Neural Actor: Neural Free-view Synthesis of Human Actors with Pose Control

LINGJIE LIU, Max Planck Institute for Informatics
MARC HABERMANN, Max Planck Institute for Informatics
VIKTOR RUDNEV, Max Planck Institute for Informatics
KRIPASINDHU SARKAR, Max Planck Institute for Informatics
JIATAO GU, Facebook AI Research
CHRISTIAN THEOBALT, Max Planck Institute for Informatics

Fig. 1. Novel view synthesis of an actor using *Neural Actor (NA)* under the control of novel poses, with the corresponding posed mesh models shown at lower right. All the poses are randomly sampled from the testing sequence.

We propose Neural Actor (NA), a new method for high-quality synthesis of humans from arbitrary viewpoints and under arbitrary controllable poses. Our method is built upon recent neural scene representation and rendering works which learn representations of geometry and appearance from only 2D images. While existing works demonstrated compelling rendering of static scenes and playback of dynamic scenes, photo-realistic reconstruction and rendering of humans with neural implicit methods, in particular under user-controlled novel poses, is still difficult. To address this problem, we utilize a coarse body model as the proxy to unwarp the surrounding 3D space into a canonical pose. A neural radiance field learns pose-dependent geometric deformations and pose- and view-dependent appearance effects in the canonical space from multi-view video input. To synthesize novel views of high-fidelity dynamic geometry and appearance, we leverage 2D texture maps defined on the body model as latent variables for predicting residual deformations and the dynamic appearance. Experiments demonstrate that our method achieves better quality than the state-of-the-arts on playback as well as novel pose synthesis, and can even generalize well to new poses that starkly differ from the training poses. Furthermore, our method also supports body shape control of the synthesized results. Please visit our project page for the full video: http://gvv.mpi-inf.mpg.de/projects/NeuralActor/.

Authors' addresses: Lingjie Liu, Max Planck Institute for Informatics, lliu@mpi-inf.mpg.de; Marc Habermann, Max Planck Institute for Informatics, mhaberma@mpi-inf.mpg.de; Viktor Rudnev, Max Planck Institute for Informatics, vrudnev@mpi-inf.mpg.de; Kripasindhu Sarkar, Max Planck Institute for Informatics, ksarkar@mpi-inf.mpg.de; Jiatao Gu, Facebook AI Research, jgu@fb.com; Christian Theobalt, Max Planck Institute for Informatics, theobalt@mpi-inf.mpg.de.

1 INTRODUCTION

Traditional methods for free-viewpoint video generation of humans from multi-view video employed passive photogrammetric methods or template fitting approaches to capture explicit model of the dynamic geometry and appearance of a moving human [Borshukov et al. 2005; Carranza et al. 2003; Casas et al. 2014; Collet et al. 2015; Li et al. 2014, 2017; Volino et al. 2014; Xu et al. 2011; Zitnick et al. 2004]. Novel views are synthesized with classical graphics renderers. Capturing such explicit moving human models from images is a very complex, time-consuming and potentially a brittle process. It is therefore hard to achieve photo-realistic free-viewpoint video quality for humans in general apparel. Furthermore, most of these techniques require animatable person-specific surface templates which need sophisticated reconstruction and rigging techniques for creating them.

Recently neural scene representation and neural rendering [Tewari et al. 2020c] were presented to overcome many limitation of the aforementioned earlier approaches based on explicit computer graphics modeling and rendering techniques. These methods implicitly learn representations of shape and appearance from images that can be rendered from new viewpoints without requiring explicit computer graphics models. However, while current neural rendering approaches show compelling results on static scenes, applying them to high quality free-viewpoint rendering of humans in general clothing is still difficult, let alone under novel user-controlled poses.

In this paper, we present a new approach, *Neural Actor (NA)*, for high-quality free-viewpoint rendering of human actors in everyday attire. NA can not only play back captured long motion sequences

but also synthesize free-viewpoint animations under user-controlled novel pose sequences. NA takes as input multi-view images of a human actor as well as the tracked poses of the actor on the basis of a coarse parametric shape model (SMPL) [Loper et al. 2015]. One challenge we need to tackle is that simply extending existing neural representations with a pose vector conditioning is not enough (see Figure 5) for achieving high-quality pose-dependent renderings. Instead, we explicitly deform the space to the canonical pose space with an inverse skinning transformation using the SMPL model [Huang et al. 2020]. We then predict residual deformation [Park et al. 2020; Pumarola et al. 2020a; Tretschk et al. 2021] for each pose with a deformation network, followed by learning pose-conditioned neural radiance fields in the canonical space. This design enables us to efficiently handle large movements.

However, the above formulation can still lead to blurry rendering results (see 'NA w/o texture' in Figure 7). This is due to the complex dynamics of the surface, pose tracking errors, and the fact that due to other dynamic effects the mapping from the skeletal pose to dynamic geometry and appearance is not a bijection, which therefore cannot be learned reliably using a deterministic neural model, such as NeRF [Mildenhall et al. 2020]. Hence, we incorporate 2D texture maps defined on the SMPL model as latent variables into the scene representation to better capture pose-dependent local shape and appearance changes. The 2D texture maps can be obtained by back-projecting the training images to the SMPL model during training, and at test time, they are predicted by an image-to-image translation network with normal maps generated from the posed SMPL model as input. In summary, our contributions are:

- We propose *Neural Actor (NA)*, a new method for realistic free-view synthesis of moving human actors with dynamic geometry and appearance. It can play back captured motions even with a large number of poses and synthesize results under challenging new user-controlled pose sequences, and also supports body shape control in the synthesized results.
- NA utilizes a new strategy to learn dynamic radiance fields using a coarse parametric body model. It disentangles the movements into inverse skinning transformations and dynamic residual deformations where only the latter needs to be learned.
- NA achieves high-quality dynamic geometry and appearance prediction without blurry artifact by incorporating 2D texture maps defined on SMPL as latent variables.
- We captured a new multi-view human performance dataset with dense camera arrays, which contains four sequences of human actors performing various motions. We will make this dataset publicly available.

2 RELATED WORK

We review learning-based approaches to neural scene representations, neural rendering, and generative models for humans. There are earlier non-learning-based works for video-based character creation [Casas et al. 2014; Li et al. 2017; Volino et al. 2014; Xu et al. 2011] and free-viewpoint videos [Borshukov et al. 2005; Carranza et al. 2003; Collet et al. 2015; Li et al. 2014; Zitnick et al. 2004], which we omit as they are conceptually less related.

Neural Scene Representations and Rendering. Neural scene representations and rendering algorithms aim at learning scene representations for novel view synthesis from only 2D images. Related works in this area can be categorized into static and dynamic scenes. DeepVoxels [Sitzmann et al. 2019a] represents the static scene as voxel grids where learnable features are attached to. SRN [Sitzmann et al. 2019b] replaces the discretized representation with a continuous learnable function. Recently, NeRF [Mildenhall et al. 2020] and its sparse-voxel variant [Liu et al. 2020a] were proposed to model the scene as a continuous 5D function that maps each spatial point to the radiance emitted in each direction and applies classical volume rendering techniques to render images. All of these works focus only on static scenes, while our work targets a dynamic setting with arbitrary poses, which is more challenging to model.

