MonoNHR: Monocular Neural Human Renderer

Hongsuk Choi∗1,3 Gyeongsik Moon∗2 Kyoung Mu Lee1 Matthieu Armando3 Vincent Leroy3

1 Dept. of ECE & ASRI, Seoul National University, Korea
2 Meta Reality Labs Research
3 NAVER LABS Europe

redstonepo@gmail.com, mks0601@fb.com,
{matthieu.armando,vincent.leroy,gregory.rogez}@naverlabs.com, kyoungmu@snu.ac.kr

Abstract

Existing neural human rendering methods struggle with a single image input due to the lack of information in invisible areas and the depth ambiguity of pixels in visible areas. In this regard, we propose Monocular Neural Human Renderer (MonoNHR), a novel approach that renders robust free-viewpoint images of an arbitrary human given only a single image. MonoNHR is the first method that (i) renders human subjects never seen during training in a monocular setup, and (ii) is trained in a weakly-supervised manner without geometry supervision. First, we propose to disentangle 3D geometry and texture features and to condition the texture inference on the 3D geometry features. Second, we introduce a Mesh Inpainter module that inpaints the occluded parts exploiting human structural priors such as symmetry. Experiments on ZJU-MoCap, AIST and HUMBI datasets show that our approach significantly outperforms the recent methods adapted to the monocular case.

1. Introduction

Novel view synthesis from a single image is a very challenging problem, but has many potential applications, e.g. in AR/VR or smartphone-based social networking services. Markerless capture from RGB data has been widely studied as a tool to generate realistic free-viewpoint renderings of humans, but it often requires synchronized and calibrated multi-camera systems. We take a step towards monocular capture of people’s appearance and shape, and tackle novel view synthesis of a person observed from a single image, which extends concurrent works to a more general setting.

Neural human rendering methods, which aim to render people from arbitrary viewpoints, showed promising results for this task. These can generally be grouped into 2 main categories: those learning subject-specific Neural Radiance Fields (NeRF) [28] to represent the appearance of a particular human [23, 31, 32, 36, 42, 48] and approaches that estimate neural surface fields [40] using pixel-aligned image features [13, 34, 35, 40, 56]. The first ones require a large number of input images [23, 48] or multi-view video frames capturing the complete surface of the target [23, 31, 32, 48], while the others rely on a detailed geometric ground-truth during training, and thus, require expensive, therefore small-scale, 3D scans datasets, preventing generalization to unseen human poses and appearances. ARCH [15] and ARCH++ [12], that follow a different approach by learning an occupancy function in some canonical body space, also suffer from this problem. We summa-

Figure 1: Given a single image input of a human (leftmost), MonoNHR generates realistic renderings from novel viewpoints (right). Tested on unseen subjects from HUMBI [54].
rize these modalities in Table 1.

Interestingly, very recent NeRF-based methods, such as pixelNeRF [53], PVA [33], and NHP [18], showed that it is feasible to render free-viewpoint images of humans from a sparse set of views, while allowing generalization to arbitrary subjects. However, these methods are not designed to synthesize occluded surfaces, as they only render surfaces visible in the input views. Thus these methods struggle with monocular inputs, where more than half the surface of the observed person can be invisible to the camera, making the texture and geometry of the invisible parts largely ambiguous. Furthermore, a depth ambiguity remains inherent to monocular observations. In this paper, we claim that prior knowledge of the human appearance and shape, such as symmetry, color consistency between surfaces, and front-back coherence, should be better exploited for this task.

Based on these observations, we propose Monocular Neural Human Renderer (MonoNHR), a novel NeRF-based architecture that robustly renders free-viewpoint images of an arbitrary human given a single image of that person. We address the issues inherent to a monocular observation in two ways. First, we disentangle the features of 3D geometry and texture by extracting geometry-dedicated features. Different from neural surface field-based methods [34, 35], MonoNHR is trained only with multi-view images without ground-truth (GT) 3D scans. Since we do not consider explicit 3D geometry, extracting geometry-dedicated features is non-trivial. To do so, we design a geometry estimation branch, separated from the texture estimation fork, that is used solely for estimating the density of the radiance field. Second, we introduce Mesh Inpainter that operates on the SMPL [24] mesh estimated from the input image. It is used only during training, to encourage the backbone network to implicitly learn human priors. Please note that contrary to SMPL-texturing works [52], our method does not rely on this 3D surface and is able to render shapes that largely differ from the SMPL model.

