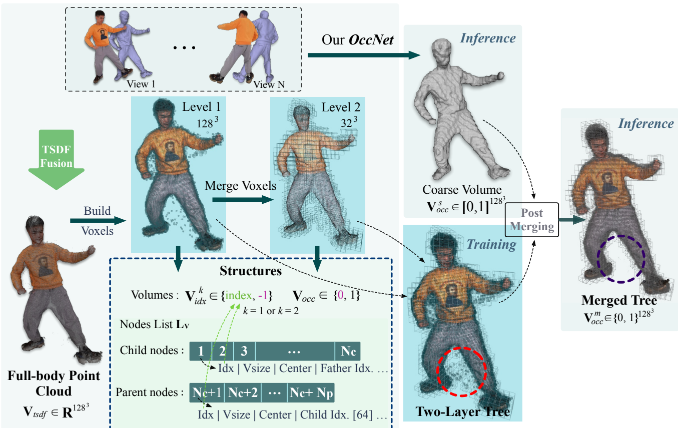

Fig. 4. The Two-layer Tree Construction process. (1) Converting the denoised depth maps into a full-body point cloud via the *TSDF-Fusion* [Newcombe et al. 2011]. (2) Building a volume based on the generated point cloud, where the voxels occupied by points are stored in GPU. (3) Merging the small voxels (Level 1) into large voxels (Level 2) in a ratio of 64 : 1, and the voxels are transformed into a node list. During inference, the tree will be merged with SRONet's reconstructed volume to eliminate floating voxels outside the body (red circle).

resolution of the final rendering result. **(5)** *Ray upsampling* extends one emitted ray (in one pixel) into four sub-rays (corresponding to four sub-pixels that share the same color and depth). *Neural Blending* then produces the appearance details for each sub-pixel by aggregating colors from the nearest two views.

3.1 Image-conditioned Depth Denoising: $\mathcal{F}_d$

Accurate geometry information from the depth plays a vital role in our method for rendering accuracy and stability. However, the raw depths acquired by Kinect cameras are often noisy and incomplete. To refine the raw depth, we train an UNet-like [Ronneberger et al. 2015] depth denoising network (denoted as $\mathcal{F}_d$) to perform the denoising process as $\mathbf{D}^i_{rf} = \mathcal{F}_d(\mathbf{I}^i, \mathbf{D}^i)_{i=1,...,\mathcal{N}}$, where $\mathbf{I}^i$ and $\mathbf{D}^i$ are the input RGB and Depth images of view i, respectively. $\mathcal{F}_d$ helps remove high-frequency noise, fill the missing parts, and output a reliable depth map $\mathbf{D}^i_{rf}$ for our rendering system. To train our depth denoising network $\mathcal{F}_d$, we employ a training dataset with high-quality 3D human scans and simulate the depth noise on the ground-truth depth maps (see Sec. 5). The designed loss functions, network structure, and training details of $\mathcal{F}_d$ are provided in the supplemental material.

Fig. 3 shows that our depth denoising module performs well on our real captured data, *i.e.*, removing noise and filling in the missing regions (*e.g.*, black dashed boxes). However, due to the sparse capture setting, full-body fused points from $\mathbf{D}^i_{rf}$ may still have holes in some invisible areas (*e.g.*, yellow dashed boxes). We construct the two-layer tree structure to cover such regions as described next.

3.2 Two-layer Tree Construction: $\mathcal{T}$

To constrain the rendering range to be near the 3D surface of the human body, we exploit the geometric cue (*i.e.*, denoised depths), by computing the fused point cloud $\mathbf{P}_{rf}$ from $\mathbf{D}^i_{rf}$ and leveraging $\mathbf{P}_{rf}$ to construct a Two-Layer Tree (denoted as $\mathcal{T}$) in the GPU end.

Construction of $\mathcal{T}$. Fig. 4 shows the two-layer tree construction process. First, we adopt the *TSDF-Fusion* [Newcombe et al. 2011] to convert the denoised depths $\mathbf{D}^i_{rf}(i = 1, ..., \mathcal{N})$ into a full-body point cloud $\mathbf{P}_{rf}$. Second, based on the fused TSDF volume $\mathbf{V}_{tsdf}$, we binarize $\mathbf{V}_{tsdf}$ to generate an occupied volume $\mathbf{V}_{occ}$. Third, we merge the valid voxels with a value 1 into large voxels in a ratio of 4^3 : 1 (*i.e.*, each large voxel can have up to 64 child voxels). Finally, we store all the valid voxels as a global list $\mathbf{L_v}$ in the GPU, where each node in $\mathbf{L_v}$ records the index, size, and position (in world coordinate) of the corresponding voxel. We have implemented $\mathcal{T}$ with CUDA acceleration (~6ms), which supports the storage of multiple batches (or performers) simultaneously.

Voxel Denoising via Post-merging. The raw point cloud $\mathbf{P}_{rf}$ often has undesirable floating voxels (red circle in Fig. 4). Hence, we apply a post-merging step to eliminate these voxels during inference. Specifically, we first use our SRONet (Sec. 3.3) to construct a soft occupied volume $\mathbf{V}^s_{occ}$, where voxel values are in $[0, 1]$. We then fuse the two volumes $\mathbf{V}^s_{occ}$ and $\mathbf{V}_{occ}$ with a union operation, as:

$$\mathbf{V}^m_{occ}(x) = \begin{cases} \mathcal{B}(\mathbf{V}^s_{occ}(x), \beta) \mid \mathbf{V}_{occ}(x) & \mathbf{V}^s_{occ}(x) \geq \gamma \\ 0 & else \end{cases}, \quad (1)$$

where $\mathcal{B}(\cdot, \beta)$ is a binarization function with threshold β, and γ is an occupancy threshold to eliminate external voxels.

Fig. 5 shows one example in which the tree after post-merging stores voxels on the human surface, producing a better rendering result. We adopt the surface localization algorithm in MonoPort [Li et al. 2020a] to accelerate (~8ms) the extraction of coarse volume $\mathbf{V}^s_{occ}$ in a resolution of 128^3.

To render a novel-view image, we project rays from pixels of the target view, and perform *Ray-Voxel Intersection* to identify voxels (in both levels of $\mathcal{T}$) that are intersected with the rays, and perform *Points Sampling* to sample the points inside the intersected voxels on the rays (See the supplemental for details).

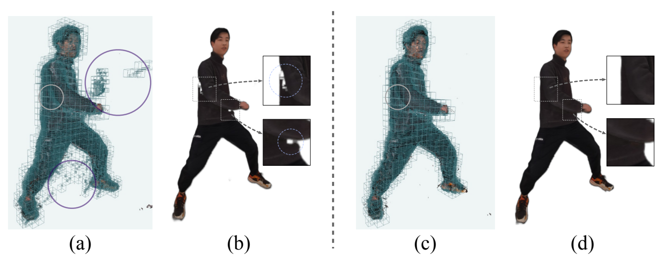

Fig. 5. Visual examples of Two-layer tree construction without voxel denoising (a) and with voxel denoising (c), and the corresponding novel-view rendering results (b,d). Our voxel denoising via post-merging can remove noisy voxels (marked in circles), which further improves rendering quality.

3.3 Generalizable Human NeRF: SRONet

We propose **SRO**Net to **S**ynergize **R**adiancce and **O**ccupancy fields with robust depths to learn robust and generalizable human representations under sparse views. We use the soft occupancy field to represent the human body surface, the reconstruction of which is conditioned on denoised depth $\mathbf{D}_{rf}$ to ensure accuracy and generalization capability. We then condition the color field of NeRF on pixel-aligned RGB and geometry features. The pixel-aligned geometry features play a key role in producing human textures that can fit

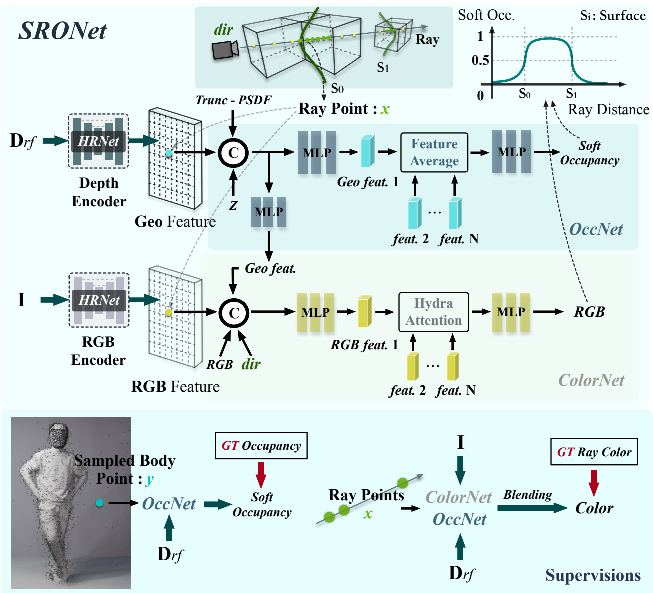

Fig. 6. Overview of SRONet. (1) *OccNet* takes denoised depths and a 3D point as inputs, and predicts the soft occupancy $\in [0, 1]$ value of the point. (2) *ColorNet* predicts the color value of the point based on the RGB images, view directions, and the geometric features. (3) We use the soft occupancy values to compute the weights for blending colors of the sampling points on a ray, to produce the final pixel color of the novel view.

the local surfaces correctly. Fig. 6 illustrates the network structure and supervision signals of our proposed SRONet.

