TAVA: Template-free Animatable Volumetric Actors

Ruilong Li[1,3], Julian Tanke[2,3], Minh Vo[3], Michael Zollhöfer[3],
Jürgen Gall[2], Angjoo Kanazawa[1], Christoph Lassner[3]

1. UC Berkeley 2. University of Bonn 3. Meta Reality Labs Research

Abstract. Coordinate-based volumetric representations have the potential to generate photo-realistic virtual avatars from images. However, virtual avatars also need to be controllable even to a novel pose that may not have been observed. Traditional techniques, such as LBS, provide such a function; yet it usually requires a hand-designed body template, 3D scan data, and limited appearance models. On the other hand, neural representation has been shown to be powerful in representing visual details, but are under explored on deforming dynamic articulated actors. In this paper, we propose *TAVA*, a method to create *T*emplate-free *A*nimatable *V*olumetric *A*ctors, based on neural representations. We rely solely on multi-view data and a tracked skeleton to create a volumetric model of an actor, which can be animated at the test time given novel pose. Since TAVA does not require a body template, it is applicable to humans as well as other creatures such as animals. Furthermore, TAVA is designed such that it can recover accurate dense correspondences, making it amenable to content-creation and editing tasks. Through extensive experiments, we demonstrate that the proposed method generalizes well to novel poses as well as unseen views and showcase basic editing capabilities. The code is available at https://github.com/facebookresearch/tava

1 Introduction

Ever since the first 3D vector graphics games in the 1980s, we are striving to build better representations of 3D objects and humans. With increasing processing power, we can afford to capture, reconstruct and encode increasingly realistic representations. This makes exploring neural representations for graphical objects particularly appealing—it is a representation that has proven powerful [27,49,48,19,37], even though still being in its infancy. Recent methods for neural 3D representations go beyond capturing plain texture by modeling radiance fields [27,2,3], achieving more photo-realistic results than rasterization-based approaches [50,23,20,38]. However, it is unclear how their representational power can be used to not only capture *static*, but also *dynamic* scenes that can be animated in a meaningful way, making the representations useful for capturing actors that can be "driven" post-capture. However, due to the high-dimensional nature of pose configurations, it is generally neither possible nor practical to capture all pose variations in one capture. This poses a new problem absent in static settings:

[†] Work was done partially while Ruilong and Julian were at Meta Reality Lab.

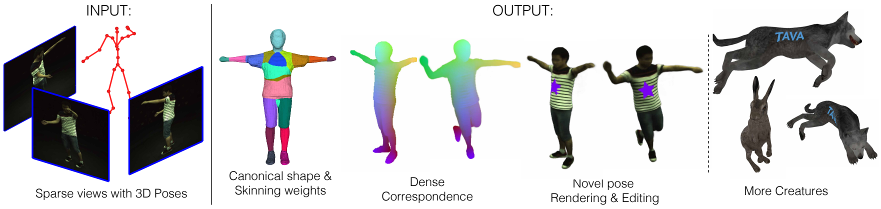


Fig. 1. *Method Overview.* **Left:** TAVA creates a virtual actor from multiple sparse video views as well as 3D poses. The same skeleton can later be used for animation. **Center:** TAVA uses this information to create a canonical shape and a pose-dependent skinning function and establishes correspondences across poses. The resulting model can be used for rendering and posing the virtual character as well as editing it. **Right:** the method can directly used for other creatures as long as a 3D skeleton can be defined.

generalization to out-of-distribution (OOD) poses. Furthermore, the neural 3D representation is desired to be *editable* as the classical representation like mesh and textures. Some works explored this aspects in the static settings [49,48], but it is not clear on how to edit neural representations on dynamic actors.

In this paper, we propose TAVA, a novel approach for **T**emplate-free **A**nimatable **V**olumetric **A**vatars (illustrated in Fig. 1). We propose to use coordinate-based radiance fields to capture appearance, leading to high quality, faithful renderings. We extend the radiance capture with a carefully designed deformation model: while it requires solely 3D skeleton information at training time, it captures non-linear pose-dependent deformations and exhibits stable generalization behavior to unseen poses thanks to being anchored in an LBS formulation. The radiance field and deformation model are optimized jointly and end-to-end, leading to a simple-to-use and powerful representation: creating it requires only a tracked skeleton and multi-view photometric data, **no** template mesh or artist-designed rigging; the appearance and the deformation model can complement each other for highest quality results. These designed properties make TAVA suitable for content creation and editing as well as correspondence-based matching.

In our experiments, we demonstrate that the proposed approach outperforms state-of-the art approaches for animating and rendering human actors on the ZJU motion capture dataset [35]. Thanks to being template-free, our approach is not limited to capturing humans: we present a detailed evaluation and ablation study on two synthetically rendered animals. This demonstrates the flexibility of the proposed approach and allows us to show additional applications in content-creation and editing.

2 Related Work

Deformable Neural Scene Representations: Coordinate-based neural scene representations produce impressive results in encoding shape [26,8,31] and appearance [24,27,40]. These methods train a coordinate-based neural network to model various properties of a scene, e.g., occupancy [26], distance to the closest surface [31], or density and color [27]. However, making implicit scene representations deformable and animatable remains a challenging research problem. Nerfies [32] and Neural 3D Video Synthesis [21] handle changes in the scene by optimizing a deformation field and a latent code for each frame. HyperNeRF [33] extends this by additionally creating a hyper-space which allows topology changes of the scene. Non-Rigid Neural Radiance Fields [43] optimize a rigidity model in addition to a deformation field. While these methods produce impressive results on dynamic scenes, they are designed to only memorize the scene and cannot control the scene beyond interpolations.

Animatable Neural Radiance Fields: Recently, many approaches for controllable animatable NeRFs have been proposed. Neural Actor [22] uses a pose-dependent radiance field by warping rays into the canonical space of a template body model while using 2D texture maps to model fine detail. NeuralBody [35] anchors latent codes on the vertices of a deformable mesh controlled by LBS. The follow-up work Animatable-NeRF [34] establishes a transformation between view and canonical space through optimizing the inverse deformation field. Other works like NARF, A-NeRF [29,42] predict the radiance field at a given 3D location based on its relative coordinates to the bones. Most recently, a concurrent work HumanNeRF [45] produces a free-viewpoint rendering of a human by modeling the inverse deformation as a mixture of affine fields [24]. Yet many of these methods [29,42] do *not* have a 3D canonical space that preserves correspondences across different poses, which is required for content-creation or editing. Some [22,34,35,45] are built on top of the SMPL [25] body template, which prohibits them to be applied to creatures beyond humans. Moreover, most of the aforementioned methods either introduce latent codes to better memorize the seen poses [42,34,35], or represent the deformation in the inverse direction from view space to the canonical space [29,42,34,45]. Thus they do not generalize well to the unseen poses because the existence of pose-conditioned MLPs. In contrast, our approach is template-free, enables editing, and is designed to be robust to unseen poses. We provide an overview of the comparison between our method and those previous works in Tab. 1.