There are many recent works for dynamic scene rendering. Thies et al. [2019] assume a coarse geometry of the scene is given and a neural texture is learned to synthesize novel views. Weng et al. [2020] specifically targets free-view point synthesis and pose control of a human just from a in-the-wild videos. While the setup is very challenging, their results are still not video-realistic. Neural Volumes [Lombardi et al. 2019] and its follow-up work [Wang et al. 2020] employ an encoder-decoder architecture to learn a compressed latent representation of a dynamic scene which allows to synthesize novel views by using interpolation in the latent space and volume rendering. Inspired by the recent success of neural radiance fields (NeRF), some works add a dedicated deformation network [Park et al. 2020; Pumarola et al. 2020b; Tretschk et al. 2021], scene flow fields [Li et al. 2020], or space-time neural irradiance fields [Xian et al. 2020] to handle non-rigid scene deformations. In contrast to our work, they mainly focus on novel view synthesis and time-interpolation while our method can also generate photorealistic results for *novel* poses that were not seen during training. Gafni et al. [2020b] demonstrated the use of scene representations for synthesizing novel head poses and facial expressions from a fixed camera view and also generalization across identities has been demonstrated [Raj et al. 2021b]. Our work focuses on free-viewpoint synthesis of novel full human body poses. Compared to modeling the appearance and dynamics of a human face, modeling entire articulated humans for rendering and novel pose synthesis is an even more challenging problem due to the articulated structure of the body, the appearance variations, self-occlusions, and severely articulated motions. Instead of using a single implicit function, Peng et al. [2021] propose a set of latent codes attached to a body model in order to replay character motions from arbitrary view points. Alternatively, Lombardi et al. [2021] propose a mixture of volume primitives to avoid unnecessary samples in empty space. Most of these works can only playback the same dynamic sequence of a scene under novel views. In contrast, we also model *novel poses* under novel views, which is a much harder task because the network has to generalize to new views *and* to new poses.

Generative Models for Humans. Recently, generative adversarial networks (GANs) have made great progress in generating photorealistic images of humans and human body parts. Approaches that convert controllable conditioning inputs into photo-realistic body parts have been proposed for eyes [Shrivastava et al. 2017], hands

[Mueller et al. 2018], and faces [Ghosh et al. 2020; Kim et al. 2018; Tewari et al. 2020a,b]. In the context of entire human bodies, many of the approaches formulate this task as an image-to-image mapping problem. Given an appearance, these methods maps the body pose in the form of renderings of a skeleton [Chan et al. 2019; Kappel et al. 2020; Kratzwald et al. 2017; Li et al. 2019; Pumarola et al. 2018; Shysheya et al. 2019; Siarohin et al. 2018; Zhu et al. 2019], projection of a dense human model [Grigor'ev et al. 2019; Liu et al. 2020b, 2019b,a; Neverova et al. 2018; Prokudin et al. 2021; Raj et al. 2021a; Sarkar et al. 2020; Wang et al. 2018], or joint position heatmaps [Aberman et al. 2019; Ma et al. 2017, 2018] to realistic human images. To better preserve the appearance from the reference image to the generated image, some methods [Liu et al. 2020b; Sarkar et al. 2020] first map the person's appearance from screen space to UV space and feed the rendering of the person in the target pose with the UV texture map into an image-to-image translation network. Instead, Textured Neural Avatar [Shysheya et al. 2019] learns a person-specific texture map implicitly through backpropagation. To model stochasticity and the ability of sampling random human appearance, [Esser et al. 2018; Lassner et al. 2017; Sarkar et al. 2021] use the Variational Auto-Encoder [Kingma and Welling 2014] framework conditioned on 2D pose. All these methods do not learn scene geometry and do not ensure multi-view consistency due to the 2D convolution kernels they use. Furthermore, the GAN-based translation methods show conspicuous "shower curtain effects", thus making them not suitable for free-viewpoint rendering. In contrast, our method learns multi-view consistent geometry that can be used by the standard ray-casting method to perform robust renderings across different camera views.

Recently, there are also works that explicitly or implicitly model the scene geometry. Wu et al. [2020] translate point clouds of the human performance into photoreal imagery from novel views. However, they can only replay the captured performance while we also focus on the synthesis of new performances. Habermann et al. [2021] jointly learn motion-dependent geometry as well as motion- and view-dependent dynamic textures from multi-view video. Although, they achieve high quality results, their method relies on a person-specific template which requires a 3D scanner and manual work for the rigging and skinning. In contrast, our approach leverages a coarse parametric model – SMPL which removes the need for a 3D scanner and the manual work and even allows reshaping of the actors body proportions.

3 NEURAL ACTOR

Problem Setup. Given a training set of K synchronized RGB videos capturing a human actor with T frames $\boldsymbol{\mathcal{I}} = \{\mathcal{I}_t^k\}$ along with its camera parameters $\boldsymbol{C} = \{C_t^k\}$ $(t = 1 \dots T, k = 1 \dots K)$, our goal is to build an animatable virtual character with pose-dependent geometry and appearance that can be driven by arbitrary poses and rendered from novel viewpoints at test time. Note that we do not consider background synthesis in this paper and thus we apply color keying to extract the foreground in the images. Since body poses are needed as input, we track the body pose $\boldsymbol{\rho}$ for each frame.

We first define pose-conditioned implicit representations based on the state-of-the-art novel view synthesis method – NeRF [Mildenhall et al. 2020] as follows:

$$F_\theta : (\boldsymbol{x}, \boldsymbol{d}; \boldsymbol{\rho}) \rightarrow (\boldsymbol{c}, \sigma) \tag{1}$$

where θ denotes the network parameters. This function gives the color $\boldsymbol{c} = (r, g, b)$ and density $\sigma \in \mathbb{R}_+$ at spatial location $\boldsymbol{x} \in \mathbb{R}^3$ and view direction $\boldsymbol{d} \in \mathbb{S}^2$, conditioned on a pose vector $\boldsymbol{\rho}$. Then, we apply the classical volume rendering technique to render an image $\mathcal{I}_{out}$ of a human actor controlled by pose $\boldsymbol{\rho}$ with camera C. Since this rendering process is differentiable, we can optimize F_θ by comparing $\mathcal{I}_{out}$ with the ground truth image $\mathcal{I}$ without 3D supervision.

Challenges. The first challenge is how to incorporate pose information into the learned scene representations. We observe that a naive design choice of F_θ by concatenating the pose vector $\boldsymbol{\rho}$ with $(\boldsymbol{x}, \boldsymbol{d})$ is not only inefficient for encoding a large number of poses into a single network for playback, but also extremely difficult to generalize to novel poses (see Figure 5). The second challenge is that learning dynamic geometric details and appearance of a moving human body from poses only is an under-constrained problem, because at any moment the dynamic geometric details and changing appearance of a moving human body, such as cloth wrinkles, are not completely determined by the skeletal pose at that moment. Also, due to inevitable pose estimation errors, the association between dynamics and skeletal poses is even more difficult to learn. The above issues often lead to blurry artifacts in the output images, especially when using a deterministic model such as NeRF.

To tackle these challenges, we improve the vanilla NeRF via *template-guided neural radiance fields*. First, our Neural Actor (NA) approach incorporates a deformable human body model (SMPL) [Loper et al. 2015] as a 3D proxy to deform implicit fields (§ 3.1). Second, to handle uncertainty in dynamic geometry and appearance, NA incorporates texture maps as latent variables (§ 3.2). The overall pipeline is shown in Figure 2.

3.1 Geometry-guided Deformable NeRF

Deformation. Recent studies [Park et al. 2020; Pumarola et al. 2020a; Tretschk et al. 2021] have shown the effectiveness of representing dynamic scenes by learning a deformation function $\Phi_t(\boldsymbol{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ to map every sample point $\boldsymbol{x}$ into a shared canonical space. By doing so, scenes across frames get connected through the canonical space as the common anchor, which improves training efficiency. However, restricted by the design choice of $\Phi_t(\boldsymbol{x})$, it is difficult for these methods to model relatively large movements efficiently and they show limited generalizability to novel poses. To overcome these drawbacks, we augment this deformation function by querying an attached human body model – SMPL [Loper et al. 2015]. SMPL is a skinned vertex-based model $(\mathcal{V}, \mathcal{F}, \mathcal{W})$ that represents a wide variety of body shapes in arbitrary human poses, where $\mathcal{V} \in \mathbb{R}^{N_V \times 3}$ are the N_V vertices, and $\mathcal{F} \in \{1 \dots N_V\}^{N_F \times 3}$ are the vertex indices defining the triangles of the surface. For each vertex $\boldsymbol{v} \in \mathcal{V}$, fixed skinning weights $\boldsymbol{\omega} \in \mathcal{W}$ are assigned, where $\sum_i \omega_i = 1, \omega_i \geq 0, \forall i$. Given a specific person (with fixed body shape), the SMPL model can be deformed according to the body pose vector $\boldsymbol{\rho}$ via Linear Blend Skinning [Jacobson et al. 2014]. Since we want to transform the space in arbitrary poses to the canonical pose space, we invoke an *inverse-skinning transformation* [Huang et al. 2020] to deform the