Depth-conditioned Occupancy Field. For a sampled point $\mathbf{x}$ on the emitted ray $\mathbf{l}$, we first predict its soft occupancy value $o_{\mathbf{x}} \in [0, 1]$ by aggregating the pixel-aligned depth features of $\mathbf{D}^i_{rf} (i = 1, ..., \mathcal{N})$, where $o_{\mathbf{x}}$ is the probability of the point $\mathbf{x}$ locating inside the human body ($o_{\mathbf{x}} = 0.5$ indicates that the point is on the surface). We use a sub-network, named OccNet, to model this occupancy field $\mathcal{F}_o$ as:

$$\mathcal{F}_o(\mathbf{x}, \mathbf{D}_{rf}) = f_2(Avg(\{f_1(\mathbf{W}^i(\mathbf{x}), \mathbf{c}^i(\mathbf{x}))\}_{i=1,...,\mathcal{N}})) := o_{\mathbf{x}}, \quad (2)$$

where $\mathbf{W}^i = E_d(\mathbf{D}^i_{rf})$ represents the depth feature map of the i-th view, and $E_d(\cdot)$ is the depth encoder. For the projected 2D image coordinate $\pi^i(\mathbf{x})$ and depth z^i of $\mathbf{x}$ in view i, $\mathbf{W}^i(\mathbf{x})$ is the fetched depth feature vector at $\pi^i(\mathbf{x})$ and $\mathbf{c}^i(\mathbf{x}) = [z^i, p^i(\mathbf{x})]$, where $p^i(\mathbf{x}) \in [-\delta_p, \delta_p]$ is the truncated PSDF value computed based on $\mathbf{D}_{rf}$ and z^i, similar to [Dong et al. 2022]. In Eq. 2, $\mathbf{W}^i(\mathbf{x})$ along with $\mathbf{c}^i(\mathbf{x})$ are fed into the first implicit function f_1 to obtain the geometric features. These features are then processed by an average pooling operator Avg and further fed into the second implicit function f_2 for occupancy querying. The queried value $o_{\mathbf{x}}$ is used for both reconstruction and rendering.

Geometry-conditioned Color Field. We predict the view-dependent color value $\mathbf{c}_{\mathbf{x}} \in \mathbb{R}^3$ by aggregating the pixel-aligned RGB features of $\mathbf{I}^i (i = 1, ..., \mathcal{N})$, conditioned on the local view direction $\mathbf{d}^i$ and geometric features $\mathbf{ft}^i_{geo}$, where $\mathbf{d}^i = \mathbf{R}^i\mathbf{d}$ with $\mathbf{d}$ being the view direction in world coordinate, and $\mathbf{ft}^i_{geo} = f_3(\mathbf{W}^i(\mathbf{x}), \mathbf{c}^i(\mathbf{x}))$. We use another sub-network, named ColorNet, to model color field $\mathcal{F}_c$ as:

$$\mathcal{F}_c(\mathbf{x}, \mathbf{I}, \mathbf{d}) = f_5(\mathcal{H}(\{f_4(\mathbf{M}^i(\mathbf{x}), \mathbf{ft}^i_{geo}, \mathbf{d}^i, \mathbf{rgb}^i)\}_{i=1,...,\mathcal{N}})) := \mathbf{c}_{\mathbf{x}}, \quad (3)$$

where $\mathbf{M}^i = E_c(\mathbf{I}^i)$ is the rgb feature map in view i, and $E_c(\cdot)$ is the rgb encoder. $\mathbf{M}^i(\mathbf{x})$ and $\mathbf{rgb}^i \in \mathbb{R}^3$ are the fetched rgb feature map and rgb pixel values of the point, respectively. f_4 and f_5 are both implicit functions to process features. We implement the feature fusion process $\mathcal{H}$ as a transformer encoder [Vaswani et al. 2017] with hydra attention blocks [Bolya et al. 2023] and adopt the fully fused scheme in [Müller et al. 2021] to accelerate this process.

Rendering. To produce the final color $\hat{\mathbf{C}}(\mathbf{l})$ for the emitted ray $\mathbf{l}$, we use the unified surface and volume rendering function in [Oechsle et al. 2021] to blend color vector $\mathbf{c}_{\mathbf{x}}$ for each sampled point $\mathbf{x}$, as:

$$\hat{\mathbf{C}}(\mathbf{l}) = \sum_{i=1}^{M} o_{\mathbf{x}}(i) \prod_{j<i} (1 - o_{\mathbf{x}}(j))\mathbf{c}_{\mathbf{x}}(i), \quad (4)$$

where $\omega_{\mathbf{x}}(i) = o_{\mathbf{x}}(i) \prod_{j<i}(1 - o_{\mathbf{x}}(j))$ is the blending weight for the i-th point $\mathbf{x}_i$ sampled on ray $\mathbf{l}$, and M is the number of sampled points. When $\mathbf{x}_i$ is far from the body surface, the occupancy value $o_{\mathbf{x}}(i)$ is close to 0 or 1. Hence, the weight $\omega_{\mathbf{x}}$ tends to have high response values only for the points near the surface. This aligns with our motivation of sampling points inside the surface voxels. Similarly, we compute depth $\hat{D}(\mathbf{l})$ of the surface intersected with the emitted ray by blending depth $d(i)$ of points, as $\hat{D}(\mathbf{l}) = \sum_{i=1}^{M} \omega_{\mathbf{x}}(i)d(i)$.

Optimization of SRONet. We adopt two loss functions to supervise the reconstruction and rendering process of SRONet.

***(1)** Geometry and Color Synergistic Loss.* We first sample point $\mathbf{y}$ around the body surface (bottom part in Fig. 6), and then measure the difference between the predicted occupancy value $o_{\mathbf{y}}$ and the ground-truth value $o^*_{\mathbf{y}}$ to train our OccNet to learn global geometric information. Meanwhile, we penalize the per-ray error between $\hat{\mathbf{C}}(\mathbf{l})$ and the ground-truth color $\mathbf{C}^*(\mathbf{l})$ to train both our ColorNet and OccNet for learning textures and enhancing geometric details. The two losses work in a synergistic manner, as:

$$L_{\text{syn.}} = \mu_o \cdot \sum_{\mathbf{x} \in S} \mathcal{L}_B(\mathbf{o_y}, \mathbf{o^*_y}) + \mu_c \cdot \sum_{\mathbf{l} \in R} \mathcal{L}_1(\hat{\mathbf{C}}(\mathbf{l}), \mathbf{C}^*(\mathbf{l})), \quad (5)$$

where S and R denote the sampled points and rays set, respectively. $\mathcal{L}_B$ and $\mathcal{L}_1$ are the BCE loss and the smooth L1 loss, while μ_o and μ_c are the balancing weights.

***(2)** Depth Consistency Loss.* We enhance the consistency between the predicted depth value $\hat{D}(\mathbf{l})$ and the GT depth value $D^*(\mathbf{l})$ to improve reconstruction and rendering details, as:

$$L_{D'} = \sum_{\mathbf{l} \in R} \mathcal{L}_2(\hat{D}(\mathbf{l}), D^*(\mathbf{l})), \quad (6)$$

where $\mathcal{L}_2$ is the L2 loss. $D^*(\mathbf{l})$ is fetched from the rendered GT depth map. The complete loss function for SRONet is then a combination of $L_{\text{syn.}}$ and $L_{D'}$, as $L_{\text{syn.}} + \lambda_{D'} L_{D'}$, where $\lambda_{D'}$ is a balance term.

3.4 Appearance Upsampling

We propose a fast ray upsampling scheme to further enhance the rendered image of SRONet with higher resolution and richer details. Compared to LookinGood [Martin-Brualla et al. 2018], which enhances the rendered images, we interpolate each emitted ray into four sub-rays during the rendering. The color of each sub-ray is predicted based on the shared color, depth, and features of the emitted ray, and two high-resolution adjacent RGB inputs, via a neural blending method.

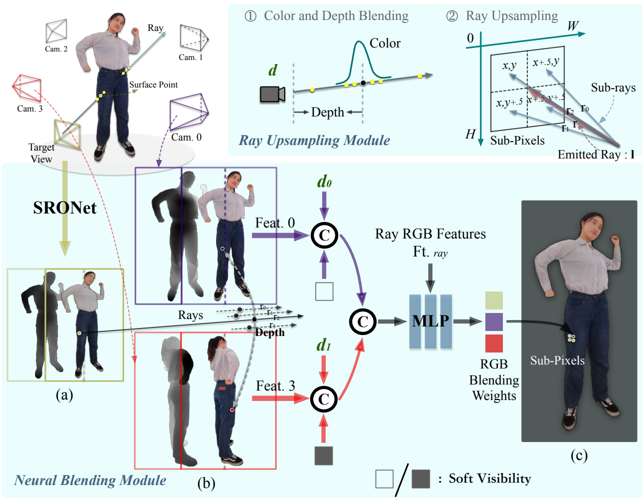

Fig. 7. Overview of the *ray upsampling* and *neural blending* schemes. (1) Four sub-rays are casted from four sub-pixels, which share color, depth, and features from the emitted ray. (2) For each sub-pixel, *neural blending* takes the image features of the two adjacent views, view directions in two local coordinates, and two visibility signals as inputs, to produce weights to blend the emitted ray color with colors of two adjacent high-resolution images.