We study the efficacy of MonoNHR on ZJU-MoCap [32], AIST [21, 44] and HUMBI [54] datasets. Experiments show that our method significantly outperforms recent NeRF-based methods on monocular images. To the best of our knowledge, MonoNHR is the first approach specifically designed for novel view synthesis of humans from a monocular image using neural radiance fields. Unlike previous works, it explicitly and effectively handles the many ambiguities inherent to monocular observations. Our contributions are summarized below:

- We present MonoNHR, a novel NeRF-based architecture that robustly renders free-viewpoint images of an arbitrary human from a monocular image. It pushes the boundaries of NeRF-based novel view synthesis research to a more general setting.

| method        | input      | supervision | unseen identity |
|---------------|------------|-------------|-----------------|
| NB [32], Ani-NeRF [31] | subject code | multi-view videos | x |
| PIFu [34,35], ARCH/ARCH++ [12,15] | monocular image | 3D scans | ✓ |
| NHP [18]      | multi-view image | multi-view images | ✓ |
| MonoNHR (Ours) | monocular image | multi-view images | ✓ |

Table 1: Comparison of recent neural human rendering methods. MonoNHR is the first work that 1) takes a monocular image as an input, 2) is supervised with multi-view images without 3D scans, and 3) is generalizable to unseen subjects (identities).

- We design the network to handle the specific challenges of the task, such as invisible surface synthesis and depth ambiguity. We tackle the former via a mesh inpainting module. For the latter, we disentangle 3D geometry and texture features, and condition texture inference based on the geometry features.

- The proposed system significantly outperforms previous methods on monocular images both quantitatively and qualitatively, and achieves state-of-the-art rendering quality on novel view synthesis benchmarks.

2. Related work

Markerless Performance Capture from RGB data often required large acquisition platforms with tens to hundreds of cameras [4, 19, 43]. The problem has also been approached through the use of sparse setups [14, 47]. We refer the reader to [49] for a broader overview. Nevertheless, all the multi-view approaches share the same limitations: synchronizing and calibrating multi-camera systems is cumbersome, requires storing and processing large amounts of data, and not always feasible in practice. A few recent approaches tackled the monocular case [10, 50, 51], but they all require pre-scanned templates of the subjects.

Furthermore, most of the performance capture methods [1, 4, 6, 9, 26, 41] solve the problem in two distinct steps: 1) reconstructing the mesh of the observed subject, and 2) coloring it using available observations, possibly considering lighting information. The main drawback of such a strategy is that the appearance is conditioned on, but also limited by, the geometry, which is inherently noisy and inaccurate if not incomplete. In this work, we wish to switch paradigms and directly model view-dependent appearance. In fact, the idea was already introduced in [5]. We would like to follow their work, by investigating the potential use of NeRF [28] to represent view-dependent ap-
Neural surface fields-based human rendering methods are closely related to NeRF representations, since they also aim at learning an implicit function, in this case, an indicator of the interior and exterior of the observed shape. They allow for accurate and detailed 3D geometry reconstructions, but they require strong 3D supervision, such as 3D scans. 3D scans are highly costly to obtain at scale, and consequently, methods are trained on a small scan dataset and tend to exhibit poor generalization capabilities to unseen human poses and appearances. PIFu [34] and its extensions [13, 35] propose to estimate the 3D surface of a human using an implicit function based on pixel-aligned image features. The pixel-aligned image features are obtained by projecting 3D points onto the image plane. Similar to more classical approaches, they first reconstruct 3D surfaces and then condition texture inference on surface reconstruction features similar to ours. However, as DoubleField [40] pointed out, the learning space of texture is highly limited around the surface and discontinuous, which hinders the optimization. Zins et al. [57] improves PIFus in a multi-view setting by introducing an attention-based view fusion layer and a context encoding module using 3D convolutions. POSEFusion [22] takes an attention-based view fusion layer and a context encoding module using 3D convolutions. POSEFusion [22] takes a sequence of point clouds as an input and learns to fuse multiple surface estimations in different time steps. DoubleField jointly learns neural radiance and surface fields, and uses raw RGB pixel values to render high-resolution images.