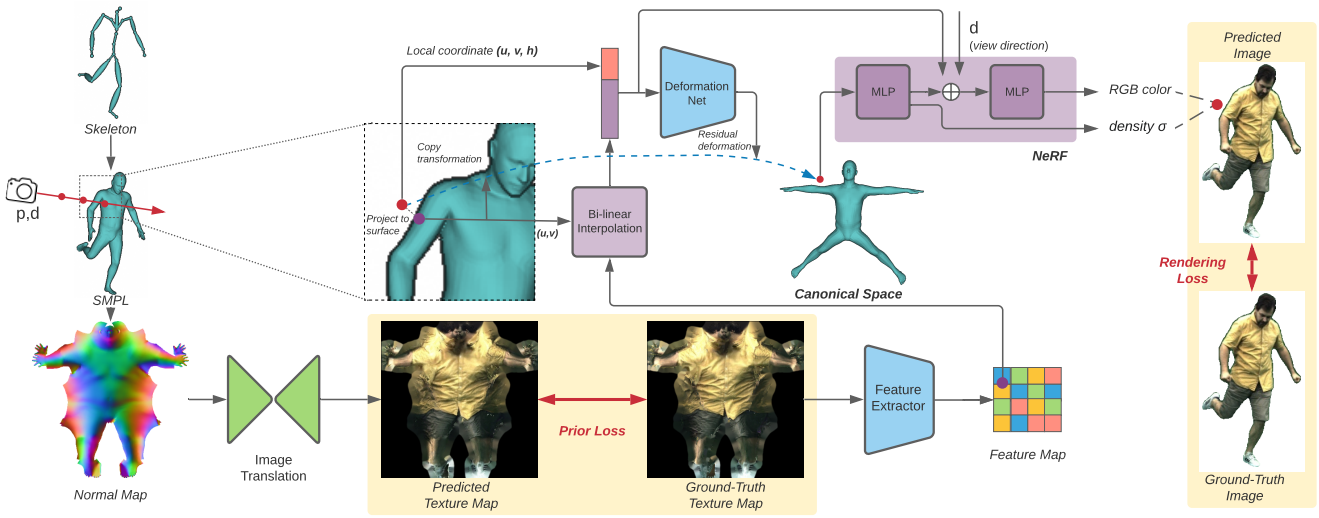

Fig. 2. Overview of *Neural Actor*. Given a pose, we synthesize images by sampling points along camera rays near the posed SMPL mesh. For each sampled point x, we assign to it the skinning weights of its nearest surface point and predict a residual deformation to transform x to the canonical space. We then learn the radiance field in the canonical pose space to predict the color and density for x using multi-view 2D supervision (§ 3.1). The pose-dependent residual deformation and color are predicted from the local coordinate of x along with the latent variables extracted from a 2D texture map of the nearest surface point of x. At training time, we use the ground truth texture map generated from multi-view training images to extract latent variables. At test time, the texture map is predicted from the normal map, which is extracted from the posed SMPL mesh via an image translation network, which is trained separately with the ground truth texture map as supervision (§ 3.2).

SMPL mesh of the pose $\boldsymbol{\rho}$ to the canonical pose space:

$$\Phi^{\text{SMPL}}(\boldsymbol{v}, \boldsymbol{\rho}, \boldsymbol{\omega}) = \sum_{j=1}^{N_J} \omega_j \cdot \left(R^j \boldsymbol{v} + \boldsymbol{t}^j\right), \tag{2}$$

where $(R^j, \boldsymbol{t}^j)$ denotes the rotation and translation at each joint j that transforms the joints back to the canonical space. Although Equation (2) is only defined on the surface of SMPL, it can be extended to any spatial point in pose $\boldsymbol{\rho}$ by simply copying the transformation from the nearest point on the surface of SMPL. For any spatial point $\boldsymbol{x}$ in the space of the pose $\boldsymbol{\rho}$, we find the nearest point on the SMPL surface as follows:

$$(u^*, v^*, f^*) = \arg\min_{u,v,f} \|\boldsymbol{x} - B_{u,v}\left(\mathcal{V}_{[\mathcal{F}(f)]}\right)\|_2^2, \tag{3}$$

where $f \in \{1 \dots N_F\}$ is the index of triangles, $\mathcal{V}_{[\mathcal{F}(f)]}$ is the three vertices of the triangle $F(f)$, and $(u, v) : u, v, u + v \in [0, 1]$ are the barycentric coordinates on the face. $B_{u,v}(.)$ is the barycentric interpolation function.

Next, we model the pose-dependent non-rigid deformation which cannot be captured by standard skinning using a residual function $\Delta\Phi_\theta(\boldsymbol{x}, \boldsymbol{\rho})$ similar to [Pumarola et al. 2020a; Tretschk et al. 2021]. The full deformation model can be represented as:

$$\Phi_\theta(\boldsymbol{x}, \boldsymbol{\rho}) = \Phi^{\text{SMPL}}(\boldsymbol{x}, \boldsymbol{\rho}, \boldsymbol{\omega}^*) + \Delta\Phi_\theta(\boldsymbol{x}, \boldsymbol{\rho}), \tag{4}$$

where $\boldsymbol{\omega}^* = B_{u^*,v^*}\left(\mathcal{W}_{[\mathcal{F}(f^*)]}\right)$ are the corresponding skinning weights from the nearest surface point. The full model allows us to pose the mesh via skinning and to model non-rigid deformations with the residual function. With this design, the learning of dynamic geometry is more efficient since we just need to learn a residual deformation for each pose. Furthermore, $\Delta\Phi_\theta(\boldsymbol{x}, \boldsymbol{\rho})$ acts as a compensation to unavoidable tracking errors for marker-less motion capture.

Rendering. Once the sampled points are deformed into the canonical space, we learn NeRF in this space following Equation (1). The final pixel color is predicted through volume rendering [Kajiya and Von Herzen 1984] with N consecutive samples $\{\boldsymbol{x}_1, \dots \boldsymbol{x}_N\}$ along the ray $\boldsymbol{r}$:

$$\mathcal{I}(\boldsymbol{r}, \boldsymbol{\rho}) = \sum_{n=1}^{N} \left(\prod_{m=1}^{n-1} e^{-\sigma_m \cdot \delta_m}\right) \cdot \left(1 - e^{-\sigma_n \cdot \delta_n}\right) \cdot \boldsymbol{c}_n, \tag{5}$$

where $\sigma_n = \sigma(\Phi_\theta(\boldsymbol{x}_n, \boldsymbol{\rho}))$, $\boldsymbol{c}_n = \boldsymbol{c}\left(\Phi_\theta(\boldsymbol{x}_n, \boldsymbol{\rho}), \boldsymbol{d}, \boldsymbol{\rho}\right)$ and $\delta_n = \|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\|_2$. Note that we use the deformed points alone to estimate densities (σ) to enforce learning the shared space, while including pose $\boldsymbol{\rho}$ to predict colors ($\boldsymbol{c}$) with pose-dependent phenomena (e.g. shadows).

Note that NeRF adopts the hierarchical sampling strategy: the second stage samples more points where the initial uniform samples have a higher probability. We interpret this as sampling based on the geometry learned in the first stage. In our setting, since the SMPL mesh is given, we adopt a geometry-guided ray marching process to speed up the volume rendering process. As shown in Figure 3, we take uniform samples but only accept samples $\boldsymbol{x}$ if $\min_{\boldsymbol{v} \in \mathcal{V}} \|\boldsymbol{x} - \boldsymbol{v}\|_2 < \gamma$, where γ is a hyperparameter which defines how close SMPL

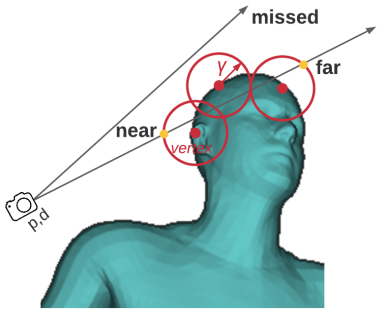

Fig. 3. Illustration of geometry-guided ray marching.

approximates the actual surface. More implementation details can be found in the Appendix.