Ray Upsampling Scheme. As shown in Fig. 7, for an emitted ray $\mathbf{l}$ corresponding to the pixel at (x, y), we first interpolate it into four sub-rays (denoted as $\mathbf{r}_0$, $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$), whose emitted source points are consistent with $\mathbf{l}$, but the sub-pixel positions are assigned as (x, y), $(x + 0.5, y)$, $(x, y + 0.5)$, $(x + 0.5, y + 0.5)$. Then, we obtain the ray color $\hat{\mathbf{C}}(\mathbf{l})$ and the intersected depth $\hat{D}(\mathbf{l})$ using SRONet. We also blend the point features $\mathbf{ft}_{color}$ (output by the hydra attention blocks $\mathcal{H}$ for each sampled point $\mathbf{x}$) for the ray $\mathbf{l}$, via $\sum_{i=1}^{M} \omega_{\mathbf{x}}(i)\mathbf{ft}_{color}(i)$, to obtain the ray color features $\mathbf{ft}_{ray}$. At last, we scatter $\hat{\mathbf{C}}(\mathbf{l})$, $\hat{D}(\mathbf{l})$ and $\mathbf{ft}_{ray}$ into each sub-ray $\mathbf{r}_i (i = 0, 1, 2, 3)$ to obtain the coarse sub-ray features. This ray upsampling scheme enables a 2× increase in spatial resolution, while simply using more rays (4×) takes about 5× more inference time.

Neural Blending Operation. We leverage two adjacent RGB inputs (in 1K resolution) to refine the ray upsampling features, similar to [Zhao et al. 2022b]. Specifically, we first use a UNet to encode the two adjacent raw RGB images into two RGB feature maps. Given the intersected depth value $\hat{D}(\mathbf{l})$ of each sub-ray $\mathbf{r}_i$, we compute the surface position $\mathbf{x} \in \mathbb{R}^3$ and back-project the surface point to the adjacent two views to fetch the colors $\mathbf{C}^{n_0}$, $\mathbf{C}^{n_1}$, and the RGB features $\mathbf{ft}^{n_0}$, $\mathbf{ft}^{n_1}$. We then back-project the surface point to the adjacent two refined depth maps $\mathbf{D}^{n_0}_{rf}$, $\mathbf{D}^{n_1}_{rf}$ to calculate the soft visibility values O^{n_0}, O^{n_1}, which can be written as:

$$O^i = \exp(-\sigma_v \cdot (z^i - d^i_{rf})^2), \tag{7}$$

where i is located in $\{n_0, n_1\}$, and z^i is the projected depth of the surface point in view i. d^i_{rf} is the fetched depth value from $\mathbf{D}^i_{rf}$ in the 2D coordinate $\pi^i(\mathbf{x})$ of view i. σ_v is a weight coefficient determined by depth units. Hence, O^i tends to be 1 when $\mathbf{x}$ is visible in view i, and 0 otherwise. Finally, we feed the RGB features ($\mathbf{ft}^{n_0}$ and $\mathbf{ft}^{n_1}$), two local view directions ($\mathbf{d}^{n_0}$ and $\mathbf{d}^{n_1}$), two visibility values (O^{n_0} and O^{n_1}), along with the ray color features ($\mathbf{ft}_{ray}$) into our neural

Stages	Operations	Time *w/o* acc.	Time *w/* acc.
$\mathcal{F}_d$	Depth denoising	≈78ms	≈18ms
HRNets	Encoding RGBD images in SRONet	≈76ms	≈19ms
Unet in $\mathcal{F}_b$	High-resolution RGB encoding in neural blending	≈25ms	≈6ms
$\mathcal{T}$	Building two-layer tree	≈6ms	-
Voxel Denoising	Voxel post-merging when inference	≈8ms	-
Intersection	Detection of voxels intersected by rays. Recording depths of intersected points	≈10ms	-
Points Sampling	Sampling points within voxels along the rays	≈2ms	-
Ray Querying	Predicting ray colors and depths	≈271ms	≈35ms
Upsampling	Ray upsampling and neural blending	≈5ms	≈1ms
Total	-	≈481ms	< 100ms

Table 1. The running time for each stage of our pipeline *w/o* and *w/* acceleration is reported. Note that we use the three alternative streams for *w/* **acc.**, which further reduces the sum time of ≈ 105ms by 9.5%~19%.

blending network $\mathcal{F}_b$ to obtain the blending weights $\mathbf{W}_\mathbf{x}$, as:

$$\mathcal{F}_b(\mathbf{x}, \mathbf{l}, \mathbf{D}_{rf}, \mathbf{d}) = f_6(\{\mathbf{ft}^i, \mathbf{d}^i, O^i\}_{i=n_0,n_1}, \mathbf{ft}_{ray}) := \mathbf{W}_\mathbf{x}, \tag{8}$$

where f_6 is the implicit function, and $\mathbf{W}_\mathbf{x} \in \mathbb{R}^3$ is used to blend the two adjacent colors and the ray color for point $\mathbf{x}$. The final color $\hat{\mathbf{C}}_\mathbf{r}$ for a sub-ray $\mathbf{r}$ can then be obtained via: $\mathbf{W}_\mathbf{x} \cdot [\mathbf{C}^{n_0}, \mathbf{C}^{n_1}, \hat{\mathbf{C}}(\mathbf{l})]$. See Fig. 7 for illustration and refer to supplemental for training details.

3.5 Parallel Acceleration

We design a parallel acceleration method to leverage multiple GPUs and a single CPU (2 Nvidia RTX 3090 and an Intel i9-13900k in this work) to accelerate the rendering. It aims to distribute the workload regarding input views and ray computations among GPUs, and build a pipeline to reduce the processing latency for each operation.

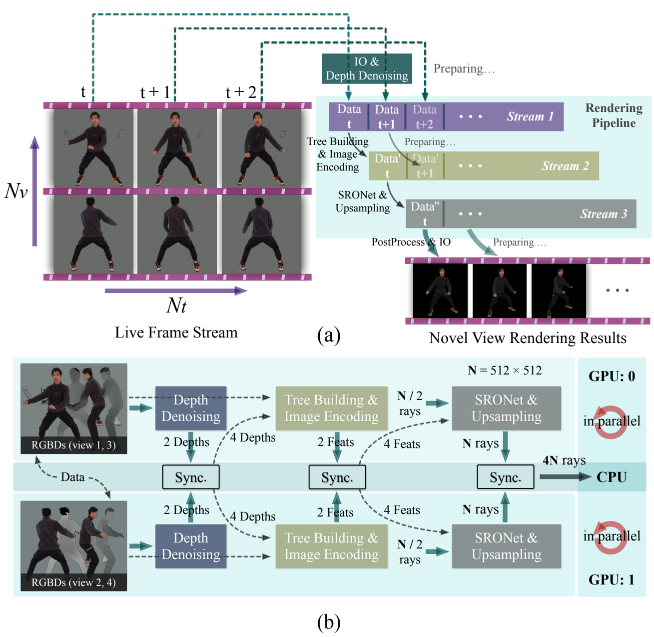

Fig. 8. The parallel computing pipeline of acceleration. Given multi-view RGBD live data as inputs, we use three data streams to process independent tasks alternately: (1) I/O & depth denoising, (2) image encoding & tree building, and (3) SRONet & upsampling. A post-processing & I/O reshape the outputs of data streams into images for display (a). GPU:0 and GPU:1 handle half of the workload of each task in parallel and send the corresponding results to the CPU for data synchronization (b).

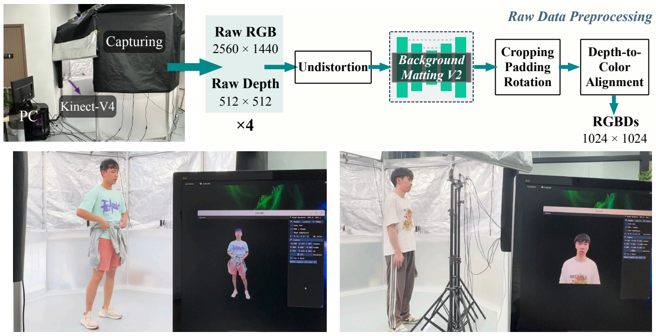

Fig. 9. Preprocessing of the Kinect-V4 raw captured data (upper row) and the online rendering demo (bottom row) for full body and portrait settings.