Compared to the above neural surface fields-based human rendering methods, MonoNHR has two clear differences. First, it does not require 3D scans for training and follows NeRF-based human rendering pipeline, i.e., using a weak supervision signal from multi-view images. Although NeRF-based human rendering methods, including ours, use SMPL fits, the SMPL fits are much easier to obtain than 3D scans using existing powerful 3D human pose and shape estimation methods [17, 29]. On the other hand, special and expensive equipments, e.g., over 100 multi-view synchronized cameras, are necessary to generate accurate 3D scans, making them difficult to obtain at a large scale. Please note that 3D scans obtained from a small number of cameras, e.g., COLMAP [37, 38], are not accurate enough to provide supervision targets for PIFu and its variants, as discussed in [20, 32]. In consequence, PIFu and its variants are trained on small scale datasets and tend not to generalize well to unseen data, especially non-upright standing poses, as discussed in [18, 32]. Second, the absence of 3D scans prevents explicit 3D geometry supervision, making disentangling geometry and texture non-trivial. We extract geometry-dedicated features for disentanglement and use them for density estimation without RGB estimation.

3. MonoNHR

The overall pipeline of MonoNHR is detailed in Figure 2. It is trained in an end-to-end manner and consists of an image feature backbone, Mesh Inpainter, a geometry branch, and a texture branch.
3.1. Backbone

First, a backbone extracts image features that will be used to condition the neural radiance field. Similarly to [53], this allows MonoNHR to learn priors relevant to human rendering, which ultimately enables our method to work on subjects that were not seen during training. We use a ResNet18 [11] as backbone. It takes a masked human image $I \in \mathbb{R}^{3 \times 256 \times 256}$ as input and produces a feature map $F \in \mathbb{R}^{c_p \times 128 \times 128}$, with $c_p = 64$ the feature’s channel dimension. An off-the-shelf semantic part segmentation method, PGN [7], is used to segment the subjects in the images. The feature map $F$ is used in two ways:

**Mesh feature.** Following NHP [18], we sample per-vertex image features from $F$, given SMPL [24] mesh vertices. We project these 3D vertices onto the image plane and use bilinear interpolation to sample the corresponding image features. Then, we concatenate the per-vertex image features and the vertices’ root joint (i.e., pelvis)-relative depths, to distinguish the features of both occluded and occluding parts. The concatenated feature is called the mesh feature and denoted as $M \in \mathbb{R}^{M \times (c_p + 1)}$, where $M$ denotes the number of SMPL mesh vertices. We feed these to the geometry branch.

**Pixel feature.** We sample 3D query points from a ray of a target view (rendering view), following the quadrature rule of the volume rendering discussed by Max [27]. The rays are bounded by the given mesh’s 3D bounding box, following NB [32] and NHP [18]. The 3D query point $x \in \mathbb{R}^3$ is equipped with appearance information from its projection on the image plane. We coin this information the pixel feature $p \in \mathbb{R}^{c_p}$. The pixel feature $p$ is fed to both geometry and texture branches with $x$’s root joint-relative depth $z$. The root joint-relative depth is defined in the input view’s camera-centered coordinate system.

3.2. Mesh Inpainter

This module reconstructs the colors of mesh vertices from a monocular image. It is a simple MLP network that consists of two layers. It takes the mesh feature $M$ as input and regresses the vertices’ colors $K \in \mathbb{R}^{M \times 3}$. $K$ is only used during training. The vertices’ GT colors are obtained by 1. projecting the vertices to multi-view images considering visibility, 2. sampling the colors from each image with bilinear interpolation and 3. averaging the visible vertices’ sampled colors. The visibility of vertices is obtained by rasterizing mesh vertices.

Mesh Inpainter encourages the backbone network to implicitly learn human priors, such as symmetry of human parts and color similarity between surfaces belonging to the same part. While these could be learned by the rendering loss at the end of the network, we aim to facilitate it by giving a more direct signal to the backbone using mesh inpainting. Mesh Inpainter’s motivation is further explained in Section E of the supplementary material.