Animatable Shapes: Non-rigid shape reconstruction often utilizes a canonical space that is fixed across frames, with a deformation model to create a mapping between the canonical and the deformed space. Traditionally, this has been achieved by extracting a low dimensional articulated mesh [4,5,6,41,10,46,17,18,44,9,12,1], such as SMPL [25], or by extracting a rigged mesh via post-processing. Several methods [15,13,25,30,13,16,51,47] have been proposed to optimize blend weights and rigs from data. ARCH [14] deforms an estimated implicit representation to fit to a clothed human using a single image. Recent approaches model inverse deformation fields [11,28,32,36,39], which map points from pose-dependent global space to pose-independent canonical space where the surface is represented. For example, SCANimate [39] regularizes the inverse skinning by using a cycle consistency loss. The main drawback of these inverse deformation approaches is that the inverse transformation is pose dependent and may not generalize well to previ-

Methods	Template-free	No Per-frame Latent Code	3D Canonical Space	Deformation
NARF [29]	✔	✔	✘	Inverse
A-NeRF [42]	✔	✘	✘	Inverse
Animatable-NeRF [34]	✘	✘	✔	Inverse
HumanNeRF [45]	✘	✔	✔	Inverse
NeuralBody [35]	✘	✘	✔[†]	Forward
Ours (TAVA)	✔	✔	✔	Forward

Table 1. *Design differences.* TAVA's use of a forward deformation model without using per-frame latent codes ensures robustness to out-of-distribution poses. Being template-free extends its use to creatures beyond humans. TAVA also allows for content-creation and editing by using a 3D canonical space. [†]Note that NeuralBody's canonical space consists of the body template *without* clothing.

ously unseen poses. SNARF [7] addresses this by learning a forward deformation field instead, mapping points from canonical to pose-dependent deformed space. However, unlike our appraoch, these methods require 3D geometry supervision and most do not optimize for appearance.

3 Method

Our goal is to create an animatable neural actor from multi-view images with known 3D skeleton information without requiring a body template. Similar to a traditional personalized body rig, we want to build a representation that not only represents the shape and appearance of the actor but also allows to animate it while maintaining correspondence among different poses and views. *TAVA* is designed to achieve the above goals with three components: (1) a canonical representation of the actor in neutral pose, (2) deformation modeling based on forward skinning, and (3) volumetric neural rendering with pose-dependent shading. Fig. 2 illustrates an overview of our method. To employ volumetric neural rendering in the view space, our method first deforms the samples along a ray back to the canonical space through inverting the forward skinning via root-finding, then queries their colors and densities in the canonical space, as well as the pose-dependent effects. Below, we first establish preliminaries, then discuss each of the components.

3.1 Preliminary: Rendering Neural Radiance Fields

NeRF [27] is a groundbreaking technique for novel view synthesis of a *static* scene. It models the geometry and view-dependent appearance of the scene by using a multi-layer perceptron (MLP). Given a 3D coordinate $\mathbf{x} = (x, y, z)$ and the corresponding viewing direction (θ, ϕ), NeRF queries the emitted color $\mathbf{c} = (r, g, b)$ and material density σ at that location using the MLP. A pixel color

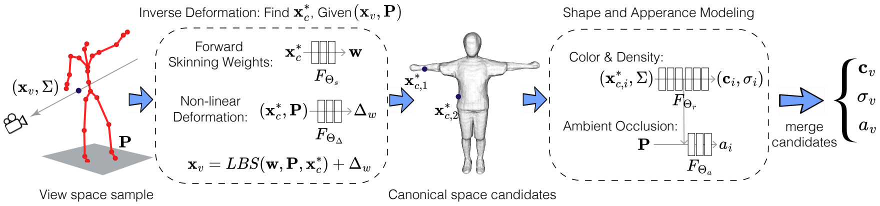

Inverse Deformation: Find $\mathbf{x}_c^*$, Given $(\mathbf{x}_v, \mathbf{P})$
Forward Skinning Weights: $\mathbf{x}_c^* \rightarrow \mathbf{w}$
F_{Θ_s}
Non-linear Deformation: $(\mathbf{x}_c^*, \mathbf{P}) \rightarrow \Delta_w$
F_{Θ_Δ}
$\mathbf{x}_v = LBS(\mathbf{w}, \mathbf{P}, \mathbf{x}_c^*) + \Delta_w$
$(\mathbf{x}_v, \Sigma)$
$\mathbf{P}$
View space sample
$\mathbf{x}_{c,1}^*$
$\mathbf{x}_{c,2}^*$
Canonical space candidates
Shape and Apperance Modeling
Color & Density:
$(\mathbf{x}_{c,i}^*, \Sigma) \rightarrow (\mathbf{c}_i, \sigma_i)$
F_{Θ_r}
Ambient Occlusion:
$\mathbf{P} \rightarrow a_i$
F_{Θ_a}
merge candidates
$\mathbf{c}_v$
σ_v
a_v

Fig. 2. *TAVA Overview.* We use volumetric rendering techniques to create the actor representation. For each sampled point, we use LBS based non-linear deformation combined with a blending weight model for which we identify the root in the canonical space. In this space, we use a color, density, and ambient occlusion model to parameterize the appearance.

$C(\mathbf{r})$ can then be computed by accumulating the view-dependent colors along the ray $\mathbf{r}$, weighted by their densities:

$$C(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(\sigma_i \delta_i))\mathbf{c}_i, \quad \text{where } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j) \ , \tag{1}$$

where δ_i denotes the distances between the sample points along the ray. To further take the size of the pixels into consideration, Mip-NeRF [2] extends NeRF to represent each ray $\mathbf{r}$ that passes through a pixel as a cone, and the samples $\mathbf{x}$ along the ray as conical frusta, which can be modeled by multivariate Gaussians $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Thus, the density σ and view-dependent emitted color $\mathbf{c}$ for a sample on the ray are given by $F_\Theta : (\boldsymbol{\mu}, \boldsymbol{\Sigma}, \theta, \phi) \rightarrow (\mathbf{c}, \sigma)$, where $\boldsymbol{\mu} = (x, y, z)$ is the center of the Gaussian and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times3}$ is its covariance matrix. The loss for optimizing the network parameters Θ of the neural radiance field is applied between the rendered pixel color $C(\mathbf{r})$ and the ground-truth $\hat{C}(\mathbf{r})$:

$$\mathcal{L}_{im} = \left|\left|C(\mathbf{r}) - \hat{C}(\mathbf{r})\right|\right|_2^2 . \tag{2}$$

Please refer to the original papers [27,2] for more details.