Specifically, we divide the SAILOR pipeline into three groups of operations, and in each GPU we accelerate these groups of operations with three alternative data streams: (1) I/O (CPU to GPU) and depth denoising; (2) two-layer tree building and RGBD images encoding (including RGBDs encoding in SRONet and the RGBs encoding in $\mathcal{F}_b$); and (3) ray querying in SRONet and neural blending-based ray upsampling. Finally, the post-processing and IO operations are used to reshape the color vectors of the emitted rays into images for display. We allocate half of the workload of each group to one GPU for parallel inference acceleration.

As illustrated in Fig. 8, GPU-0 handles the RGBD data of views 1 and 3, while GPU-1 handles those of views 2 and 4. Each $\mathcal{F}_d$ in the GPU predicts two refined depths, all of which are sent to the CPU for data synchronization. The CPU then sends the four reduced depth maps back to two GPUs for the two-layer tree construction. The interaction between the CPU and GPUs for image encoding performs in the same way as $\mathcal{F}_d$. We allocate half of the total rays (*i.e.*, $512^2/2$) to each GPU, which are fed into SRONet for ray querying and subsequent upsampling. Moreover, we utilize the surface rendering scheme to accelerate ray querying. After ray synchronization, we obtain the color vector of 4×512^2 logits, which is reshaped to an RGB image in $1K$ resolution as the final rendering result. We also use TensorRT with half-precision to accelerate $\mathcal{F}_d$, our SRONet, and the neural blending module. Besides, we adopt the fully fused scheme [Müller et al. 2021] to accelerate all the implicit functions $f_i (i = 1, ..., 6)$ and the hydra attention operation $\mathcal{H}$. Tab. 1 reports the time cost of each main operation in SAILOR. The accelerated SAILOR can finally render the free-view video in 1K resolution at around 10 fps. When the camera-target distance in the novel view is

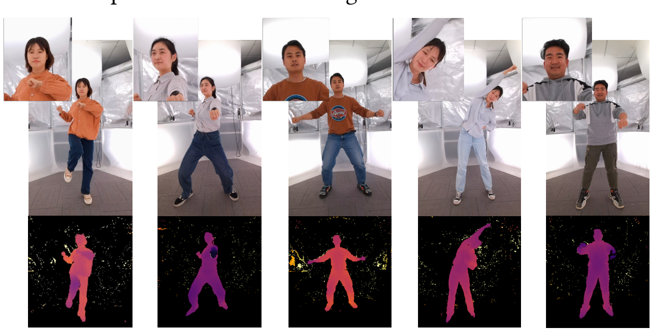

Fig. 10. Some RGB-D examples provided in our *real-captured human dataset*. Our dataset involves a variety of human subjects wearing daily clothing and performing different actions, with clear facial expressions captured.

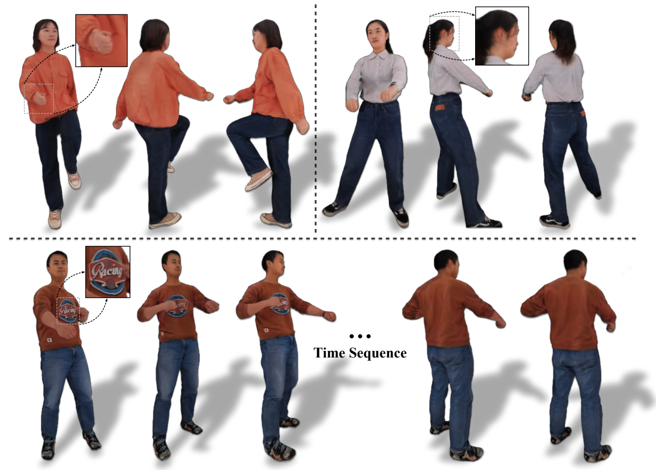

Fig. 11. Full-body rendering results of SAILOR on our dataset. Our dataset can be used to evaluate the novel-view rendering accuracy of a performer at a single timestamp (upper row) and over a time period (bottom row).

less than 80% of the average distance in the input views, exceeding the original sampling rate, we will perform bilinear interpolation to obtain the zoomed rendering results.

Fig. 9 shows the preprocessing steps (upper row) and two online examples (bottom row). Given 4-view captured RGBD images, our system performs undistortion (~1.5ms), background-matting (~4ms using TensorRT), image deformation (~1.2ms, including cropping, padding, and rotation), and depth-to-color alignment (~0.13ms) with acceleration to obtain the RGBD inputs for SAILOR.

4 OUR DATASET

We construct a *real-captured human dataset*, consisting of 160k+ frames of multi-view RGBD dynamic human motions, captured by Azure Kinect-V4 from 40 performers (20 female and 20 male actors). Each actor performed approximately 4,000 frames of action, wearing daily clothing. Typical actions are listed in Fig. 13.

The dataset for each performer contains captured RGBD sequences in 8 views, the pre-calibrated camera internal and external parameters, and the foreground segmentation RGB images (produced by background-matting-v2 [Lin et al. 2021]). The resolutions of the captured RGB and depth data are $2,560 \times 1,440$ and $1,024 \times 1,024$, respectively. For novel-view rendering evaluation, we use RGBD images of 4 fixed perspective views (the interval between two adjacent views is 90 degrees, and the indexes of cameras are 0,4,6,7, respectively) as inputs. RGBD images of the other four views (*i.e.*, indexes of 1,2,3,5) are used to evaluate rendering quality.

Our dataset contains various actions, diverse facial expressions, and complex geometries. Fig. 10 shows some examples. Our dataset can be considered a challenging human performance capture benchmark for evaluating SAILOR and other rendering methods. Fig. 11 shows some rendering results from SAILOR on this dataset.

5 RESULTS

Training and Evaluation. We train and evaluate our method using the public available *THuman*2.0 [Yu et al. 2021b] dataset, which contains 500 high-quality 3D human scans. We split the dataset into

Methods	Mesh		Normal	
	P2S$\times10^{-2}\downarrow$	**Chamfer**$\times10^{-2}\downarrow$	$\boldsymbol{L2}\times10^{-1}\downarrow$	**Cosine**$\times10^{-3}\downarrow$
PIFuHD	1.7268	1.7423	0.512	1.576
StereoPIFu	0.5832	0.5425	0.328	1.193
IPNet	0.8563	0.7247	0.196	0.751
PIFu(RGBD)	0.3246	0.3055	0.207	0.816
GTPIFu	0.2733	**0.2572**	**0.188**	0.676
Ours	**0.2695**	0.2739	0.241	**0.507**

Table 2. Geometric comparisons on the *THuman2.0* dataset [Yu et al. 2021b], between our reconstruction results and those produced by PIFuHD [Saito et al. 2020], StereoPIFu [Hong et al. 2021], PIFu(RGBD) [Saito et al. 2019], IPNet [Bhatnagar et al. 2020] and GTPIFu [Dong et al. 2022]. The best and second best results are marked in **bold** and underline, respectively.

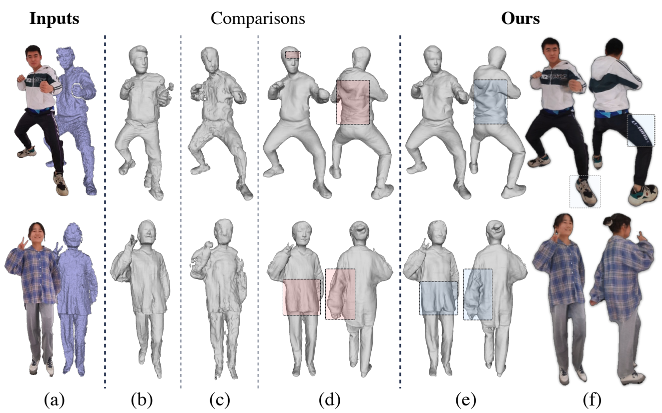

Fig. 12. Visual comparisons between our reconstruction results and those produced by existing reconstruction methods. One of four input RGBD images (a), results of PIFuHD [Saito et al. 2020] (b), IPNet [Bhatnagar et al. 2020] (c), GTPIFu [Dong et al. 2022] (d), our reconstruction results (e), and our novel-view rendering results (f) are shown, respectively. Our results contain more realistic details (*e.g.*, boxed regions).

training and test sets with a ratio of 4 : 1. For the input raw depth maps, we follow [Fankhauser et al. 2015] to synthesize the sensor noise on $\mathbf{D}_{gt}$ (See the supplemental for details) to produce noisy depth $\mathbf{D}$. In addition, we also evaluate the rendering performance of our method on our real-captured dataset.

5.1 Comparisons of Geometry

We compare our method to five state-of-the-art human reconstruction methods (with available codes), including the PIFuHD [Saito et al. 2020], StereoPIFu [Hong et al. 2021], PIFu(RGBD) [Saito et al. 2019], IPNet [Bhatnagar et al. 2020] and GTPIFu [Dong et al. 2022] on our test set. Since PIFuHD and StereoPIFu do not provide training codes, we directly use their pre-trained models for comparisons. The other three methods are re-trained using our data. We use Point-to-Surface (P2S) distance(cm), Chamfer distance(cm), $L2$ ($1e^{-1}$), and Cosine distance ($1e^{-3}$) as metrics.