3.2 Texture Map as Latent Variables

NeRF is only capable of learning a deterministic regression function, which makes it unsuitable to handle uncertainties involved in modeling dynamic details. As mentioned earlier, the mapping from the skeletal pose to dynamic appearance is not a bijection. Consequently, direct regression often leads to blurry outputs (see 'NA w/o texture' in Figure 7). A common approach is to incorporate latent variable z, i.e. $p(\sigma, \boldsymbol{c}|\boldsymbol{\rho}) = \int_z p(\sigma, \boldsymbol{c}|z, \boldsymbol{\rho}) \cdot p(z|\boldsymbol{\rho})dz$. As an example, we can choose z as spherical Gaussian and model the NeRF output $(\sigma, \boldsymbol{c})$ using conditional VAEs [Kingma and Welling 2013; Lombardi et al. 2019].

In contrast to common choices, we take the full advantage of the SMPL template and learn structure-aware latent variables. More specifically, we take a 2D texture map $\mathcal{Z} \in \mathbb{R}^{H\times W\times C}$ as a latent variable, which is defined based on a fixed UV parameterization $\mathcal{A} \in [0,1]^{N_F\times 3\times 2}$ which maps points on the 3D mesh surface to a 2D UV plane. There are three advantages of choosing $\mathcal{Z}$ in such a way:

(1) Compared to a compressed representation (e.g. latent vectors used in [Lombardi et al. 2019]), the texture map has higher resolution, making it possible to capture local details. The local information can be used to infer the local geometry and appearance changes in the scene representations.
(2) A simple posterior $q(\mathcal{Z}|\boldsymbol{I}, \boldsymbol{\rho})$ is available. That is, during training, the texture map $\mathcal{Z}$ for each training frame can be obtained by back-projecting the training images of each frame to all visible vertices and generate the final texture map by calculating the median of the most orthogonal texels from all views, as done in Alldieck et al. [2018]. As we do not need to update q, learning of $p(\sigma, \boldsymbol{c}|\boldsymbol{\rho})$ can readily be split into two parts, learning of the prior $p(\mathcal{Z}|\boldsymbol{\rho})$, and learning of the rendering $p(\boldsymbol{I}|\mathcal{Z}, \boldsymbol{\rho})$.
(3) Inspired by Liu et al. [2020b], learning of the prior model $p(\mathcal{Z}|\boldsymbol{\rho})$ can be formulated as an image-to-image translation task which maps the normal maps generated from the posed mesh to texture maps. To better preserve temporal consistency, we use vid2vid [Wang et al. 2018] to predict high-resolution texture maps from normal maps. The GAN loss of vid2vid enables learning from the distribution.

As shown in Figure 2, we apply an additional feature extractor $G(.)$ after $\mathcal{Z}$ to extract high-level features of the surface appearance that contains significantly more information than the RGB values of the texture maps. For any point x, its pose-dependent local properties depend on the extracted features of $\mathcal{Z}$ at its nearest surface point searched through Equation (3) and its local coordinate (u, v, h) where (u, v) is the texture coordinate of the nearest surface point and h is the signed distance to the surface. The feature extractor is trained together with the geometry-guided fields (§ 3.1) for predicting both residual deformations (Equation (4)) and dynamic appearance (Equation (5)).

4 EXPERIMENTS

Datasets. To validate our approach, we tested on eight sequences from three different datasets[1], including one captured by ourselves, which contain different actors wearing various textured clothing. We used two sequences, $S1$ and $S2$, from the *DeepCap* dataset [Habermann et al. 2020] which contains 11 and 12 cameras, respectively, at a resolution of 1024×1024. $S1$ contains 38,194 training frames and 23,062 testing frames; $S2$ has $33,605$ training frames and 23,062 testing frames. We also evaluated on two sequences, $D1$ and $D2$, from the *DynaCap* dataset [Habermann et al. 2021], under a dense camera setup with a resolution of 1285×940. The two sequences have approximately 20,000 and 7,000 frames for training and testing, respectively. For the *D1* sequence, we used 43 cameras for training and 4 uniformly distributed held-out cameras for the evaluation of our method on novel camera views; for the *D2* sequence, we used 100 cameras for training. To further evaluate our method on a wider variety of body poses and more challenging textured clothing, we captured a new multi-view human performance corpus with $79-86$ cameras at a resolution of 1285×940. It contains four sequences, $N1$-$N4$, and each has $12{,}000-16{,}000$ frames for training and around $8,000$ frames for testing. We will make the dataset publicly available. See Figure 12 in the Appendix for more details about the datasets.

In addition, to demonstrate the generalizability of the proposed method, we additionally test our method with various dancing motions from the *AIST* dataset [Li et al. 2021; Tsuchida et al. 2019] as the driving poses. Note that the dancing poses are quite distinct from the training poses making the reenactment task challenging.

Data Processing. Since we are only interested in foreground synthesis, we use color keying to extract the foreground in each image. We then employ an off-the-shelf SMPL tracking system[2] to optimize the shape parameters of SMPL as well as the global translation and the SMPL's pose parameters (a 72-dimension vector). The pose parameters include the root orientation and the axis-angle representations of the relative rotations of 23 body parts with respect to its parent in the kinematic tree. We further normalize the global translation for camera positions and the tracked geometry in our model as well as the baseline models.

We follow the standard texture generation step following [Alldieck et al. 2018]. to generate normal maps and ground truth textures maps for training the image translation network. In our early development, we observed in the experiments that the boundary pixels in the UV

[1] All data capture and evaluation of these sequences was performed at MPII by MPII.
[2] https://github.com/zju3dv/EasyMocap

Fig. 4. Qualitative reenactment results with the driving poses from the *DeepCap* test set and the *AIST* dataset. Note that our method can synthesize photorealistic images of human characters even for the unseen poses and views that strongly differ from the training poses.

space cannot preserve continuity in the 3D space, which would affect the texture feature extraction stage in our method. We take two measures to alleviate this problem. First, we cut the seam of the SMPL mesh in Blender and unwarp the mesh into one piece in the UV space. Second, we perform inpainting on the dilated region of the generated texture maps. An example of the resulted texture map is presented in Figure 2.

Implementation Details. We model the residual deformation networks $\Delta\Phi$ as 2-layer MLPs, and follow the network design of NeRF [Mildenhall et al. 2020] to predict density and color of each spatial location in the canonical space. We apply positional encoding to spatial location with a maximum frequency of $L = 6$. The texture map $\mathcal{Z}$ is at a resolution of 512×512. We use the backbone of ResNet34, which was pre-trained on ImageNet, as the texture feature extractor $G(.)$ to extract features from texture maps. We extract feature maps prior to the first 4 pooling layers, upsample them using bilinear interpolation and concatenate them to form multi-level features as the output $G(\mathcal{Z})$ in $256 \times 256 \times 512$, similar to [Yu et al.