3.2 Canonical Neural Actor Representation

We represent an articulated subject as a volumetric neural actor in its canonical space. The representation includes a *Lambertian* neural radiance field F_{Θ_r} to represent the geometry and appearance of this actor, and a neural blend skinning function F_{Θ_s}, which describes how to animate the actor:

$$F_{\Theta_r} : (\mathbf{x}_c, \boldsymbol{\Sigma}) \rightarrow (\mathbf{c}, \sigma), \qquad F_{\Theta_s} : \mathbf{x}_c \rightarrow \mathbf{w}, \tag{3}$$

where $\mathbf{c} = (r, g, b)$ is the material color, σ is the material density, and $\mathbf{w}$ is the skinning weights to blend all bone transformations for animation. Similar to Mip-NeRF [2], we use a multivariate Gaussian $(\mathbf{x}_c \in \mathbb{R}^3, \boldsymbol{\Sigma} \in \mathbb{R}^{3\times3})$ to estimate the integral of samples within the volume of the discrete samples. Note that in most of the cases, an articulated actor is a Lambertian object, so we exclude view directions from the input of F_{Θ_r}.

Discussion. This formulation not only models the *canonical* geometry and appearance of an avatar, but also describes its *dynamic* attributes through the skinning weights $\mathbf{w}$. Unlike previous works, such as SNARF [7], which models a pose-dependent geometry in the canonical space, and NARF [29] and A-NeRF [42], which entirely skips canonical space modeling, our method is based on a canonical representation that fully eliminates *any* effects of pose on the geometry and appearance. Moreover, the skinning weights learnt in the canonical space remain valid for a large range of poses, meaning that the actor is ready to be animated in novel poses outside of the training distribution (see Sec. 4.2 for validation of its robustness to out-of-distribution novel poses). Last but not least, our representation eases the correspondence finding problem across different poses and views, because the matching can be done in the pose-independent canonical space (see Sec. 4.2 for results).

3.3 Skinning-based Deformation

Forward Skinning. With the skinning weights $\mathbf{w} = (w_1, w_2, ..., w_B, w_{bg}) \in \mathbb{R}^{B+1}$ defined in the canonical space, and given a pose $\mathbf{P} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_B\} \in \mathbb{R}^{B\times4\times4}$, we use forward LBS to define the deformation of a point $\mathbf{x}_c$ in the canonical space to $\mathbf{x}_v$ in the view space:

$$\mathbf{x}_v = LBS(\mathbf{w}(\mathbf{x}_c; \Theta_s), \mathbf{P}, \mathbf{x}_c) = \left[\sum_{j=1}^{B} w_j(\mathbf{x}_c; \Theta_s) \cdot \mathbf{T}_j + w_{bg} \cdot \mathbf{I_d}\right] \mathbf{x}_c, \quad (4)$$

where $\mathbf{I}_d \in \mathbb{R}^{4\times4}$ is an identity matrix. Similar to [45], we extend the classic LBS defined only on the surface geometry of an object to the entire 3D space by introducing an additional term $w_{bg} \cdot \mathbf{I_d}$. This term allows the points in the background and empty space to *not* follow the skeleton when it is deformed. However, LBS is not sufficient for capturing some of the non-linear deformations, such as muscles and clothing dynamics [25]. Thus, we introduce an additional term $F_{\Theta_\Delta} : (\mathbf{x}_c, \mathbf{P}) \rightarrow \Delta_w \in \mathbb{R}^3$ on top of the learned LBS to model these deformations:

$$\mathbf{x}_v = LBS(\mathbf{w}(\mathbf{x}_c; \Theta_s), \mathbf{P}, \mathbf{x}_c) + \Delta_w(\mathbf{x}_c, \mathbf{P}; \Theta_\Delta). \quad (5)$$

Inverse Skinning. To render this model, we need to query color and density in the view space. So it is required to find the the correspondence $\mathbf{x}_c$ in the canonical space for each $\mathbf{x}_v$ in the view space. As our forward skinning in Eq. 5 is defined through neural networks, there is no analytical form for the inverse skinning. So, inspired by SNARF [7], we pose this as a root finding problem:

$$\text{Find } \mathbf{x}_c^*, \quad \text{s.t. } f(\mathbf{x}_c^*) = LBS(\mathbf{w}(\mathbf{x}_c^*; \Theta_s), \mathbf{P}, \mathbf{x}_c^*) + \Delta_w(\mathbf{x}_c^*, \mathbf{P}; \Theta_\Delta) - \mathbf{x}_v = \mathbf{0} \quad (6)$$

and solve it numerically using Newton's method:

$$\mathbf{x}_c^{(k+1)} = \mathbf{x}_c^{(k)} - (\mathbf{J}^{(k)})^{-1} f(\mathbf{x}_c^{(k)}), \tag{7}$$

where $\mathbf{J}^{(k)} \in \mathbb{R}^{3\times 3}$ is the Jacobian of $f(\mathbf{x}_c^{(k)})$ at the k-th step. Since the inverse skinning might be a one-to-many mapping when there is contact happening between body parts, we initialize Newton's method with multiple candidates using the inverse rigid transformation $\{\mathbf{x}_{c,i}^{(0)}\} = \{\mathbf{T}_i^{-1} \cdot \mathbf{x}_v\}$. However, simply applying all $B+1$ transformations to initialize the Newton's method would lead to $B+1$ canonical candidates to be processed, making it impractical for volumetric rendering as the complexity grows linearly in B. As points are less likely to be affected by bones further away, we only use the transformations of its $K = 5$ nearest bones by measuring the Euclidean distance between the point and the bones in the view space. This dramatically reduces the computational burden of the root finding process as well as the following canonical querying, making it feasible for neural rendering. With that, our inverse skinning leads to multiple correspondences for a point in the view space through root finding (r.f.):

$$\mathbf{x}_v \xrightarrow{\text{r.f.}} \{\mathbf{x}_{c,1}^*, \mathbf{x}_{c,2}^*, ..., \mathbf{x}_{c,K}^*\} \tag{8}$$

The gradients of the network parameters Θ_s and Θ_Δ can then be analytically computed for the inverse skinning [7]:

$$\frac{\partial \mathbf{x}_{c,i}^*}{\partial \Theta_s} = -\left[\frac{\partial \mathbf{x}_v}{\partial \mathbf{x}_{c,i}^*}\right]^{-1}\left[\frac{\partial \mathbf{x}_v}{\partial \Theta_s}\right]. \qquad \frac{\partial \mathbf{x}_{c,i}^*}{\partial \Theta_\Delta} = -\left[\frac{\partial \mathbf{x}_v}{\partial \mathbf{x}_{c,i}^*}\right]^{-1}\left[\frac{\partial \mathbf{x}_v}{\partial \Theta_\Delta}\right]. \tag{9}$$

Please refer to the supplemental material for their derivations.