Quantitative Comparison. Tab. 2 reports the comparison results between our method and existing approaches. Among the compared methods, PIFuHD [Saito et al. 2020] and StereoPIFu [Hong et al. 2021] reconstruct the human body from monocular and dual RGB images, respectively. The unstable reconstruction process due to a lack of geometry and view information may degrade their performance. IPNet [Bhatnagar et al. 2020] leverages the point cloud, while PIFu(RGBD) [Saito et al. 2019] and GTPIFu [Dong et al. 2022] utilize depth information to model geometry information, which result in more stable reconstruction and higher performance. By incorporating depth denoising into our rendering model, our method achieves comparable reconstruction performance to the state-of-the-art reconstruction method GTPIFu [Dong et al. 2022]: SAILOR is better in terms of P2S and Cosine metrics while GTPIFu achieves the higher performance under the Chamfer and L_2 metrics. In addition, we compare SAILOR to DiffuStereo [Shao et al. 2022c] on their released demo data (0001 and 0029 data of the Thuman2.0 dataset), as their training code is not available. Given the same 4-view data as inputs, The Chamfer/ P2S/L2 distances of ours are 0.3120/0.2976/0.0030, which is better than those of DiffuStereo (0.4612/0.4197/0.0044).

Last, we report comparisons on the real-captured examples (Fig. 13). The average L1(cm) distances to depth maps for 4 holdout views are 7.040(PIFuHD), 6.385(IPNet), 0.9051(GTPIFu), and 0.9040(Ours), showing that our method plays favorably against them.

Visual Comparison. Fig. 12 shows comparisons between our results and those of the PIFuHD [Saito et al. 2020], IPNet [Bhatnagar et al. 2020], and GTPIFu [Dong et al. 2022]. While PIFuHD [Saito et al. 2020] and IPNet [Bhatnagar et al. 2020] tend to produce obvious geometric artifacts under sparse views (Fig. 12(b,c)), the results of GTPIFu [Dong et al. 2022] (Fig. 12(d)) and ours (Fig. 12(e)) are more accurate, as we both exploit the robust geometric cues from the depth denoising. The face regions of GTPIFu [Dong et al. 2022] may contain more high-frequency details than ours, as they leverage another PIFu to model the face regions separately. However, this is computationally heavy. Our results contain more accurate details on some body regions (*e.g.*, wrinkles of clothes) than those of GTPIFu, since the local high-frequency body geometry information may be suppressed by the joint optimization of depth denoising and occupancy prediction in [Dong et al. 2022]. In contrast, the joint optimization of the occupancy and color fields in our SRONet exploits the ground-truth 3D and RGBD signals for capturing more high-frequency geometric details.

5.2 Comparisons of Rendering

We compare SAILOR to six state-of-the-art generalizable rendering methods (with available codes), including PixelNeRF [Yu et al. 2021a], IBRNet [Wang et al. 2021a], MPSNeRF [Gao et al. 2022], NHP [Kwon et al. 2021], KeypointNeRF [Mihajlovic et al. 2022], NPBG++ [Rakhimov et al. 2022] and PIFu(RGBD) [Saito et al. 2019]. For fair comparisons, we either re-train (unavailable pre-trained weights) or fine-tune (available pre-trained weights) these methods on the training set of THuman2.0 dataset [Yu et al. 2021b]. PSNR, SSIM, and MAE are used to measure the rendering accuracy. We also report the Learned Perceptual Image Patch Similarity (LPIPS) [Richard Zhang and Wan 2018] for reference. We generate noise of 5 different degrees (*i.e.*, 0.25cm, 0.5cm, 1.0cm, 1.5cm, and 2.0cm of Gaussian standard deviation) on the depth maps of the 3D meshes to evaluate our method against NPBG++ [Rakhimov et al. 2022] and PIFu(RGBD) [Saito et al. 2019].

Quantitative Comparison. Tab. 3 reports the rendering comparisons on the THuman2.0 dataset [Yu et al. 2021b] (upper part) and our dataset (bottom part). From Tab. 3, we can see that our method generally outperforms existing rendering methods regarding all

Fig. 13. Visualization of rendering comparisons on our real-captured dataset (row 1-5) and the THuman2.0 dataset [Yu et al. 2021b] (row 6), between our results and those of PixelNeRF [Yu et al. 2021a], IBRNet [Wang et al. 2021a], MPSNeRF [Gao et al. 2022], KeypointNeRF [Mihajlovic et al. 2022], NPBG++ [Rakhimov et al. 2022] and PIFu(RGBD) [Saito et al. 2019].

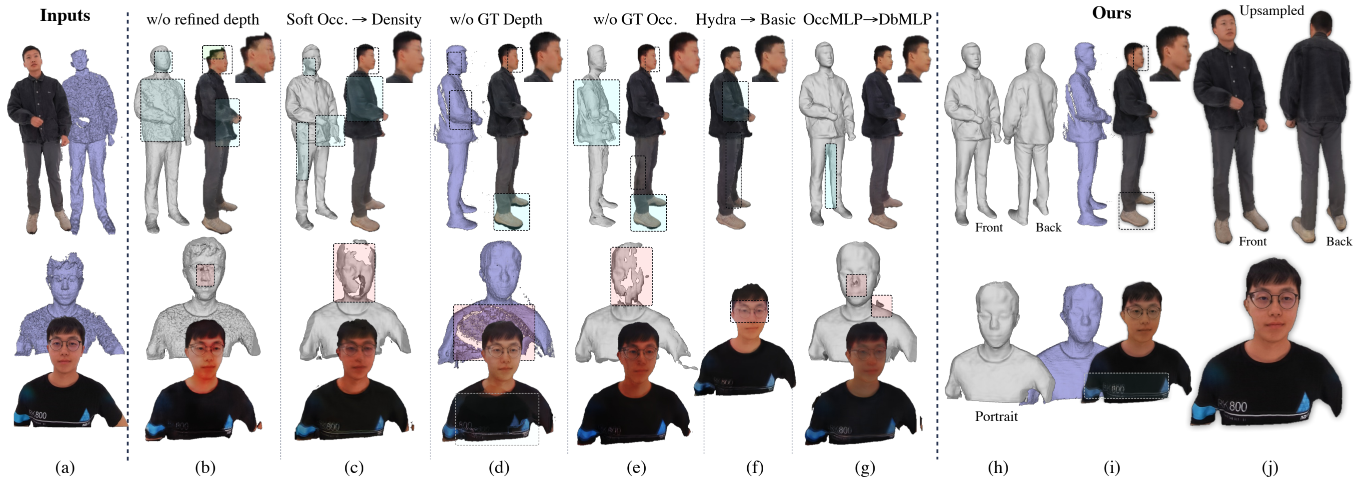

Fig. 14. Ablation Study on our real captured dataset. 1 of 4 input RGBD views (a). Reconstruction and rendering results of different ablated versions (b to g). Our reconstruction results (h). Our rendering results without upsampling (i). Our final rendering results (j).

Models	Avg Time (s)↓	*THuman2.0* [Yu et al. 2021b] Dataset			
		PSNR ↑	**SSIM** ↑	**LPIPS** $\times10^{-1}$ ↓	**MAE**$\times10^{-2}$ ↓
PixelNeRF	≈ 390.0	30.215	0.938	1.179	0.865
IBRNet	≈ 25.7	34.469	0.963	0.742	0.497
MPSNeRF	≈ 32.2	30.317	0.945	0.866	0.754
NHP	≈ 102.5	31.488	0.957	0.851	0.647
KeypointNeRF	≈ 52.3	31.590	0.953	0.746	0.658
NPBG++	≈ 5.5	32.136	0.962	0.558	0.533
PIFu(RGBD)	≈ 8.5	33.296	0.967	**0.270**	0.543
Ours	≈ 0.2	**34.882**	**0.969**	0.354	**0.392**

Metrics	Methods (evaluation on our *Real-captured* Dataset)						
	PixelNeRF	IBRNet	MPSNeRF	NHP	NPBG++	PIFu(RGBD)	**Ours**
PSNR↑	23.876	25.946	25.172	24.568	25.949	28.254	**29.969**
SSIM↑	0.908	0.929	0.925	0.933	0.924	0.950	**0.962**
LPIPS↓	0.146	0.110	0.110	0.108	0.0809	0.0428	**0.0359**

Table 3. Comparisons of rendering results on the THuman2.0 dataset and our real-captured dataset, produced by our method and existing generalizable methods. We re-train or fine-tune all competing methods on our training dataset for a fair evaluation. The rendering time (for images in 1k resolution) is calculated using a **single** RTX 3090 GPU as their public codes do not include a multi-card accelerated rendering manner. The best and second best results are marked in **bold** and underline, respectively. Refer to the supplemental for detailed comparisons of the real-captured data on 10 independent performers.

three objective metrics (*i.e.*, PSNR, SSIM, and MAE) on the Thuman2.0 dataset, and along with LPIPS on the real-captured dataset. Tab. 3 also summarizes the average rendering time of existing methods and ours to obtain an image of 1*K* resolution using a single RTX 3090 GPU. Our method runs much faster than existing methods. We also compare our method with GTPIFu [Dong et al. 2022] (about 31s using MVS-Texturing [Waechter et al. 2014] to generate a textured mesh), on our real-captured data in Fig. 12. The PSNR/SSIM/LPIPS of GTPIFu and Ours are 28.610/0.922/0.0289 and 28.444/0.926/0.0388, respectively. For our data in Fig. 13 (1-5 rows), the values are 25.411/0.920/0.105 (KeypointNeRF [Mihajlovic et al. 2022]), while ours are 29.355/0.958/0.0383. These experiments verify that our method is able to produce high-quality novel view rendering results for performers with diverse actions and clothing.