	D1			*D2*			*S1*			*S2*		
Models	PSNR↑	LPIPS↓	FID ↓	PSNR↑	LPIPS↓	FID ↓	PSNR↑	LPIPS↓	FID↓	PSNR↑	LPIPS↓	FID↓
NeRF + pose	22.791	0.156	146.14	23.339	0.123	134.22	22.328	0.158	126.44	23.445	0.134	112.11
NeuralVolumes	20.648	0.171	135.57	21.020	0.143	122.36	18.661	0.190	123.04	19.076	0.173	98.063
NeuralBody	**23.768**	0.119	117.73	**23.872**	0.112	124.39	**22.967**	0.114	92.098	**23.946**	0.096	81.527
NHR	22.237	**0.075**	162.62	22.997	0.070	138.25	14.530	0.217	124.56	22.419	0.073	149.16
NeuralActor (Ours)	23.547	0.084	**44.921**	23.785	**0.065**	**46.812**	22.495	**0.084**	**34.361**	23.531	**0.066**	**19.714**
	N1			*N2*			*N3*			*N4*		
Models	PSNR↑	LPIPS↓	FID ↓	PSNR↑	LPIPS↓	FID ↓	PSNR↑	LPIPS↓	FID↓	PSNR↑	LPIPS↓	FID↓
NeRF + pose	22.892	0.174	125.83	23.922	0.142	126.34	24.621	0.113	106.30	23.648	0.162	153.30
NeuralVolumes	20.901	0.183	104.13	21.372	0.155	100.82	21.394	0.132	108.42	20.617	0.181	115.00
NeuralBody	**23.159**	0.153	107.72	**24.006**	0.115	91.218	**25.273**	0.093	76.124	**24.192**	0.140	111.78
NHR	21.630	0.098	117.42	22.806	**0.075**	175.00	23.719	0.062	91.535	22.744	0.092	164.23
NeuralActor (Ours)	22.799	**0.084**	**30.218**	24.345	0.080	**39.918**	25.014	**0.057**	**25.946**	23.861	**0.079**	**28.525**

Table 1. The quantitative comparisons on test poses of eight sequences. We use three metrics: PSNR, FID [Heusel et al. 2017] and LPIPS [Zhang et al. 2018] to evaluate the rendering quality. To reduce the influence of white background, all the scores are calculated from the images cropped with a maximum 2D bounding box which is estimated from the foreground masks of all the target images. The scores are averaged over all the training views of every 10^{th} test poses. Note that since PSNR is a metric based on least squares measurement, it does not faithfully measure image sharpness and so cannot properly account for the nuances of human visual perception [Zhang et al. 2018].

2020]. A detailed illustration of the proposed architecture is shown in Figure 11.

The texture feature extractor is trained together with the residual deformation network as well as NeRF in the canonical space on 8 Nvidia V100 32G GPUs for 300K iterations with a batch size of 1024 rays per GPU for around 2 days. For learning the prior, we used vid2vid [Wang et al. 2018][3] with default settings to predict texture maps at 512×512 pixels from normal maps in 512×512. We trained vid2vid on 4 Nvidia Quadro RTX 8000 48G GPUs with batchsize 4 per GPU for about 10K iterations for around 3 days. Since these two steps are independent, we can train them in parallel.

4.1 Qualitative Results

Since our model only requires the posed SMPL mesh as condition for novel view synthesis, it can be used for applications such as *reenactment* and *body-reshape*.

Reenactment. We directly use the pose parameters from the driving person and the shape parameters from the target person to get the posed SMPL mesh. Figure 4 shows example reenactment results where we use the testing poses from the *DeepCap* dataset [Habermann et al. 2020] and the AIST dataset [Li et al. 2021; Tsuchida et al. 2019] as driving poses, respectively. Our method successfully synthesizes faithful imagery of humans with fine-scale details in various motions and generalize well to challenging motions.

Body Reshape. As shown in Figure 6, we can adjust the shape parameters (PC1 and PC2) of the SMPL template to synthesize animations of the human in different shapes. Specifically, at inference time, we first warp the posed space to the canonical pose space for the reshaped human template via inverse kinematic transformation, and then transform the canonical pose space of the reshaped template to that of the original shape template and finally infer the color and density in the original canonical space. Such a technique will be potentially useful for the movie industry, e.g. we are able to synthesize animations of a giant or dwarf by modifying the shape parameters of any actor, without the need of finding the actual human in that shape. Please refer to the full videos at our project page for more results.

[3]https://github.com/NVlabs/imaginaire

Models	PSNR↑	LPIPS↓	FID↓
NeRF + pose	24.875	0.079	45.649
NeuralVolumes	24.248	0.149	131.86
NeuralBody	24.447	0.116	119.04
NHR	22.587	**0.072**	164.85
NeuralActor (Ours)	**24.875**	0.079	**45.649**
NeuralActor w. GT (Ours)	**27.567**	**0.071**	**43.089**

Table 2. Novel camera view synthesis on the training poses of D1

4.2 Comparisons

We validate the proposed method on two tasks: novel view synthesis and novel pose synthesis, comparing with recent baselines.

Novel Camera View Synthesis. For this comparison, we evaluate on the *D1* sequence, where 43 cameras are used for training and 4 uniformly distributed held-out cameras are used for evaluation. We compare our method with four state-of-the-art neural rendering methods. More precisely, we compare to:

- *NeRF+pose*: We extend NeRF [Mildenhall et al. 2020], a state-of-the-art view synthesis method for static scenes, to a pose-conditioned NeRF by providing the body pose as conditioning to the vanilla NeRF.
- *Neural Volumes (NV) [Lombardi et al. 2019]*: NV utilizes an encoder-decoder architecture to learn a latent representation of

Fig. 5. Qualitative comparisons of novel pose synthesis on eight sequences. Our method can faithfully recover the pose-dependent wrinkles and appearance details which cannot be achieved by other baseline methods.

a dynamic scene that enables it to produce novel content. We follow the original setting of NV, and provide it with the images captured from three uniformly distributed cameras at both training and test stages for each pose to encode the content into the latent space.

- *Neural Body (NB) [Peng et al. 2021]*: NB extends the vanilla NeRF by utilizing sparseCNNs to encode spatial features from the poses mesh. We follow the original setting of NB.
- *Multi-View Neural Human Rendering (NHR) [Wu et al. 2020]*: Different from NB, NHR extracts 3D features directly from the input point clouds and project them into 2D features. We use the vertices of SMPL model as input.

We show the quantitative results in Table 2 and include the visual results in the full video at our project page. For all the baseline methods, it is difficult to perform realistic rendering for playback when the training set contains a large number of different poses, e.g. pose sequences with 20K frames. Note that *NV* and *NB* have demonstrated good results in their work for playing back a short sequence, e.g. 300 frames; however, encoding a large number of frames, e.g. 20K frames into a single scene representation network tends to produce

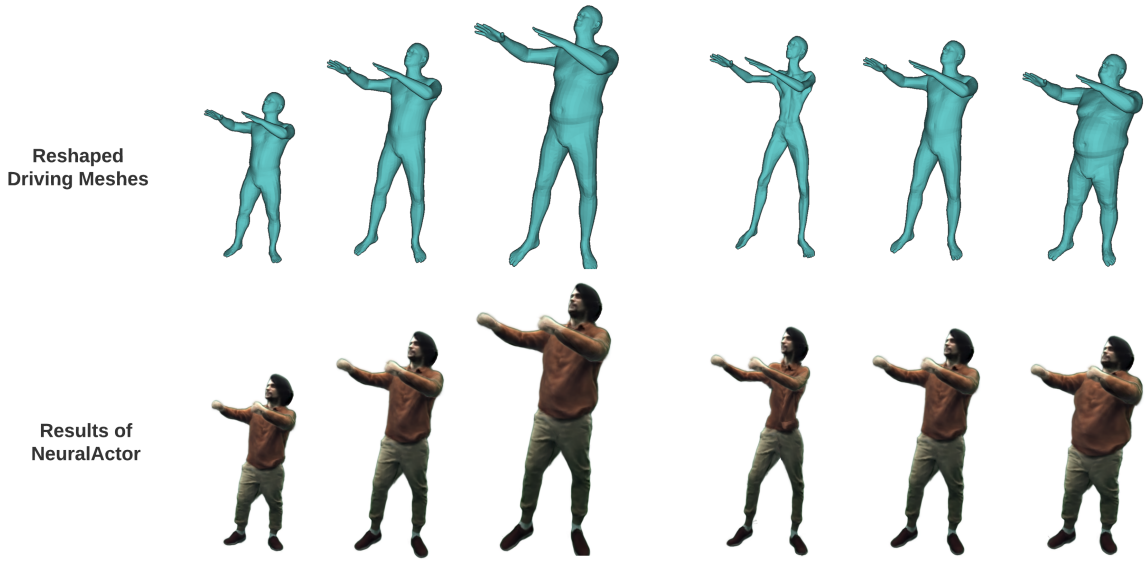

Fig. 6. Rendering results of our method for different body shape configurations of the same actor. Note that our method can produce photorealistic results for shapes that strongly differ from the original shape of the actor.