3.4 Deformation-based Neural Rendering

Similar to Mip-NeRF [2], we render the color of a pixel by accumulating the samples $(\mathbf{x}_v, \mathbf{\Sigma})$ along each pixel ray, using Eq. 1. Instead of directly querying the color and density of $\mathbf{x}_v$ in the view space, we first find the point's canonical correspondence candidates using the inverse skinning $\mathbf{x}_v \xrightarrow{\text{r.f.}} \{\mathbf{x}_{c,1}^*, \mathbf{x}_{c,2}^*, ..., \mathbf{x}_{c,K}^*\}$, and then query the colors and densities for all those candidates in the canonical space .

$$F_{\Theta_r} : (\mathbf{x}_{c,i}^*, \mathbf{\Sigma}) \rightarrow (\mathbf{c}_i^*, \sigma_i^*). \tag{10}$$

However, for a dynamic object, the shading on the surface may change depending on pose due to self-occlusion. This can lead to colors in the view space being darker than the colors in the canonical space, providing inconsistent supervision signals. However, it is non-trivial to accurately model this self-occlusion without ray tracing (including secondary rays) and known global illumination. A simple but effective estimator, widely used in modern rendering engines like Unreal and Blender is ambient occlusion, in which the shading caused by occlusion is modeled by *a scaling factor* multiplied with the color values, where the value is calculated by the percentage of view directions being occluded around each point on the surface. Since it is an attribute defined at each coordinate that depends on the global geometry of the actor, we model this shading effect use a coordinate based MLP F_{Θ_a} conditioned on the pose $\mathbf{P}$ of the actor:

$$F_{\Theta_r} : (\mathbf{x}^*_{c,i}, \mathbf{\Sigma}) \rightarrow \mathbf{h} \rightarrow (\mathbf{c}^*_i, \sigma^*_i), \qquad F_{\Theta_a} : (\mathbf{h}, \mathbf{P}) \rightarrow a^*_i, \tag{11}$$

where $\mathbf{h}$ is an intermediate activation from F_{Θ_r}, and a^*_i is the ambient occlusion at this location under pose $\mathbf{P}$. Note that only the ambient occlusion a^*_i is pose-conditioned, which makes sure the actor (geometry and appearance) is represented in a canonical space that is pose-independent, as described in Sec. 3.2.

With $(\mathbf{c}^*_i, \sigma^*_i, a^*_i)_{i=1,...,K}$ queried in the canonical space, we then need to merge the K candidates to get the final attributes $(\mathbf{c}_v, \alpha_v, a_v)$ for the sample $(\mathbf{x}_v, \mathbf{\Sigma})$ in the view space. In the case of articulated objects, where multiple canonical point may originate from the same location, the one with the maximum density would dominate that location. Similar to previous works [7,11], we choose the attributes of $\mathbf{x}_v$ from all canonical candidates based on their density:

$$\mathbf{c}_v = \mathbf{c}^*_{c,t} \quad \sigma_v = \sigma^*_{c,t} \quad a_v = a^*_{c,t} \qquad \text{where } t = \operatorname*{argmax}_i(\{\sigma^*_{c,i}\}), \tag{12}$$

then we use $(\mathbf{c} = a_v * \mathbf{c}_v, \sigma = \sigma_v)$ as the final emitted color and density in the view space, for the volumetric rendering in Eq. 1.

Note that in general there is no way to guarantee that the inverse root finding converges. In practice, root finding fails for 1% to 8% of the points in the view space, making it impossible to query their attributes. For these points, an option is to just simply set their densities to zero, which would only be problematic if the points are close to the surface. A slightly better way is to estimate the color and density for those points by interpolating the attributes from their nearest valid neighbors along the ray. We conduct experiments on both strategies in Sec. 4.2, which results in slightly better performance. We choose the second strategy in our full model.

3.5 Establishing Correspondences

As our method is endowed with a 3D canonical space, we have the ability to trace surface correspondences across different views and poses. When rendering an image using Eq. 1, besides accumulating colors $\{\mathbf{c}_m\}$ of the samples $\{\mathbf{x}_{v,m}\}$ along the ray $\mathbf{r}$, we also accumulate the corresponding canonical coordinates $\{\mathbf{x}_{c,m}\}$:

$$X(\mathbf{r}) = \sum_{m=1}^{N} T_m(1 - \exp(\sigma_m \delta_m))\mathbf{x}_{c,m}, \tag{13}$$

where $X(\mathbf{r}) \in \mathbb{R}^3$ is the coordinate in the canonical space that corresponds to the pixel of the ray. With that, for images under different poses / views, we can compute their *dense* pixel-to-pixel correspondences by matching $X(\mathbf{r})$ in the canonical space using the nearest neighbor algorithm.

3.6 Training loss

Besides the image loss $\mathcal{L}_{im}$ defined in Eq. 2, we also employ two auxiliary losses that help the training. Due to the fact that all the points along a bone should have the same transformation, we encourage the skinning weights $\mathbf{w}$ of samples $\bar{\mathbf{x}}_c$ on the bones to be one-hot vectors $\hat{\mathbf{w}}$ (noted as $\mathcal{L}_w$). We also encourage the non-linear deformations $\boldsymbol{\Delta}_v$ of those samples to be zero given any pose $\mathbf{P}$ (noted as $\mathcal{L}_\Delta$). We use MSE to calculate both, $\mathcal{L}_w = ||\mathbf{w}(\bar{\mathbf{x}}_c) - \hat{\mathbf{w}})||_2^2$ and $\mathcal{L}_\Delta = ||\Delta_w(\bar{\mathbf{x}}_c, \mathbf{P}) - \mathbf{0})||_2^2$. Our final loss is: $\mathcal{L} = \mathcal{L}_{im} + \lambda\mathcal{L}_w + \beta\mathcal{L}_\Delta$, where λ is set to 1.0 and β is set to 0.1 in all our experiments.