3.3. Geometry branch

The geometry branch regresses a density value $\sigma$ of a 3D query point $x$. To this aim, we leverage sparse 3D convolutions similarly to NB [32] and NHP [18].

**Sparse 3D convolution.** We work with a sparse 3D volume, in which lie the sparse features of $M$. This sparse 3D volume is fed to a sparse 3D convolutional neural network (CNN) [8] that extracts multi-scale feature volumes,
denoted mesh volumes. From each mesh volume, we sample the feature corresponding to \( x \) based on its 3D camera coordinates using trilinear interpolation to get the voxel feature \( v \in \mathbb{R}^c \), where \( c_v = 192 \) is the feature’s channel dimension. The voxel feature \( v \) is used only for the geometry estimation; therefore, it is a geometry-dedicated feature.

**MLP.** Then, the density of \( x \) is predicted as a function of the voxel feature \( v \), the pixel feature \( p \), and the root joint-relative depth \( z \), as follows:

\[
\sigma(x) = M_{\sigma}(v, p, z),
\]

where \( M_{\sigma} \) is another Multi-Layer Perceptron (MLP) network with four layers.

### 3.4. Texture branch

The texture branch regresses a RGB value \( c \) for a query point \( x \). The RGB color of \( x \) is estimated as a function of the pixel feature \( p \), the root joint-relative depth \( z \), and the target view’s ray direction \( d \in \mathbb{R}^3 \), conditioned on the predicted density value \( \sigma \), as follows:

\[
c(x) = M_c(p, z, Rd; \sigma),
\]

where \( M_c \) represents an MLP network with five layers, and \( R \) denotes the world-to-camera rotation matrix. \( R \) transforms the view direction to the input image’s coordinate system. \( \sigma \) provides information about the occupancy of a given 3D query point. Such occupancy information is highly useful as RGB values should exist on the occupied points. Please note that although we provide \( \sigma \) to the texture branch, the voxel feature \( v \) is still a geometry-dedicated feature as its role is to predict accurate density \( \sigma \).

### 3.5. Volume rendering

Given a target view, we use a classical differentiable ray-marching algorithm [16] to render the target image following NeRF [28]. Concretely, the final color of a pixel is computed as the integral of RGB values along the ray shot from the camera center \( o \in \mathbb{R}^3 \), weighted by predicted volume densities. The integral is approximated via stratified sampling [28], and we use the quadrature rule [27] to limit memory usage and predict continuous radiance fields in practice. For each pixel, we sample along the ray \( N \) query points \( \{x_i\}_{i=1}^N \), where \( x_i = (o + z_i \cdot d) \) and \( z_i \in [z_{\text{near}}, z_{\text{far}}] \). \( z_{\text{near}} \) and \( z_{\text{far}} \) are the absolute depths of the two intersections between the ray and the given SMPL mesh’s 3D bounding box. Then, the pixel color of the ray is computed as below:

\[
\hat{C}(r) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma(x_i) \delta_i)) c(x_i)
\]

and

\[
T_i = \exp(-\sum_{j=1}^{i-1} \sigma(x_j) \delta_j),
\]

where \( \delta_i = ||x_{i+1} - x_i||_2 \) is the distance between adjacent sampled points, and \( r \) is the camera ray. In practice, we set \( N \) to 64 following [18, 32].

### 3.6. Loss functions

We supervise MonoNHR with two loss functions.

**Inpainting loss.** The inpainting loss compares the predicted and GT colors of vertices. It is defined as below:

\[
L_{\text{inpaint}} = \sum_{i \in R} \left\| \hat{K} - K \right\|_2^2,
\]

where \( \hat{K} \) and \( K \) indicate the predicted and GT colors of vertices, respectively. Section 3.2 describes how we obtain the GT color of vertices \( K \).