blurriness in the results due to the large variations in the training data. Simply feeding pose vectors into NeRF (similarly used in [Gafni et al. 2020a]) is not efficient for training since full deformations need to be learned. *NHR* also has difficulties in encoding a large number of various poses and leads to blurry results. In contrast, our method improves the training efficiency and resolves the blurriness issue by disentangling the full deformation into an inverse kinematic transformation and residual deformation and learning a prior with texture maps to resolve the blurriness issue. With these strategies, we can synthesize high-quality renderings of humans with sharp dynamic appearance even for the playback of a long sequence. We note that our results can further be improved when the multi-view images are provided for generating 'Ground Truth' texture maps at test time (see 'NA w. GT').

Novel Pose Synthesis. For novel pose synthesis, we first conduct the comparison with the above four baselines on eight sequences, where the test poses are used for evaluation. The qualitative and quantitative results are reported in Figure 5 and Table 1, respectively. *NeRF+pose* and *NV* produce severe artifacts in the results, such as missing body parts and blurriness. *NB* and *NHR* also suffer from blurriness and cannot preserve dynamic details in the results. Our method can achieve high-quality results with sharp details which are significantly better than the baseline methods.

We further compare with a recent mesh-based method, Real-time Deep Dynamic Characters (*DDC*) [Habermann et al. 2021], on the *D1* sequence. The original *DDC* requires a person-specific template captured by a 3D scanner. Since our method only needs SMPL model, as requested, the authors helped conduct a comparison with SMPL model as input (*DDC* with *SMPL*). We also provided the original result of *DDC* with the person-specific template for reference. As shown in the Figure 9, *DDC* works well with a person-specific template, however, deforming a coarse SMPL mesh is more challenging, which leads to artifacts on the deformed geometry, such as the head.

Our method is also related to Textured Neural Avatar (*TNA*). However, because its code and data are not available, we will just conceptually discuss the difference with that work. Different from our method, *TNA* is unable to synthesize the dynamic appearance of humans. Moreover, their results are not view-consistent and often suffer from artifacts such as missing body parts (see the full video at our project page).

4.3 Ablation Study

We conducted ablation studies on *D1* and evaluate on four test views for every 10th frame.

Effect of Texture Features. In Figure 7, we first analyze the effect of using texture features as latent variables. In our method, each sampled point is concatenated with the texture features extracted from the 2D texture map at its nearest surface point as conditioning for the prediction of the residual deformation and dynamic appearance. We compare with: 1) w/o texture: neither texture nor extracted texture features are provided. Here, we use the pose vector as conditioning; 2) w/o feature extractor: no feature extraction is performed on the 2D texture map, that is, the texture color of the nearest surface point is used as conditioning; and 3) w/o texture w/ normal: we extract the high-dimensional features on the normal map and use the features as conditioning.

We found that, compared to a compressed pose vector, the 2D texture map contains more spatial information, such as pose-dependent local details. Furthermore, the feature extractor can encode both local and global information and thus achieves better quality. We also observed that using the features directly extracted from the normal map

Models	PSNR↑	LPIPS↓	FID↓
NeuralActor (Full model)	23.547	**0.084**	**44.921**
w/o. texture inputs	**24.181**	0.131	108.30
raw texture inputs	23.110	0.142	105.45
normal map inputs	19.316	0.167	148.56

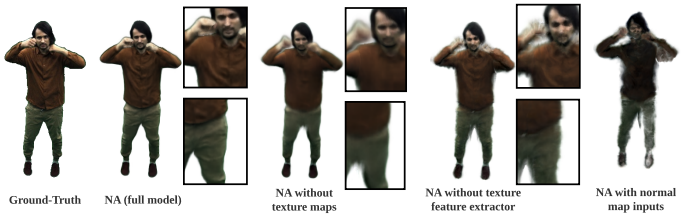

Fig. 7. Ablation on using texture features as latent variables. (Left) Quantitative results; (Right) Visual comparison.

Models	PSNR↑	LPIPS↓	FID↓
NeuralActor (Full model)	23.547	**0.084**	**44.921**
w/o residual deformation	23.532	0.093	56.580
w/o geometry guidance	21.635	0.137	72.379
using nearest vertex	**23.625**	0.092	70.768

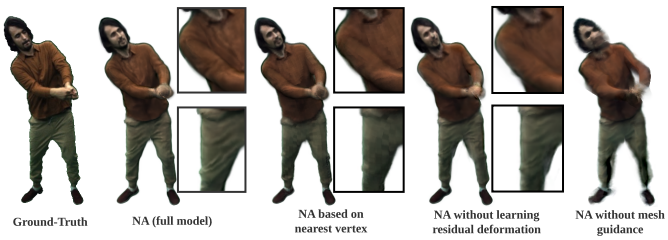

Fig. 8. Ablation on geometry-guided deformation prediction. (Left) Quantitative results; (Right) Visual comparison.

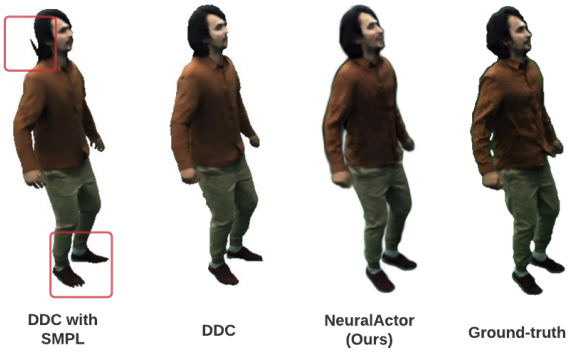

Fig. 9. Comparison to DDC. While the template-based approach DDC achieves a photoreal quality, it requires a personalized 3D scan of the actor and manual work is needed for the rigging and skinning. To bring their setting closer to ours, we also compare to DDC where we replace the template with the SMPL model. Note that naively applying their approach with a SMPL model results in geometric artifacts. In contrast, our approach achieves a similar quality to their original method even without requiring a personalized template.

results in very poor results. This is because the whole normal map can represent pose information while a single pixel on the normal map does not provide any information.

Effect of Geometry-guided Deformation. We further evaluated the effect of using the SMPL model as a 3D proxy to disentangle inverse kinematic transformations and residual non-rigid deformations. We compare with: 1) w/o residual deformations: the spatial point in the posed space transforms to the canonical space with only an inverse kinematic transformation; 2) w/o geometry guidance: we directly predict the full movements with the deformation network. As shown in Figure 8, modeling the full deformations as inverse kinematic transformations and the residual non-rigid deformations achieves the best quality. Directly learning full deformations is not efficient thus results in severe artifacts. We further compared copying the information (skinning weights and texture features) from the nearest point on the surface with copying from the nearest vertex. Since our body model is coarse, copying information from the nearest surface points leads to an improvement.

5 LIMITATIONS

Our proposed method leverages the SMPL model for unwarping to the canonical space. Consequently, our method can handle clothing types that roughly follow the topological structure of the SMPL model, but cannot handle more loose clothing such as skirts. Therefore future work is needed to leverage explicit cloth models on top of the SMPL model for the unwarping step. Our method is not able to faithfully generate the fingers (see Figure 10 for a failure example), since we use the SMPL model which does not model articulated hands, resulting in noisy texture for hands. In fact, even when using an improved human model (such as SMPL-X), robust hand synthesis can still be challenging because of the difficulty in accurately tracking hand gestrures due to the low resolution of hand images within a full body image. It will be our future work to study how to synthesize human characters with hands.