4 Experiments

4.1 Datasets

We conduct experiments on 1) four human subjects (313, 315, 377, 386) in the ZJU-Mocap dataset [35], a public multi-view video dataset for human motion, and 2) two synthetic animal subjects (Hare, Wolf) introduced in this paper, rendered from multiple views using Blender.

Data Splits. Prior works [34,35] create the *train* and *val* sets on the ZJU-Mocap dataset by simply splitting each video with 500 $\sim$ 2200 frames into two splits, where the *training* set has 60 $\sim$ 300 frames and the *validation* set has 300 $\sim$ 1000 frames. This is not an ideal split to evaluate pose synthesis performance because 1) a training set with 60 consecutive frames in a 30fps video does not sufficiently cover pose variation to learn from, and 2) due to the repetitive motion of the actors, quite often similar poses are in both the *training* and *validation* sets, which should be avoided for evaluating a method on pose generalization. Therefore we establish a new protocol to split the dataset by clustering the frames based on pose similarity. Specifically, for each subject, we first randomly withhold a chunk of consecutive frames to be the *test* set, for the purpose of the final evaluation. Then, we use the K-Medoids algorithm on the remaining frames to cluster them into $K = 10$ clusters, based on pose similarity measured by the V2V Euclidean distance using ground-truth mesh. The most different cluster is selected as the val_{pose}^{ood} set, in which the frames are all considered to contain the *out-of-distribution* poses. For the remaining 9 clusters, we randomly split each cluster $2:1$ to form *train* and val_{pose}^{ind} sets, where the frames in val_{pose}^{ind} still contains new poses which are considered to be *in the distribution* of the training set. For the view splits, we follow the protocol from [34,35] for ZJU-Mocap, where 4 views are used for training and 17 views for testing. The animal subjects have 10 random views for training and 10 for testing. We denote val_{view} as our novel-view synthesis evaluation set, which contains all the training poses but rendered from different viewpoints. Please see the supplemental material for more details.

4.2 Evaluation and Comparison

Baselines. We compare our work with two types of previous methods: 1) Template-free methods, including NARF [29] and A-NeRF [42], as well as 2) SMPL-based methods, including Animatable-NeRF [34] and NeuralBody [35]. As our baseline, we use Pose-NeRF: we slightly modify Mip-NeRF [2] to learn the density and color conditioned on pose. We conduct experiments for all the moethods above on ZJU-Mocap, but exclude Animatable-NeRF and NeuralBody for the animal subjects (they require a template 3D model). Although code is available for each method, we noticed that each is using a different set of hyper-parameters for neural rendering (e.g., number of MLP layers, number of samples, near and far planes) and different training schedules, all of which are not related to method design but can greatly affect the performance. To make as-fair-as-possible comparisons, we integrated the template-free methods, NARF and A-NeRF, into our code base, which shares the same set of hyper-parameters[*]. For Animatable-NeRF and NeuralBody, we use the original implementations since their designs are based on the SMPL body template.

	Novel-view		Novel-pose (ind)		Novel-pose (ood)	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
SMPL-based Methods						
Animatable-NeRF [34]	30.75	0.971	29.34	0.966	28.67	0.961
NeuralBody [35]	**33.91**	**0.983**	**33.43**	**0.984**	**30.33**	**0.969**
Template-free Methods						
Pose-NeRF	31.88	0.975	32.09	0.976	28.43	0.954
A-NeRF [42]	32.45	0.978	32.65	0.978	30.41	0.967
NARF [29]	32.94	0.980	33.21	0.980	30.60	0.968
Ours	**33.11**	**0.981**	**33.35**	**0.981**	**30.69**	**0.969**

Table 2. *Comparisons on the ZJU Mocap subjects.* We compare with both, template-free and template-based methods.

Novel-view Synthesis. In this task, we conduct experiments on both, the ZJU-Mocap dataset and the two animal subjects Hare and Wolf, using val_{view} set. As shown in Tab. 3 and Tab. 2, our method outperforms other template-free methods measured by PSNR and SSIM. On the ZJU-Mocap dataset, our method achieves comparable performance with two template-based methods, Animatable-NeRF and NeuralBody, which greatly benefit from the SMPL body template, but do not work on other creatures like animals. See Fig. 4 for a qualitative comparison.

Novel-pose Synthesis. Due to the high interdependency of appearance changes caused by pose and motion, novel-pose synthesis is a more challenging task than novel-view synthesis, especially for poses that are out of the training distribution.

[*] For NARF, our re-implementation achieves better performance than it's official implementation. Please refer to the supplmental material for further details.

	Novel-view		Novel-pose (ind)			Novel-pose (ood)		
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	P2P ↓	PSNR ↑	SSIM ↑	P2P ↓
Pose-NeRF	23.40	0.974	21.93	0.941	197.11	16.62	0.925	88.85
A-NeRF [42]	31.26	0.976	31.22	0.977	31.52	25.66	0.967	19.04
NARF [29]	36.55	0.988	36.65	0.988	9.28	30.92	0.982	8.46
Ours	**37.30**	**0.991**	**37.45**	**0.991**	**4.30**	**35.77**	**0.990**	**3.38**

Table 3. *Comparisons on the animal subjects.* P2P is pixel-to-pixel error, for measuring image correspondences across different poses.