Visual Comparison. Fig. 13 visualizes the qualitative comparisons on both our real-captured dataset (first five rows) and the THuman2.0 dataset [Yu et al. 2021b] (last row). We can see that PixelNeRF [Yu et al. 2021a] tends to produce topological errors, surface distortions, blurry or wrong textures, and sometimes background noise, as colorimetric constraints are far from enough for performance capture under sparse views. IBRNet [Wang et al. 2021a] performs slightly better, especially near the input views, but still suffers from the same problems as PixelNeRF [Yu et al. 2021a]. KeypointNeRF [Mihajlovic et al. 2022] encodes relative spatial 3D information via sparse 3D keypoints, and MPSNeRF [Gao et al. 2022] incorporates SMPL [Loper et al. 2015] as topological constraints, similar to NHP [Kwon et al. 2021]. However, their results typically have incorrect shapes and textures in regions occluded in the input views and regions with large/complex motions. NPBG++ [Rakhimov et al. 2022] leverages the point cloud as the 3D human representation, and their results typically suffer from missing parts and color distortion due to the noisy point clouds. PIFu(RGBD) [Saito et al. 2019] performs relatively better among these methods, as they combine depths with surface fields for reconstruction. However, due to rendering from the textured mesh, inaccurate geometries can lead to rendering noise. In addition, they cannot handle view-dependent effects due to no view direction encodings. In contrast, our rendering results have better edges (shapes) and high-quality texture details.

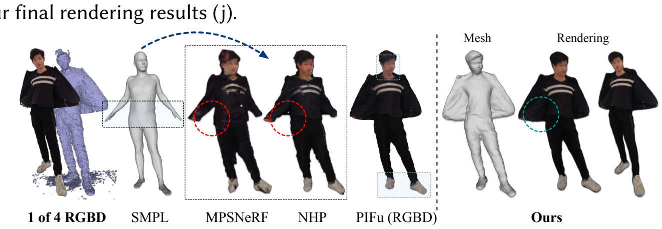

Fig. 15. Our method can render the geometric parts of the clothing correctly, while SMPL-based methods MPSNeRF [Gao et al. 2022] and NHP [Kwon et al. 2021] tend to fail, and PIFu(RGBD) [Saito et al. 2019] tends to render unrealistic details especially in the facial region due to the textured mesh.

Fig. 15 further evaluates our method in handling topology changes of clothing. Since SMPL [Loper et al. 2015] models only represent the naked human bodies, methods (*e.g.*, MPSNeRF [Gao et al. 2022] and NHP [Kwon et al. 2021]) based on SMPL representation cannot reconstruct such topological changes (*e.g.*, red circled regions).

Models	*THuman2.0* [Yu et al. 2021b] Dataset			
	PSNR ↑	**SSIM** ↑	**LPIPS** $\times 10^{-1}$ ↓	**MAE**$\times 10^{-2}$ ↓
w/o GT Depth	33.171	0.961	0.348	0.495
w/o GT Occ.	32.871	0.959	0.356	0.512
w/o Denoised Depth	32.588	0.955	0.425	0.548
Soft Occ. → Density	32.441	0.961	0.382	0.494
OccMLP → DbMLP	33.499	0.962	0.351	0.489
w/o Upsampling	34.865	0.967	**0.291**	0.467
Ours	**34.882**	**0.969**	0.354	**0.392**
Models	*Our Real Captured Dataset*			
	PSNR ↑	**SSIM** ↑	**LPIPS** $\times 10^{-1}$ ↓	**MAE**$\times 10^{-2}$ ↓
w/o GT Depth	30.225	0.955	0.459	0.647
w/o GT Occ.	29.637	0.950	0.553	0.712
w/o Denoised Depth	28.489	0.953	0.549	0.712
Soft Occ. → Density	29.540	0.950	0.622	0.655
OccMLP → DbMLP	30.085	0.947	0.675	0.705
w/o Upsampling	29.890	0.960	0.479	0.674
Ours	**30.228**	**0.968**	**0.454**	**0.634**

Table 4. Ablation Study on the *THuman*2.0 dataset [Yu et al. 2021b] (upper part) and our real captured data (lower part). The best and second best results are marked in **bold** and underline, respectively.

In addition, PIFu(RGBD) [Saito et al. 2019] takes advantage of the raw depth and may somewhat model clothing changes. However, it tends to generate undesirable results under novel views. In contrast, our method exploits the denoised depths with a hybrid representation of surface and color fields, thus effectively reconstructing and rendering the clothing (*e.g.*, clothing in the bottom part of Fig. 9).

5.3 Ablation Study

We conduct ablation studies on both the *THuman*2.0 dataset [Yu et al. 2021b] (upper part) and our dataset (lower part) in Tab. 4.

Depth & Geometric Supervisions. We first remove the depth and occupancy supervisions (denoted as "*w/o* GT Depth" and "*w/o* GT Occ.", respectively) to train our SRONet. The first two rows in both sub-tables of Tab. 4 show that the performance degrades regarding all metrics, especially "*w/o* GT Occ.". Fig. 14(e,d) shows that the predicted depths and the reconstructed 3D models contain obvious errors, resulting in low-quality rendering results. These results indicate that OCC. supervision contributes to our results effectively by providing global geometric constraints.

Depth Denoising Module: $\mathcal{F}_d$. We then remove the $\mathcal{F}_d$ and use the raw depths (denoted as "*w/o* Denoised Depth") to train SRONet. Here, we use RGBD images instead of only depth maps as the input of our OCCNet. Similarly, performance in all metrics drops without $\mathcal{F}_d$ in both datasets (3rd rows). From Fig. 14(b), we can see that the holes in the depth can result in incomplete rendering results (*e.g.*, the head region). Comparing (b) to (h) demonstrates that $\mathcal{F}_d$ can complete the holes and remove the noise in the raw depths, which helps render high-quality images.

Hybrid Representation: $\mathcal{F}_o$ & $\mathcal{F}_c$. We conduct two experiments to verify the superiority of our hybrid representation.

(1) We replace the soft occupancy field of our OCCNet with the density field, and replace the color blending function with the volume rendering function in NeRF (denoted as "Soft Occ. → Density"). In such a way, the hybrid representation degrades back to NeRF, and the results of both datasets show that the performance degrades accordingly (4th rows). Fig. 14(c) shows that geometric and texture errors both occur in the reconstructed and rendering results, and the denoised depths cannot effectively correct these errors.

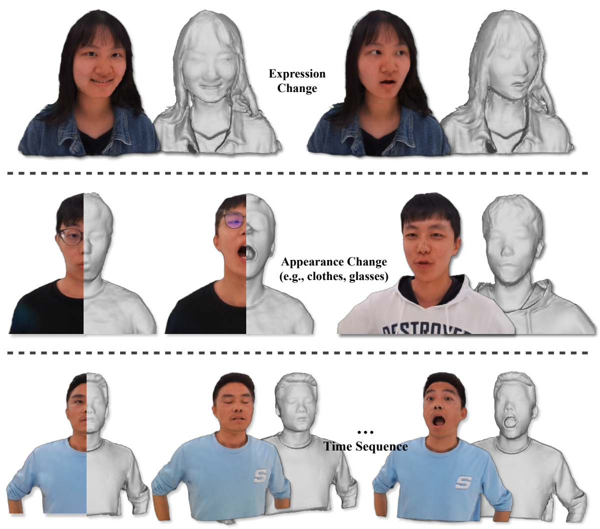

Fig. 16. Novel-view portrait reconstruction and rendering results of SAILOR without re-training or fine-tuning. Our method can handle sudden expression changes and complex geometries such as long hair (1st row), and significant appearance changes (2nd row, such as wearing glasses and different clothes), and can track the subject with diverse expressions over a long time period (3rd row).

Models	PSNR ↑	SSIM ↑	LPIPS $\times 10^{-1}$ ↓	MAE$\times 10^{-2}$ ↓	Time↓
Volume →Surface	**34.889**	0.964	0.464	0.430	≈ **112ms**
Ours	34.882	**0.969**	**0.354**	**0.392**	≈ 271ms

Table 5. Comparisons between the volume rendering and surface rendering on the THuman2.0 dataset [Yu et al. 2021b]. "Volume→surface" indicates replacing the color blending (Eq. 4) with the surface rendering in our SRONet. The best results are marked in **bold**. The reported time (Ours) corresponds to the process of *Ray Querying* in Tab. 1.