**Rendering loss.** The rendering loss compares the predicted pixel color from the volume rendering and the GT pixel color. We randomly sample four views to calculate the loss. It is defined as below:

\[
L_{\text{render}} = \sum_{r \in R} \left\| \hat{C}(r) - C(r) \right\|_2^2,
\]

where \( R \) is a set of 3D query points on camera rays passing through image pixels, and \( C(r) \) means the GT pixel color.

MonoNHR is trained in an end-to-end manner, and the total loss is defined as:

\[
L_{\text{total}} = L_{\text{render}} + \lambda L_{\text{inpaint}},
\]

where \( \lambda \) is set to \( 10^{-3} \).

### 4. Experiments

#### 4.1. Datasets

**Datasets.** We train and test MonoNHR on ZJU-MoCap [32], AIST [21, 45], CAPE [25], and HUMBI [55]. For ZJU-MoCap, we split 10 videos of 10 subjects (1 video per 1 subject) to 7 training videos and 3 testing videos. For AIST, we follow the official training and testing splits. The training set has 834 videos of 20 dancing subjects, and the testing set has 374 videos of 10 dancing subjects. Please note that the testing splits of all datasets contain poses and identities never seen during training. CAPE is used to compare 3D geometry with non-NeRF methods as the above two datasets do not provide GT 3D geometry. We use the above datasets for numerical studies, for more meaningful comparisons with prior work, but we also train on HUMBI for qualitative evaluation, for its larger size and diversity (see section 4.4). More details about data settings can be found in the supplementary material.

**Evaluation metrics.** We report peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to evaluate the quality of novel view synthesis, following [18, 32, 53]. PSNR has been widely used to assess the quality of digital
Figure 3: Comparison of pixelNeRF [53], NHP [18], and MonoNHR. All results are obtained from monocular images with unseen poses and identities. Top two rows: results on ZJU-MoCap, bottom two rows: results on AIST.

images. SSIM is designed based on luminance, contrast, and structure to better suit the workings of the human visual system [39]. For the 3D geometry evaluation, we report Chamfer distance (CD) in millimeters.

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

Overall comparison. The Figure 3 and Table 2 show that MonoNHR produces the best results on both ZJU-MoCap [32] and AIST [21,44].

While pixelNeRF [53] and NHP [18] tend to succeed in recovering only geometry or texture, MonoNHR produces robust novel view images in both aspects. Concretely, pixelNeRF has a weakness in geometry reconstruction, and NHP struggles on recovering textures of invisible surfaces. For example, on AIST, pixelNeRF has higher PSNR but lower SSIM than NHP. PixelNeRF shows bent and missing human limbs and produces particularly unrecognizable structures of humans, when the target view is significantly different from the input view, and the human pose is complex. NHP synthesizes better novel views than pixelNeRF in terms of geometry, but tends to exhibit noisy textures on invisible surfaces. We think the noisy texture is due to the entanglement of geometry and texture features, which hinders a network from simultaneously optimizing them and eventually leads it to a local optimum that only optimizes either the geometry or the texture. On the other hand, MonoNHR provides robust qualitative results of geometry and texture from diverse novel views, including the opposite view to the input image. The superiority of MonoNHR not only comes from the disentanglement of geometry and texture features, but also from Mesh Inpainter that facilitates the system to learn human priors. For instance, the first two rows of Figure 3 show that MonoNHR recovers robust geometry and texture of the subjects’ occluded arms, which proves that it successfully learns human priors, such as symmetry of human parts, to compensate for the lack of information on invisible areas. We refer the reader to the supplementary material for additional comparisons.

Comparison using estimated 3D meshes. We also compare MonoNHR with NHP [18] on ZJU-MoCap [32] using estimated body meshes of SPIN [17] in Table 3. SPIN is a monocular method that regresses SMPL [24] parameters to produce a body mesh. MonoNHR highly outperforms NHP, which also uses a body mesh at test time, considering that PSNR is in a log scale. Figure 4 further validates MonoNHR’s robustness on monocular images. MonoNHR preserves the subject’s 3D shape (e.g., loose clothes around the torso of the right-hand subject) while NHP relatively shrinks the subject’s overall shape (e.g., the arms of the right-hand subject). Also, the results of MonoNHR show more realistic textures in novel views than NHP (e.g., the shirt of the left-hand subject).