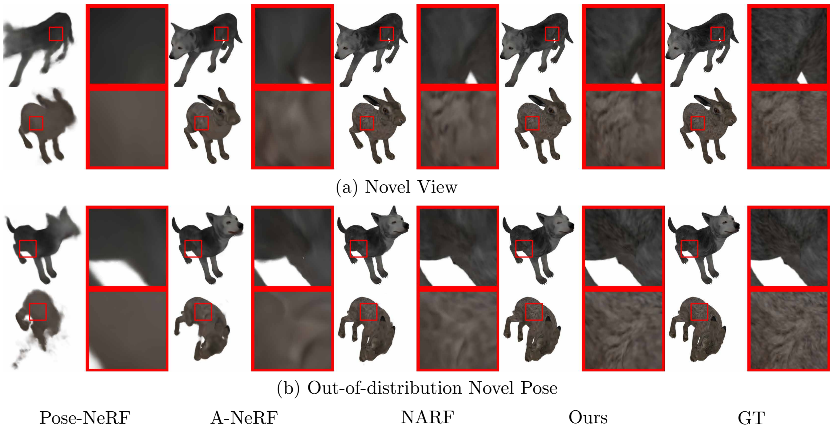
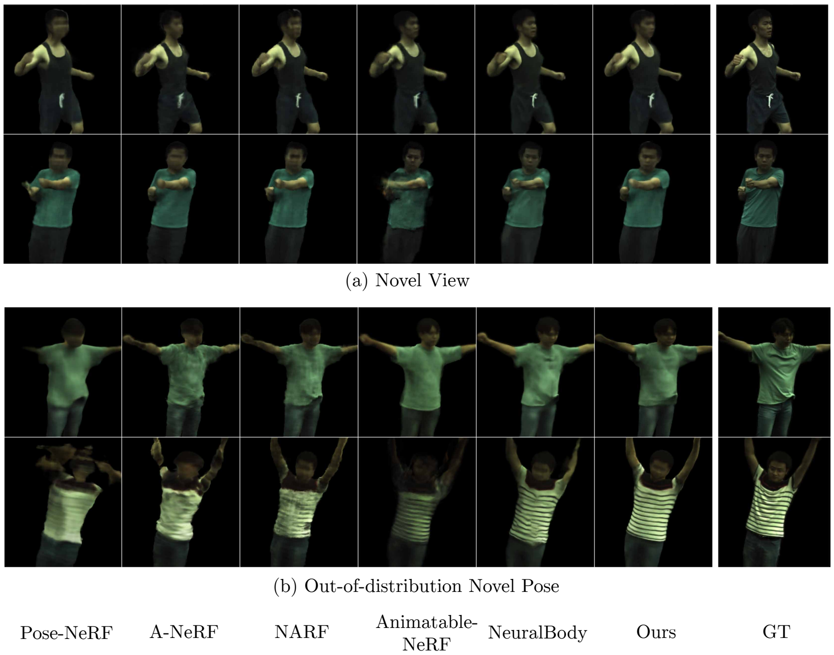


Fig. 3. *Comparison with template-free methods on the Hare and Wolf subjects.*

To carefully study this problem, we conduct experiments on both, in-distribution (InD) novel poses, using the val_{pose}^{ind} set, and out-of-distribution(OOD) novel poses, using the val_{pose}^{ood} set. Our experiments reveal that for InD poses, the performance of nearly all the methods are consistent with their performance on the novel-view task, as shown in Tabs. 2, 3. However, there is a huge drop in performance from InD poses to OOD pose (0.67db∼ 2.66db on ZJU-Mocap; 1.68db∼ 5.73db on animals). This is not surprising if the method contains neural networks that directly infer appearance information from pose input: generalization to vastly different pose inputs can not be expected. One of the main goals in this paper is to reduce this reliance of the neural networks to the pose input, for improving the robustness of the method to the OOD poses. Our method benefits from explicitly incorporating the *forward* LBS. We observe only an 1.68db performance drop comparing InD to OOD poses on the animal subjects, whereas other methods suffer from $\sim$ 5db performance drops, as shown in Tab. 3. Since these two synthetic subjects are not rendered with pose-dependent shading, and do not have "clothing" deformations, we disabled the ambient occlusion a (set to 1.) and non-linear deformation Δ_v (set to 0.) terms in our method during both, training and inference. These synthetic subjects allows us to study the underlying



(a) Novel View

(b) Out-of-distribution Novel Pose

Fig. 4. *Rendering quality comparison with all baseline methods on the ZJU-Mocap Dataset.* Note that Animatable-NeRF and NeuralBody rely on the SMPL body model, and the other approaches do not.

formulation of the articulation deformation, and we show here the *forward* LBS-based deformation is more reliable than the *inverse* deformation used in the baselines which takes pose as input to the MLP. The results on the ZJU-Mocap dataset in Tab. 2 show that our method outperforms both, the template-free and template-based methods, on the OOD poses. All the methods are prone to nearly the same drop in performance on the ZJU-Mocap dataset comparing InD to OOD poses. This is, because currently all of these methods, including ours, are still implicitly modeling pose-dependent shading effects (e.g., self-occlusion) as either a neural network or a latent code during training, which does not generalize well to OOD poses. Our method, though, provides a possibility to factor out the shading effects during inference and reveal the albedo color of the actor, which yields better generalization but is not suitable for evaluation comparing to the ground-truth, as shown in Fig. 8. See Figs. 3, 4 for qualitative comparisons.

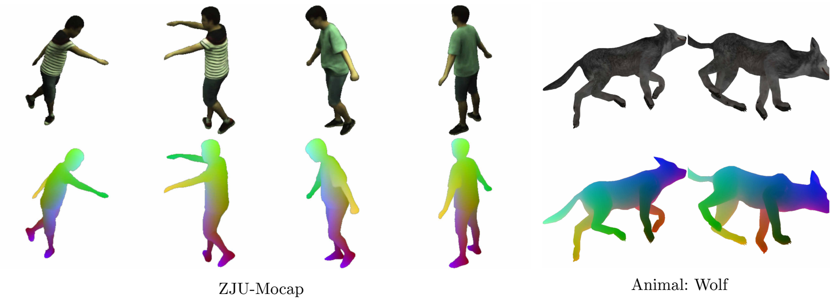


Fig. 5. *Rendering with Dense Correspondence.* We show results of our novel-view rendering with dense correspondences. On the ZJU Mocap dataset, correspondences across different subjects can also be built because they share the same canonical pose.

Pixel-to-Pixel Correspondences. We quantitatively evaluate correspondence on the animal subjects against Pose-NeRF, A-NeRF [42] and NARF [29]. We show qualitative results on ZJU-Mocap since no ground-truth correspondences are available (see Fig. 5). Even though neither of the baseline methods demonstrate that they can establish correspondences, we still tried our best to create a valid comparison[†]. For quantitative evaluation, we randomly sample 2000 image pairs (A, B) in val_{pose}^{ood} set, and use the ground truth mesh to establish ground-truth pixel-to-pixel correspondences $(\chi_A \rightarrow \chi_B)$ for every pair of images $(A \rightarrow B)$, where χ_A and χ_B are the corresponding image coordinates. Then, we use each method to render this pair of images, and find the correspondences of χ_A in B as χ_B^*. The pixel-to-pixel error (P2P) is then calculated as the average distance between χ_B and χ_B^*: P2P $= ||\chi_B - \chi_B^*||_2^2$. As shown in Tab. 3 and Fig. 6, our

[†] Pose-NeRF, A-NeRF and NARF all query the color and density of $(\mathbf{x}_v, \mathbf{P})$ in a higher dimensional (> 3) space, where we do the nearest neighbor matching for them using our approach as described in Sec. 3.5. Please refer to the supp.mat. for further details.