6 CONCLUSION

We presented Neural Actor, a new method for high-fidelity image synthesis of human characters from arbitrary viewpoints and under arbitrary controllable poses. To model moving human characters, we have utilized a coarse parametric body model as 3D proxy to unwarp

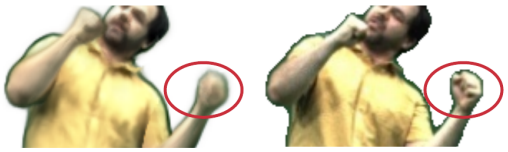

Fig. 10. A failure case of rendering hands. (Left) our result; (Right) ground-truth.

the 3D space surrounding the posed body mesh into a canonical pose space. Then a neural radiance field in the unwarped 3D space is used to learn pose-induced geometric deformations as well as both pose-induced and view-induced appearance effects in the canonical space. In addition, to synthesize high-fidelity dynamic geometry and appearance, we use 2D texture maps defined on the body model as latent variables for predicting residual deformations and the dynamic appearance. Extensive experiments demonstrated that our method outperforms the state-of-the-arts in terms of rendering quality and produces faithful pose-dependent appearance changes and wrinkle patterns. Furthermore, our method generalizes well to novel poses that starkly differ from the training poses, and supports the synthesis of human actors with controllable new shapes.

ACKNOWLEDGMENTS

We thank Oleksandr Sotnychenko, Kyaw Zaw Lin, Edgar Tretschk, Sida Peng, Shuai Qing and Xiaowei Zhou for the help; Jiayi Wang for the voice recording; MPII IST department for the technical support. Christian Theobalt was supported by ERC Consolidator Grant 770784. Lingjie Liu was supported by Lise Meitner Postdoctoral Fellowship.

REFERENCES

Kfir Aberman, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. 2019. Deep Video-Based Performance Cloning. *Comput. Graph. Forum* 38, 2 (2019), 219–233. https://doi.org/10.1111/cgf.13632

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018. Video Based Reconstruction of 3D People Models. In *IEEE Conference on Computer Vision and Pattern Recognition*.

George Borshukov, Dan Piponi, Oystein Larsen, John P Lewis, and Christina Tempelaar-Lietz. 2005. Universal capture-image-based facial animation for the matrix reloaded. In *ACM Siggraph 2005 Courses*. ACM, 16.

Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel. 2003. Free-Viewpoint Video of Human Actors. *ACM Trans. Graph.* 22, 3 (July 2003), 569—-577. https://doi.org/10.1145/882262.882309

Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 2014. 4D Video Textures for Interactive Character Appearance. *Comput. Graph. Forum* 33, 2 (May 2014), 371–380. https://doi.org/10.1111/cgf.12296

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody Dance Now. In *International Conference on Computer Vision (ICCV)*.

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 Transactions on Graphics (TOG)* 34, 4 (2015), 69.

Patrick Esser, Ekaterina Sutter, and Björn Ommer. 2018. A variational u-net for conditional appearance and shape generation. In *Computer Vision and Pattern Recognition (CVPR)*. 8857–8866.

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2020a. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. *https://arxiv.org/abs/2012.03065* (2020).

Guy Gafni, Justus Thies, Michael ZollhÃ¶fer, and Matthias NieÃŸner. 2020b. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. arXiv:2012.03065 [cs.CV]

Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael J. Black, and Timo Bolkart. 2020. GIF: Generative Interpretable Faces. In *International Conference on 3D Vision (3DV)*.

A. K. Grigor'ev, Artem Sevastopolsky, Alexander Vakhitov, and Victor S. Lempitsky. 2019. Coordinate-Based Texture Inpainting for Pose-Guided Human Image Generation. *Computer Vision and Pattern Recognition (CVPR)* (2019), 12127–12136.

Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2021. Real-time Deep Dynamic Characters. *ACM Transactions on Graphics* 40, 4, Article 94 (aug 2021).

Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2020. DeepCap: Monocular Human Performance Capture Using Weak Supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6629–6640.

Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. 2020. ARCH: Animatable Reconstruction of Clothed Humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. 2014. Skinning: Real-time Shape Deformation. In *ACM SIGGRAPH 2014 Courses*.

James T Kajiya and Brian P Von Herzen. 1984. Ray tracing volume densities. *ACM SIGGRAPH computer graphics* 18, 3 (1984), 165–174.

Moritz Kappel, Vladislav Golyanik, Mohamed Elgharib, Jann-Ole Henningson, Hans-Peter Seidel, Susana Castillo, Christian Theobalt, and Marcus Magnor. 2020. High-Fidelity Neural Human Motion Transfer from Monocular Video. arXiv:2012.10974 [cs.CV]

H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, N. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. 2018. Deep Video Portraits. *ACM Transactions on Graphics 2018 (TOG)* (2018).

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

Bernhard Kratzwald, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool. 2017. Towards an Understanding of Our World by GANing Videos in the Wild. (2017). https://arxiv.org/abs/1711.11453 arXiv:1711.11453.

Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. 2017. A Generative Model for People in Clothing. In *Proceedings of the IEEE International Conference on Computer Vision*.

Guannan Li, Yebin Liu, and Qionghai Dai. 2014. Free-viewpoint Video Relighting from Multi-view Sequence Under General Illumination. *Mach. Vision Appl.* 25, 7 (Oct. 2014), 1737–1746. https://doi.org/10.1007/s00138-013-0559-0

Kun Li, Jingyu Yang, Leijie Liu, Ronan Boulic, Yu-Kun Lai, Yebin Liu, Yubin Li, and Eray Molla. 2017. SPA: Sparse Photorealistic Animation Using a Single RGB-D Camera. *IEEE Trans. Cir. and Sys. for Video Technol.* 27, 4 (April 2017), 771–783. https://doi.org/10.1109/TCSVT.2016.2556419

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. 2021. Learn to Dance with AIST++: Music Conditioned 3D Dance Generation. arXiv:2101.08779 [cs.CV]

Yining Li, Chen Huang, and Chen Change Loy. 2019. Dense Intrinsic Appearance Flow for Human Pose Transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2020. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *ArXiv* abs/2011.13084 (2020).

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

Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhöfer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt. 2020b. Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation. *IEEE Transactions on Visualization and Computer Graphics* PP (05 2020), 1–1. https://doi.org/10.1109/TVCG.2020.2996594

Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. 2019b. Neural Rendering and Reenactment of Human Actor Videos. *ACM Transactions on Graphics (TOG)* (2019).

Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. 2019a. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5904–5913.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 65.

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of Volumetric Primitives for Efficient Neural