3D geometry comparison. Figure 5 compares the 3D geometry reconstruction of MonoNHR with NHP [18] on ZJU-MoCap. MonoNHR clearly distinguishes the geometry of the top subjects’ shirts and pants, while NHP shows vague boundaries between them. Also, on the SPIN example on the right (b), the result from MonoNHR reflects the widened deformation of the subject’s shirt in the input image, but NHP recovers the shirt geometry in a tightened shape. The results prove that MonoNHR learns better ge-
Figure 4: Comparison between NHP [18] and MonoNHR on ZJU-MoCap using estimated meshes of SPIN [17].

Figure 5: 3D geometry visualization of NHP [18] and MonoNHR.

Figure 6: Comparison between state-of-the-art non-NeRF methods and ours on ZJU-MoCap.

Table 4: Comparison between state-of-the-art non-NeRF methods and ours on CAPE.

| method          | CD ↓ |
|-----------------|------|
| PIFu [34]       | 79.67|
| PaMIR [56]      | 74.24|
| MonoNHR (Ours)  | 23.94|

4.3. Ablation studies

Disentanglement of geometry and texture. To show the benefit of the proposed disentanglement, we design a model where voxel features $v$ are fed to both geometry and texture branches. Results confirm that disentangling geometry and texture improves predictions both qualitatively and quantitatively (Table 5). As shown in Figure 7a, noise on the invisible surface's texture is removed, and the overall visual quality is enhanced.

Mesh Inpainter. Using the Mesh Inpainter improves results both qualitatively (Fig. 7b) and quantitatively (Table 5). The critical challenge of the monocular setting is inferring the geometry and texture of invisible areas. To this end, the backbone should produce proper features for unseen parts based on the visible ones, which amounts to learning human priors, such as symmetry of human parts and color similarity between surfaces belonging to the same part. While these could be learned by the rendering loss at the end of the network, we tried to facilitate it by giving a more direct signal to the backbone using mesh inpainting.

Geometry-conditioned texture estimation. Conditioning on geometry when estimating texture is essential (Fig. 7c and Table 5). Indeed, geometry provides information on whether a 3D query point is occupied or not, and such occu-
### Table 5: Ablation studies on AIST.

| setting                  | PSNR ↑ | SSIM ↑ |
|--------------------------|--------|--------|
| w/o Mesh Inpainter       | 17.52  | 0.7064 |
| w/o disentanglement      | 17.53  | 0.7167 |
| w/o geo. cond.           | 17.38  | 0.6634 |
| Ours: full model         | 17.61  | 0.7186 |

4.4. Qualitative results

The HUMBI dataset [54] has a high diversity of human subjects in terms of body shape, age, ethnicity, clothing and accessories. We train on this data in order to evaluate the robustness and generalization capabilities of our method. Figures 1 and 8 show some results on test subjects unseen during training. In the latter, we show the benefits of using neural rendering by comparing with surface renderings of the inpainted SMPL mesh. Despite the accuracy of the SMPL annotations on this dataset, we can clearly see the inherent limits of texture based approaches. They typically fail in regions where the true surface largely differs from the template shape which is often the case e.g. with hair or wide cloths. In contrast, even though our approach makes use of the SMPL vertices, we exhibit good robustness to these cases.

5. Conclusion

We proposed MonoNHR, a NeRF-based approach that renders robust free-viewpoint images of an arbitrary human given a single monocular image. By disentangling 3D geometry features from texture features and enforcing the feature extractor to exploit human priors (e.g., symmetry), we reached state-of-the-art performance for the problem of novel view synthesis of an arbitrary person from a monocular observation. We also showed quantitative and qualitative results using estimated meshes, which take a step toward the authentic monocular setting. We believe our results are inspiring in that 1) robust view synthesis for partially observed humans is feasible with a NeRF-based approach, and 2) we tackles the monocular problem without explicit 3D supervision, and thus it can be more easily scaled-up to large datasets than methods that rely on 3D scans. In future work, adversarial losses could be employed to learn human priors further and improve the realism of novel view synthesis.
References

[1] Matthieu Armando, Jean-Sébastien Franco, and Edmond Boyer. Adaptive mesh texture for multi-view appearance modeling. In 3DV, 2019.