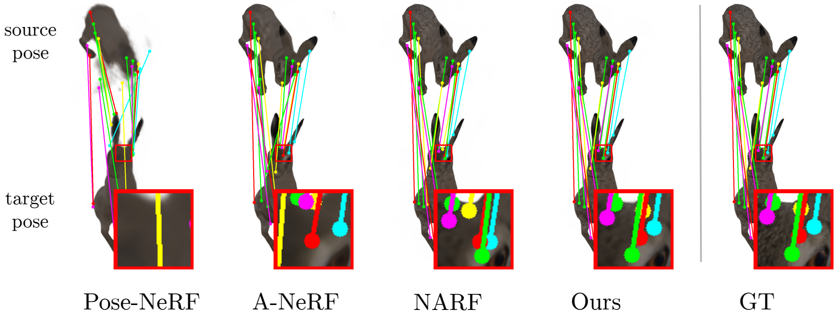


Fig. 6. *Correspondence comparison on Hare.* We find the correspondence for the same set of pixels in the source image in the target image. Both source and target are rendered in novel poses.

method achieves over 2x more accurate correspondences (3.38px v.s. 8.46px error in a 800 × 800 image) compared to the baselines. We visualize the extracted dense correspondences of our method in Fig. 5, which shows that correspondences across different subjects can be established as long as they share the same canonical pose (T-pose in ZJU-Mocap). Further more, we demonstrate that accurate correspondences can be used for content editing in Fig. 7.

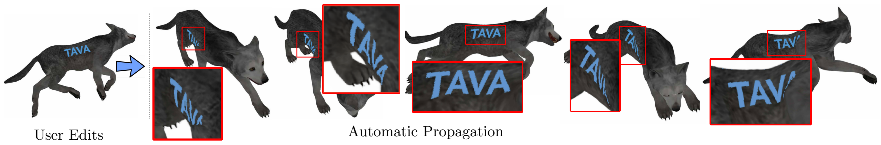


Fig. 7. *Rendering & Editing.* We show results of our novel-pose rendering with content editing. We manually attach a logo to the image on the left, then use our pixel-to-pixel correspondences to automatically propagate the logo to different poses & views.

4.3 Ablation Studies.

Thanks to our model design, we can train a full model with the non-linear deformation $\varDelta_v$ and the ambient occlusion AO enabled, then strip them out at inference time. Fig. 8 shows a qualitative result to visually demonstrate the impacts on the full model. Notice that without AO, the shading effects are removed during rendering, which produces overall brighter images than the ground-truth. This is an expected effect, but prohibits quantitative evaluation. Furthermore, we ablate these two design decisions during training. For the non-linear deformation, our ablation is to simply disable it during training to see if the LBS is enough to model the deformation. For the AO, our ablation is to

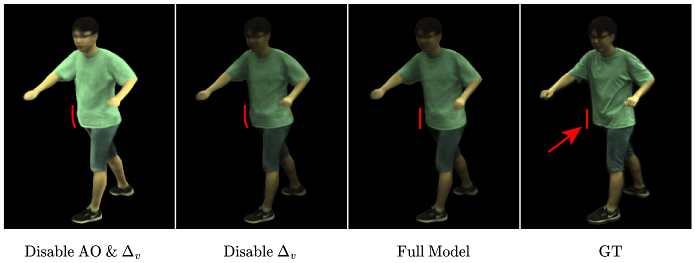


Fig. 8. *Ablation on the ambient occlusion (AO) and non-linear deformation (Δ_v) terms.* Due to our designs, we can train our full model with both enabled, then disable them during inference to ablate their effects.

compare it with predicting a pose-dependent color by conditioning pose to the color branch of F_{Θ_r}, and disabling the AO branch F_{Θ_a}. As shown in Tab. 4, both design decisions contribute to the final model performance. Lastly, we show the two different strategies to deal with root finding failures described in Section 3.4 in Tab. 4. Using the interpolation strategy results in slightly better performance.

	Novel-view		Novel-pose (ind)		Novel-pose (ood)	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
w/o non-linear Δ_v	31.86	0.974	32.19	0.975	30.56	0.965
w/o AO (pose-dep color)	32.94	0.980	33.20	0.980	30.57	0.968
w/o r.f. interplation	33.02	0.980	33.31	0.981	**30.72**	**0.969**
Ours	**33.11**	**0.981**	**33.35**	**0.981**	30.69	**0.969**

Table 4. *Model ablations on the ZJU Mocap subjects.*

5 Discussion

In this paper, we proposed a volumetric representation for articulated actors based on learned skinning, shape, and appearance. We also model pose-dependent deformation and shading effects. Extensive evaluations demonstrate that our approach consistently outperforms previous methods when generalizing to out-of-distribution unseen poses. Our approach can recover much more accurate dense correspondences across different poses and views than prior works, enabling content editing applications. Moreover, it does not require any body templates, enabling applications for creatures beyond humans. While our approach has clear advantages, there are few limitations. First, our method trains much slower (5 to 8 times) than the baselines, due to the nature of the root finding process for inverse deformation. A future direction could be to use invertible neural networks to avoid the root finding process. Second, even though our forward LBS ensures generalization to unseen poses, pose-dependent *non-linear* deformation and shading effects are still challenging to estimate correctly for unseen poses. Such effects are fundamentally challenging to model, particularly for lighting-dependent shading. An interesting future direction could be to model these effects across multiple subjects so that information from all subjects can be used to improve non-linear deformation model performance.