(2) We investigate the way of DoubleField [Shao et al. 2022b] by making our OCCNet predict both occupancy and density values, for reconstruction and rendering, respectively (denoted as "OccMLP→DbMLP"). In this case, the occupancy and radiance fields are combined in an implicit manner by sharing the features in OCCNet. However, quantitative results (5th rows) show that the performance is inferior to ours. By observing Fig. 14(g) we find that this strategy suffers from local geometry errors (*e.g.*, floating surfaces and holes in the reconstructed portrait) and produce color artifacts (*e.g.*, greenish colors on the rendered face). These results suggest that such an implicit integration of occupancy and radiance fields may not be ideal, as the density field may affect the surface field negatively (shape-color ambiguities) in the early stage. In contrast, by discarding the density field and explicitly combining the occupancy field with the color field, SAILOR is able to avoid this issue.

Appearance Upsampling. We now remove the ray upsampling and neural blending $\mathcal{F}_b$ from SAILOR (denoted as "*w/o* Upsampling"). Tab. 4 (6th rows) show that performance for almost all metrics drops without the $\mathcal{F}_b$, although the LPIPS metric is slightly improved on the *THuman*2.0 dataset [Yu et al. 2021b]. Considering that LPIPS is a subjective metric, these results demonstrate that $\mathcal{F}_b$ increases the spatial resolution and improves the rendering quality. Fig. 14(i) shows that $\mathcal{F}_b$ is able to correct the colors by integrating colors of neighboring views, especially for portrait rendering.

Rendering Scheme. We note that surface rendering may result in a fast ray querying in SRONet, as only the surface points calculated using the depths $\hat{D}(\mathrm{I})$ are involved in ColorNet, which is suitable for rendering acceleration. We investigate the rendering scheme by replacing the volume rendering in our SRONet with surface rendering, denoted as "Volume→Surface". Results on the *THuman*2.0 dataset [Yu et al. 2021b] are reported in Tab. 5. While we can see that volume rendering and surface rendering yield close performance in terms of PSNR and SSIM, volume rendering performs better when measured with the LPIPS and MAE metrics. This provides two choices, *i.e.*, users may switch to using surface rendering for further acceleration or adopting the volume rendering in our SRONet for a more accurate rendering result.

5.4 Generalization in New Settings

New Camera Setting. Our method does not require the cameras to be put exactly as the setting (refer to sec 1.4 in the supplemental) of our capture system. To verify this, we test our method on the real-captured data in Fig. 13 using a single input view (front view). The PSNR/SSIM/LPIPS values of two adjacent test views (45 degrees) are 25.356/0.939/0.0588. Moreover, the ablation studies of sensor numbers are included in our supplemental.

New Clothing Types. We show that our method can handle scenarios where clothing topology changes (*e.g.*, hats, loose pants, putting on and taking off coats, in Fig. 9), long hair, as well as handling accessories such as glasses and phones to some extent. Please refer to our provided third-person videos for more details.

Portrait Reconstruction and Rendering. Fig. 16 shows our novel-view portrait rendering and reconstruction results of three performers using SAILOR. Here, we use 3 Azure Kinect-V4 sensors for capturing RGBDs. It shows that SAILOR can handle some sudden expression changes and complex geometries such as long hair (1st row), appearance changes such as wearing glasses and different clothes (2nd row), and can track the subject with diverse expressions over a long time period (3rd row).

6 CONCLUSION

In this paper, we have proposed a novel method (SAILOR) for creating high-quality human free-view videos from very sparse RGBD videos with low latency. The core of SAILOR is a depth-conditioned hybrid representation of PIFu and NeRF, capable of preserving locally accurate geometry and producing vivid view-dependent textures. We have proposed a novel network (SRONet) for this hybrid representation. In addition, we have designed a neural blending-based ray interpolation scheme, a tree-based voxel denoising scheme, and a parallel computing pipeline for acceleration. To evaluate rendering performance, we have constructed a real-captured RGBD benchmark of 40 performers. Experiments show that SAILOR can handle unseen performers without fine-tuning, outperform existing human reconstruction and performance capture methods, and can be applied to human portrait reconstruction and rendering.

Our method does have some limitations. First, our rendering results may exhibit temporal color/illumination flickering, overlay artifacts and fail to capture fine details (*e.g.*, hands and fingers), due to inaccurate matting, camera calibration, and a lack of temporal modeling. Second, our dataset capture assumes a uniform illumination. Modeling temporal constraints and complex lighting can be interesting for future research.

ACKNOWLEDGMENTS

We thank all the anonymous reviewers for their professional and constructive comments. This work was partially supported by a GRF grant from RGC of Hong Kong (Ref.: 11205620). Weiwei Xu is supported by NSFC (No. 61732016). Qilin Sun would like to acknowledge the support from NSFC (No. 62302423). Besides, this work was supported by Ant Group, and this paper is supported by the Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

REFERENCES

Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. 2019. Tex2Shape: Detailed Full Human Body Geometry from a Single Image. In *Int. Conf. Comput. Vis.*

Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. 2020. Combining implicit function learning and parametric models for 3d human reconstruction. In *Eur. Conf. Comput. Vis.*

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. 2023. Hydra attention: Efficient attention with many heads. In *Eur. Conf. Comput. Vis.*

Yukang Cao, Guanying Chen, Kai Han, Wenqi Yang, and Kwan-Yee K Wong. 2022. JIFF: Jointly-aligned Implicit Face Function for High Quality Single View Clothed Human Reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Kennard Chan, Guosheng Lin, Haiyu Zhao, and Weisi Lin. 2022a. S-PIFu: Integrating Parametric Human Models with PIFu for Single-view Clothed Human Reconstruction. In *Adv. Neural Inform. Process. Syst.*

Kennard Yanting Chan, Guosheng Lin, Haiyu Zhao, and Weisi Lin. 2022b. IntegratedPIFu: Integrated Pixel Aligned Implicit Function Fornbsp;Single-View Human Reconstruction. In *Eur. Conf. Comput. Vis.*

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021a. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Int. Conf. Comput. Vis.*

Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. 2021b. Animatable Neural Radiance Fields from Monocular RGB Videos. arXiv:2106.13629

Shenchang Eric Chen and Lance Williams. 1993. View interpolation for image synthesis. In *Proc. of SIGGRAPH.*

Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. High-Quality Streamable Free-Viewpoint Video. *ACM Trans. Graph.* (2015).

Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. 2008. Performance capture from sparse multi-view video. In *ACM SIGGRAPH.*

Paul Debevec, Yizhou Yu, and George Borshukov. 1998. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics.*

Zheng Dong, Ke Xu, Ziheng Duan, Hujun Bao, Weiwei Xu, and Rynson Lau. 2022. Geometry-aware Two-scale PIFu Representation for Human Reconstruction. In *Adv. Neural Inform. Process. Syst.*

Mingsong Dou, Philip L. Davidson, S. Fanello, S. Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. 2017. Motion2fusion: real-time volumetric performance capture. *ACM Trans. Graph.* (2017).

Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. 2016. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.* (2016).

Péter Fankhauser, Michael Bloesch, Diego Rodriguez, Ralf Kaestner, Marco Hutter, and Roland Siegwart. 2015. Kinect v2 for mobile robot navigation: Evaluation and modeling. In *ICAR.*

Qiao Feng, Yebin Liu, Yu-Kun Lai, Jingyu Yang, and Kun Li. 2022. FOF: Learning Fourier Occupancy Field for Monocular Real-time Human Reconstruction. In *Adv. Neural Inform. Process. Syst.*

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Xiangjun Gao, Jiao long Yang, Jongyoo Kim, Sida Peng, Zicheng Liu, and Xin Tong. 2022. MPS-NeRF: Generalizable 3D Human Rendering from Multiview Images. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022).

Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. 2019. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph.* (2019).

Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2021. Real-time deep dynamic characters. *ACM Trans. Graph.* (2021).

Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2019. Livecap: Real-time human performance capture from monocular video. *ACM Trans. Graph.* (2019).

Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. 2020. Deepcap: Monocular human performance capture using weak supervision. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.* (2018).

Yang Hong, Juyong Zhang, Boyi Jiang, Yudong Guo, Ligang Liu, and Hujun Bao. 2021. StereoPIFu: Depth Aware Clothed Human Digitization via Stereo Vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. 2023. SHERF: Generalizable Human NeRF from a Single Image. arXiv:2303.12791

Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. 2023. HumanRF: High-Fidelity Neural Radiance Fields for Humans in Motion. *ACM Trans. Graph.* (2023).