[2] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, and Huchuan Lu. Animatable neural radiance fields from monocular rgb video. arXiv preprint arXiv:2106.13629, 2021.

[3] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In CVPR, 2021.

[4] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Esseev, David Calabrese, Hugues Hoppe, Adam G. Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. ACM TOG, 2015.

[5] Paul E. Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Mans. In SIGGRAPH, 2000.

[6] Jean-Sébastien Franco and Edmond Boyer. Efficient polyhedral modeling from silhouettes. TPAMI, 2009.

[7] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, and Liang Lin. Instance-level human parsing via part grouping network. In ECCV, 2018.

[8] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In CVPR, 2018.

[9] Kaiwen Guo, Peter Lincoln, Philip L. Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Bourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Ryan Fanello, Graham Fyffe, Christoph Rhe- mann, Jonathan Taylor, Paul E. Debevec, and Shahram Izadi. The relightables: volumetric performance capture of humans with realistic relighting. ACM TOG, 2019.

[10] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Livecap: Real-time human performance capture from monocular video. ACM TOG, 2019.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2015.

[12] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. Arch++: Animation-ready clothed human reconstruction revisited. ArXiv, abs/2108.07845, 2021.

[13] Yang Hong, Juyong Zhang, Boyi Jiang, Yudong Guo, Ligang Liu, and Hujun Bao. Stereopifu: Depth aware clothed human digitization via stereo vision. In CVPR, 2021.

[14] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, ECCV, 2018.

[15] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[16] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. In SIGGRAPH, 1984.

[17] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In ICCV, 2019.

[18] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural Human Performer: Learning generalizable radiance fields for human performance rendering. NeurIPS, 2021.

[19] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Volume sweeping: Learning photoconsistency for multi-view shape reconstruction. IJCV, 2021.

[20] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In ECCV, 2018.

[21] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation. In ICCV, 2021.

[22] Zhe Li, Tao Yu, Zerong Zheng, Kaiwen Guo, and Yebin Liu. Posefusion: Pose-guided selective fusion for single-view human volumetric capture. In CVPR, 2021.

[23] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. ACM TOG Asia, 2021.

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

[25] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In CVPR, 2020.

[26] Shugao Ma, Tomas Simon, Jason M. Saragih, Dawei Wang, Yuecheng Li, Fernando De la Torre, and Yaser Sheikh. Pixel codec avatars. In CVPR, 2021.

[27] Nelson Max. Optical models for direct volume rendering. TVCG, 1995.

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In ECCV, 2020.

[29] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3D hand pose estimation for whole-body 3D human mesh estimation. In CVPRW, 2022.

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

[31] Sida Peng, Junting Dong, Qianqian Wang, Shangzhuan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable neural radiance fields for human body modeling. In ICCV, 2021.

[32] Sida Peng, Yuqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In CVPR, 2021.

[33] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. Pixel-aligned volumetric avatars. In CVPR, 2021.
[34] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In ICCV, 2019.

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

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

[37] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In CVPR, 2016.

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

[39] De Rosal Ignatius Moses Setiadi. Psnr vs ssim: imperceptibility quality assessment for image steganography. MTA, 2021.

[40] Ruizhi Shao, Hongwen Zhang, He Zhang, Yanpei Cao, Tao Yu, and Yebin Liu. Doublefield: Bridging the neural surface and radiance fields for high-fidelity human rendering. arXiv preprint arXiv:2106.03798, 2021.

[41] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. IEEE Computer Graphics and Applications, 2007.

[42] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. In Advances in Neural Information Processing Systems, 2021.

[43] Christian Theobalt, Edilson de Aguiar, Carsten Stoll, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from multi-view video. In Rémi Ronfard and Gabriel Taubin, editors, Image and Geometry Processing for 3-D Cinematography. 2010.

[44] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In ISMIR, 2019.

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

[46] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. TPAMI, 2021.

[47] Pierre Zins, Yuanlu Xu, Edmond Boyer, Stefanie Wohrer, and Tony Tung. Learning implicit 3d representations of dressed humans from sparse views. In 3DV, 2021.