6 Acknowledgement

This work is partly supported by the CONIX Research Center, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH 2005 Papers, pp. 408–416 (2005)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: International Conference on Computer Vision (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR (2022)
4. Borshukov, G., Piponi, D., Larsen, O., Lewis, J.P., Tempelaar-Lietz, C.: Universal capture-image-based facial animation for" the matrix reloaded". In: Siggraph 2005 Courses (2005)
5. Carranza, J., Theobalt, C., Magnor, M.A., Seidel, H.P.: Free-viewpoint video of human actors. Transactions on Graphics (2003)
6. Casas, D., Volino, M., Collomosse, J., Hilton, A.: 4d video textures for interactive character appearance. In: Computer Graphics Forum (2014)
7. Chen, X., Zheng, Y., Black, M.J., Hilliges, O., Geiger, A.: Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11594–11604 (2021)
8. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Conference on Computer Vision and Pattern Recognition (2019)
9. Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., Sullivan, S.: High-quality streamable free-viewpoint video. Transactions on Graphics) (2015)
10. De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. In: ACM SIGGRAPH 2008 papers, pp. 1–10 (2008)
11. Deng, B., Lewis, J.P., Jeruzalski, T., Pons-Moll, G., Hinton, G., Norouzi, M., Tagliasacchi, A.: Nasa neural articulated shape approximation. In: European Conference on Computer Vision. pp. 612–628. Springer (2020)
12. Guo, K., Lincoln, P., Davidson, P., Busch, J., Yu, X., Whalen, M., Harvey, G., Orts-Escolano, S., Pandey, R., Dourgarian, J., et al.: The relightables: Volumetric performance capture of humans with realistic relighting. ACM Transactions on Graphics (ToG) **38**(6), 1–19 (2019)
13. Hasler, N., Thormählen, T., Rosenhahn, B., Seidel, H.P.: Learning skeletons for shape and pose. In: SIGGRAPH symposium on Interactive 3D Graphics and Games (2010)
14. Huang, Z., Xu, Y., Lassner, C., Li, H., Tung, T.: Arch: Animatable reconstruction of clothed humans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3093–3102 (2020)
15. James, D.L., Twigg, C.D.: Skinning mesh animations. Transactions on Graphics (2005)
16. Jiang, B., Zhang, J., Cai, J., Zheng, J.: Disentangled human body embedding based on deep hierarchical neural network. Transactions on Visualization and Computer Graphics (2020)
17. Li, H., Luo, L., Vlasic, D., Peers, P., Popović, J., Pauly, M., Rusinkiewicz, S.: Temporally coherent completion of dynamic shapes. ACM Transactions on Graphics (TOG) **31**(1), 1–11 (2012)

18. Li, K., Yang, J., Liu, L., Boulic, R., Lai, Y.K., Liu, Y., Li, Y., Molla, E.: Spa: Sparse photorealistic animation using a single rgb-d camera. Transactions on Circuits and Systems for Video Technology (2016)
19. Li, R., Bladin, K., Zhao, Y., Chinara, C., Ingraham, O., Xiang, P., Ren, X., Prasad, P., Kishore, B., Xing, J., et al.: Learning formation of physically-based face attributes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3410–3419 (2020)
20. Li, R., Xiu, Y., Saito, S., Huang, Z., Olszewski, K., Li, H.: Monocular real-time volumetric performance capture. In: European Conference on Computer Vision. pp. 49–67. Springer (2020)
21. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. In: Conference on Computer Vision and Pattern Recognition (2021)
22. Liu, L., Habermann, M., Rudnev, V., Sarkar, K., Gu, J., Theobalt, C.: Neural actor: Neural free-view synthesis of human actors with pose control. ACM Transactions on Graphics (TOG) **40**(6), 1–16 (2021)
23. Liu, S., Li, T., Chen, W., Li, H.: A general differentiable mesh renderer for image-based 3d reasoning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
24. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. ACM Trans. Graph. **38**(4), 65:1–65:14 (Jul 2019)
25. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. Transactions on Graphics (2015)
26. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Conference on Computer Vision and Pattern Recognition (2019)
27. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)
28. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4d reconstruction by learning particle dynamics. In: International conference on computer vision (2019)
29. Noguchi, A., Sun, X., Lin, S., Harada, T.: Neural articulated radiance field. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5762–5772 (2021)
30. Osman, A.A., Bolkart, T., Black, M.J.: Star: Sparse trained articulated human body regressor. In: European Conference on Computer Vision (2020)
31. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Conference on Computer Vision and Pattern Recognition (2019)
32. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. In: International Conference on Computer Vision (2021)
33. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. ACM Trans. Graph. **40**(6) (dec 2021)
34. Peng, S., Dong, J., Wang, Q., Zhang, S., Shuai, Q., Zhou, X., Bao, H.: Animatable neural radiance fields for modeling dynamic human bodies. In: International Conference on Computer Vision (2021)

35. Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9054–9063 (2021)
36. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Conference on Computer Vision and Pattern Recognition (2021)
37. Raj, A., Tanke, J., Hays, J., Vo, M., Stoll, C., Lassner, C.: Anr: Articulated neural rendering for virtual avatars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3722–3731 (2021)
38. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2304–2314 (2019)
39. Saito, S., Yang, J., Ma, Q., Black, M.J.: Scanimate: Weakly supervised learning of skinned clothed avatar networks. In: Conference on Computer Vision and Pattern Recognition (2021)
40. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems (2019)
41. Starck, J., Hilton, A.: Surface capture for performance-based animation. IEEE computer graphics and applications **27**(3), 21–31 (2007)
42. Su, S.Y., Yu, F., Zollhöfer, M., Rhodin, H.: A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. Advances in Neural Information Processing Systems **34** (2021)
43. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: International Conference on Computer Vision. IEEE (2021)
44. Volino, M., Casas, D., Collomosse, J.P., Hilton, A.: Optimal representation of multi-view video. In: British Machine Vision Conference (2014)
45. Weng, C.Y., Curless, B., Srinivasan, P.P., Barron, J.T., Kemelmacher-Shlizerman, I.: Humannerf: Free-viewpoint rendering of moving people from monocular video. CVPR (2022)
46. Xu, F., Liu, Y., Stoll, C., Tompkin, J., Bharaj, G., Dai, Q., Seidel, H.P., Kautz, J., Theobalt, C.: Video-based characters: creating new human performances from a multi-view video database. In: ACM SIGGRAPH 2011 papers (2011)
47. Xu, Z., Zhou, Y., Kalogerakis, E., Landreth, C., Singh, K.: Rignet: Neural rigging for articulated characters. Transactions on Graphics (2020)
48. Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Conference on Computer Vision and Pattern Recognition (2022)
49. Yu, A., Li, R., Tancik, M., Li, H., Ng, R., Kanazawa, A.: Plenoctrees for real-time rendering of neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5752–5761 (2021)
50. Zhi, T., Lassner, C., Tung, T., Stoll, C., Narasimhan, S.G., Vo, M.: Texmesh: Reconstructing detailed human texture and geometry from rgb-d video. In: European Conference on Computer Vision. pp. 492–509. Springer (2020)
51. Zhou, K., Bhatnagar, B.L., Pons-Moll, G.: Unsupervised shape and pose disentanglement for 3d meshes. In: European Conference on Computer Vision (2020)