Zhang Jiakai, Liu Xinhang, Ye Xinyi, Zhao Fuqiang, Zhang Yanshun, Wu Minye, Zhang Yingliang, Xu Lan, and Yu Jingyi. 2021. Editable Free-Viewpoint Video using a Layered Neural Representation. In *Proc. of SIGGRAPH.*

Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. 2022. NeuMan: Neural Human Radiance Field from a Single Video. In *Eur. Conf. Comput. Vis.*

Hanbyul Joo, Tomas Simon, and Yaser Sheikh. 2018. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Jaehyeok Kim, Dongyoon Wee, and Dan Xu. 2023. You Only Train Once: Multi-Identity Free-Viewpoint Neural Human Rendering from Monocular Videos. arXiv:2303.05835

Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. 2020. Vibe: Video inference for human body pose and shape estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. 2021. Neural human performer: Learning generalizable radiance fields for human performance rendering. In *Adv. Neural Inform. Process. Syst.*

J. P. Lewis, Matt Cordner, and Nickson Fong. 2000. Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *Proc. of SIGGRAPH.*

Ruilong Li, Yuliang Xiu, Shunsuke Saito, Zeng Huang, Kyle Olszewski, and Hao Li. 2020a. Monocular real-time volumetric performance capture. In *Eur. Conf. Comput. Vis.*

Yue Li, Marc Habermann, Bernhard Thomaszewski, Stelian Coros, Thabo Beeler, and Christian Theobalt. 2021. Deep Physics-aware Inference of Cloth Deformation for Monocular Human Performance Capture. In *3DV.*

Zhe Li, Tao Yu, Chuanyu Pan, Zerong Zheng, and Yebin Liu. 2020b. Robust 3d self-portraits in seconds. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2022. Efficient Neural Radiance Fields with Learned Depth-Guided Sampling. In *ACM SIGGRAPH Asia.*

Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. 2021. Real-time high-resolution background matting. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural Sparse Voxel Fields. *NeurIPS.*

Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. 2021. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Trans. Graph.* (2021).

Yebin Liu, Qionghai Dai, and Wenli Xu. 2009. A point-cloud-based multiview stereo algorithm for free-viewpoint video. *IEEE Trans. Vis. Comput. Graph.* (2009).

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of Volumetric Primitives for Efficient Neural Rendering. *ACM Trans. Graph.* (2021).

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael Black. 2015. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.* (2015).

Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. 2018. LookinGood: Enhancing Performance Capture with Real-Time Neural Re-Rendering. *ACM Trans. Graph.* (2018).

Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. 2000. Image-Based Visual Hulls. In *Proc. of SIGGRAPH.*

Marko Mihajlovic, Aayush Bansal, Michael Zollhoefer, Siyu Tang, and Shunsuke Saito. 2022. KeypointNeRF: Generalizing Image-based Volumetric Avatars using Relative Spatial Encoding of Keypoints. In *Eur. Conf. Comput. Vis.*

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Eur. Conf. Comput. Vis.*

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time Neural Radiance Caching for Path Tracing. *ACM Trans. Graph.* (2021).

Richard A Newcombe, Dieter Fox, and Steven M Seitz. 2015a. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Richard A Newcombe, Dieter Fox, and Steven M Seitz. 2015b. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR.*

Michael Oechsle, Songyou Peng, and Andreas Geiger. 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Int. Conf. Comput. Vis.*

Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Annual Symposium on User Interface Software and Technology.*

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021a. Nerfies: Deformable Neural Radiance Fields. In *Int. Conf. Comput. Vis.*

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. 2021b. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ACM Trans. Graph.* (2021).

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body From a Single Image. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Bo Peng, Jun Hu, Jingtao Zhou, Xuan Gao, and Juyong Zhang. 2023. IntrinsicNGP: Intrinsic Coordinate based Hash Encoding for Human NeRF. arXiv:2302.14683

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable neural radiance fields for modeling dynamic human bodies. In *Int. Conf. Comput. Vis.*

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Adv. Neural Inform. Process. Syst.*

Ruslan Rakhimov, Andrei-Timotei Ardelean, Victor Lempitsky, and Evgeny Burnaev. 2022. NPBG++: Accelerating Neural Point-Based Graphics. In *IEEE Conf. Comput. Vis. Pattern Recog.*

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

Alexei A Efros Eli Shechtman Richard Zhang, Phillip Isola and Oliver Wan. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI.*

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. 2019. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Int. Conf. Comput. Vis.*

Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Ruizhi Shao, Liliang Chen, Zerong Zheng, Hongwen Zhang, Yuxiang Zhang, Han Huang, Yandong Guo, and Yebin Liu. 2022a. FloRen: Real-Time High-Quality Human Performance Rendering via Appearance Flow Using Sparse RGB Cameras. In *ACM SIGGRAPH Asia.*

Ruizhi Shao, Hongwen Zhang, He Zhang, Mingjia Chen, Yanpei Cao, Tao Yu, and Yebin Liu. 2022b. DoubleField: Bridging the Neural Surface and Radiance Fields for High-fidelity Human Reconstruction and Rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Ruizhi Shao, Zerong Zheng, Hongwen Zhang, Jingxiang Sun, and Yebin Liu. 2022c. DiffuStereo: High Quality Human Reconstruction via Diffusion-based Stereo Using Sparse Cameras. In *Eur. Conf. Comput. Vis.*

Shih-Yang Su, Timur Bagautdinov, and Helge Rhodin. 2022. DANBO: Disentangled Articulated Neural Body Representations via Graph Neural Networks. In *Eur. Conf. Comput. Vis.*

Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. 2021. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Adv. Neural Inform. Process. Syst.* (2021).

Zhuo Su, Lan Xu, Zerong Zheng, Tao Yu, Yebin Liu, and Lu Fang. 2020. RobustFusion: Human Volumetric Capture with Data-Driven Visual Cues Using a RGBD Camera. In *Eur. Conf. Comput. Vis.*

Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Int. Conf. Comput. Vis.*

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*

Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. 2008. Articulated Mesh Animation from Multi-View Silhouettes. *ACM Trans. Graph.* (2008).

Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik. 2009. Dynamic shape capture using multi-view photometric stereo. In *ACM SIGGRAPH Asia*.

Michael Waechter, Nils Moehrle, and Michael Goesele. 2014. Let There Be Color! — Large-Scale Texturing of 3D Reconstructions. In *Eur. Conf. Comput. Vis.*

Liao Wang, Ziyu Wang, Pei Lin, Yuheng Jiang, Xin Suo, Minye Wu, Lan Xu, and Jingyi Yu. 2021b. IButter: Neural Interactive Bullet Time Generator for Human Free-Viewpoint Rendering. In *ACM Int. Conf. Multimedia*.

Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. 2022. Fourier PlenOctrees for Dynamic Radiance Field Rendering in Real-Time. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. 2021a. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. 2022. HumanNeRF: Free-Viewpoint Rendering of Moving People From Monocular Video. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. 2020. Multi-view neural human rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2021. Space-time neural irradiance fields for free-viewpoint video. In *IEEE Conf. Comput. Vis. Pattern Recog.*

D. Xiang, F. Prada, C. Wu, and J. Hodgins. 2020. MonoClothCap: Towards Temporally Coherent Clothing Capture from Monocular RGB Video. In *3DV*.

Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. 2022. ICON: Implicit Clothed humans Obtained from Normals. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. 2018. MonoPerfCap: Human Performance Capture From Monocular Video. *ACM Trans. Graph.* (2018).

Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. 2019. Differentiable Surface Splatting for Point-Based Geometry Processing. *ACM Trans. Graph.* (2019).

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021a. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. 2021b. Function4D: Real-time Human Volumetric Capture from Very Sparse Consumer RGBD Sensors. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. 2018. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Jiakai Zhang, Liao Wang, Xinhang Liu, Fuqiang Zhao, Minzhang Li, Haizhao Dai, Boyuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. 2022a. NeuVV: Neural Volumetric Videos with Immersive Rendering and Editing. arXiv:2202.06088

Yizhong Zhang, Jiaolong Yang, Zhen Liu, Ruicheng Wang, Guojun Chen, Xin Tong, and Baining Guo. 2022b. VirtualCube: An Immersive 3D Video Communication System. In *IEEE VR*.

Fuqiang Zhao, Yuheng Jiang, Kaixin Yao, Jiakai Zhang, Liao Wang, Haizhao Dai, Yuhui Zhong, Yingliang Zhang, Minye Wu, Lan Xu, and Jingyi Yu. 2022a. Human Performance Modeling and Rendering via Neural Animated Mesh. *ACM Trans. Graph.* (2022).

Fuqiang Zhao, Wei Yang, Jiakai Zhang, Pei Lin, Yingliang Zhang, Jingyi Yu, and Lan Xu. 2022b. HumanNeRF: Efficiently Generated Human Radiance Field From Sparse Inputs. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Hao Zhao, Jinsong Zhang, Yu-Kun Lai, Zerong Zheng, Yingdi Xie, Yebin Liu, and Kun Li. 2022c. High-Fidelity Human Avatars from a Single RGB Camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Zerong Zheng, Tao Yu, Yebin Liu, and Dai Qionghai. 2021. PaMIR: Parametric Model-Conditioned Implicit Representation for Image-based Human Reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021).

Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. 2019. Deephuman: 3D human reconstruction from a single image. In *Int. Conf. Comput. Vis.*

Hao Zhu, Xinxin Zuo, Haotian Yang, Sen Wang, Xun Cao, and Ruigang Yang. 2022. Detailed Avatar Recovery From Single Image. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022).