Rendering. arXiv:2103.01954 [cs.GR]
Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.
Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2017. Pose guided person image generation. In *Advances in Neural Information Processing Systems*. 405–415.
Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc van Gool, Bernt Schiele, and Mario Fritz. 2018. Disentangled Person Image Generation. *Computer Vision and Pattern Recognition (CVPR)* (2018).
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. *arXiv preprint arXiv:2003.08934* (2020).
Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 49–59.
Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. 2018. Dense Pose Transfer. *European Conference on Computer Vision (ECCV)* (2018).
Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2020. Deformable Neural Radiance Fields. *arXiv preprint arXiv:2011.12948* (2020).
Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *CVPR*.
Sergey Prokudin, Michael J. Black, and Javier Romero. 2021. SMPLpix: Neural Avatars from 3D Human Models. In *Winter Conference on Applications of Computer Vision (WACV)*. 1810–1819.
Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2018. Unsupervised Person Image Synthesis in Arbitrary Poses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2020a. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *arXiv preprint arXiv:2011.13961* (2020).
Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2020b. D-NeRF: Neural Radiance Fields for Dynamic Scenes. arXiv:2011.13961 [cs.CV]
Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. 2021a. ANR: Articulated Neural Rendering for Virtual Avatars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
Amit Raj, Michael Zollhoefer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. 2021b. PVA: Pixel-aligned Volumetric Avatars. arXiv:2101.02697 [cs.CV]
Kripasindhu Sarkar, Lingjie Liu, Vladislav Golyanik, and Christian Theobalt. 2021. HumanGAN: A Generative Model of Humans Images. arXiv:2103.06902 [cs.CV]
Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. 2020. Neural Re-Rendering of Humans from a Single Image. In *European Conference on Computer Vision (ECCV)*.
A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. 2017. Learning from Simulated and Unsupervised Images through Adversarial Training. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2242–2251. https://doi.org/10.1109/CVPR.2017.241
Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor Lempitsky. 2019. Textured Neural Avatars. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Aliaksandr Siarohin, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe. 2018. Deformable GANs for Pose-based Human Image Generation. In *CVPR 2018*.
Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Niessner, Gordon Wetzstein, and Michael Zollhofer. 2019a. DeepVoxels: Learning Persistent 3D Feature Embeddings. In *Computer Vision and Pattern Recognition (CVPR)*.
Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. 2019b. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*. 1119–1130.
Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. 2020a. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.
Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zöllhofer, and Christian Theobalt. 2020b. StyleRig: Rigging StyleGAN for 3D Control over Portrait Images, CVPR 2020. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.
Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. 2020c. State of the art on neural rendering. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 701–727.
Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. Graph.* 38 (2019), 66:1–66:12.
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. arXiv:2012.12247 [cs.CV]
Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. 2019. AIST Dance Video Database: Multi-genre, Multi-dancer, and Multi-camera Database for Dance Information Processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*. Delft, Netherlands, 501–510.
Marco Volino, Dan Casas, John Collomosse, and Adrian Hilton. 2014. Optimal Representation of Multiple View Video. In *Proceedings of the British Machine Vision Conference*. BMVA Press.
Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1152—1164.
Ziyan Wang, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael Zollhöfer. 2020. Learning Compositional Radiance Fields of Dynamic Human Heads. arXiv:2012.09955 [cs.CV]
Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. 2020. Vid2Actor: Free-viewpoint Animatable Person Synthesis from Video in the Wild. arXiv:2012.12884 [cs.CV]
Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. 2020. Multi-View Neural Human Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1682–1691.
Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2020. Space-time Neural Irradiance Fields for Free-Viewpoint Video. arXiv:2011.12950 [cs.CV]
Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. 2011. Video-based Characters: Creating New Human Performances from a Multi-view Video Database. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. ACM, New York, NY, USA, Article 32, 10 pages. https://doi.org/10.1145/1964921.1964927
Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2020. pixelNeRF: Neural Radiance Fields from One or Few Images. *arXiv preprint arXiv:2012.02190* (2020).
Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.
Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. 2019. Progressive Pose Attention Transfer for Person Image Generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2347–2356.
C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. 2004. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)* 23, 3 (2004), 600–608.

A ADDITIONAL IMPLEMENTATION DETAILS

A.1 Architecture

As shown in Figure 2, our method consists of 4 components: (1) an image translation network; (2) a texture feature extractor; (3) a deformation network; and (4) a NeRF model. We describe each component in detail as follows:

Image translation network. We adopt vid2vid [Wang et al. 2018] with default settings using the official implementation[4]. The size of the normal map and the texture map is 512×512.

Texture feature extractor. We use the feature extractor of ResNet34 backbone pretrained on ImageNet to extract features from texture maps. The extractor is jointly trained with the deformation network and the NeRF model on our dataset. We extract feature maps prior to the first 4 pooling layers, upsample using bilinear interpolation and concatenate them to form the feature maps of 512 channels.

Deformation network & NeRF.. See Figure 11 for the network architecture for these two components.

A.2 Algorithms

Volume rendering. As described in § 3.1, to speed up the rendering process, we adopt a geometry-guided ray marching process for volume rendering. See Algorithm 1 for implementation details. In our implementation, we set $N = 64$ and $\gamma = 0.06$ or 0.08 for all sequences.

ALGORITHM 1: Geometry-guided Ray Marching

Input: camera $\boldsymbol{p}_0$, ray direction $\boldsymbol{d}$, mesh vertices $\mathcal{V}$, γ, N
Initialize: $z_{\min} = +\infty, z_{\max} = -\infty$
for $\boldsymbol{v} \in \mathcal{V}$ **do**
  $z_0 = (\boldsymbol{v} - \boldsymbol{p}_0)^\top \cdot \boldsymbol{d}$
  if $\|\boldsymbol{v} - \boldsymbol{p}_0\|_2^2 - z_0^2 < \gamma^2$ **then**
    $\Delta z = \sqrt{\gamma^2 - \left(\|\boldsymbol{v} - \boldsymbol{p}_0\|_2^2 - z_0^2\right)}$
    if $z_0 + \Delta z > z_{\max}$ **then**
      $z_{\max} = z_0 + \Delta z$
    end
    if $z_0 - \Delta z < z_{\min}$ **then**
      $z_{\min} = z_0 - \Delta z$
    end
  end
end
if $z_{\min} < z_{\max}$ **then**
  Uniformly sample N points in $[z_{\min}, z_{\max}]$ and perform volume rendering.
end
else
  Ray missed the geometry. Abort.
end

Finding the nearest surface point. For each point on the ray, we search the nearest point and its (u, v) coordinate from the associated SMPL model following Alogrithm 2 where `Project_to_plane` and `Project_to_edge` are the functions of finding the nearest points on the planes and line segments, respectively.

We implement specialized CUDA kernels for both algorithms to achieve better efficiency.

ALGORITHM 2: Distance to Nearest Surface Point

Input: sampled point $\boldsymbol{x}$, mesh $\{\mathcal{V}, \mathcal{F}\}$
Initialize: $l_{\min} = +\infty$
for $\boldsymbol{f} \in \mathcal{F}$ **do**
  $\boldsymbol{x}_0$ = `Project_to_plane`$(\boldsymbol{x}, \boldsymbol{f})$
  if $\boldsymbol{x}_0$ *is inside* $\boldsymbol{f}$ **then**
    $l = \|\boldsymbol{x} - \boldsymbol{x}_0\|_2$
  end
  else
    $\boldsymbol{a} = \mathcal{V}_{[f_1]}$, $\boldsymbol{b} = \mathcal{V}_{[f_2]}$, $\boldsymbol{c} = \mathcal{V}_{[f_3]}$
    $\boldsymbol{x}_c$ = `Project_to_edge`$(\boldsymbol{x}, \boldsymbol{ab})$
    $\boldsymbol{x}_b$ = `Project_to_edge`$(\boldsymbol{x}, \boldsymbol{ac})$
    $\boldsymbol{x}_a$ = `Project_to_edge`$(\boldsymbol{x}, \boldsymbol{bc})$
    $l = \min_{\hat{\boldsymbol{x}} \in \{\boldsymbol{x}_a, \boldsymbol{x}_b, \boldsymbol{x}_c\}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2$
  end
  if $l < l_{\min}$ **then**
    $l_{\min} = l$
  end
end
return $l_{\min}$

B ADDITIONAL BASELINE SETTINGS

Neural Radiance Fields (NeRF) [Mildenhall et al. 2020] + pose. We extend the vanilla NeRF, which is designed for static scene rendering, to a pose-conditioned NeRF. Specifically, we concatenate the pose vector (72-dimension vector) with the positional encoding of (x, y, z) for each frame. We use a Pytorch reimplementation of NeRF[5] and follow the default hyper-parameters. For a fair comparison, we employ the same sampling strategy used in our method for NeRF, as described in appendix A.2. We train NeRF for 300K iterations on 8 GPUs with the same batch size as our model.

Neural Volumes (NV) [Lombardi et al. 2019]. We use the original code open-sourced by the authors[6]. We use batch size of 4 per GPU and 128×128 rays per image. We normalize the global translation of the scenes while keeping the rotation. All models on eight sequences were trained for 300K iterations on 4 GPUs. Since NV requires images to encode the scene content into a latent vector, we provide the images captured by three uniformly distributed cameras to obtain the latent vector for each pose at both training and testing stage.

Neural Body (NB) [Peng et al. 2021]. We follow the author provided code [7] and run all the experiments using the default training settings.

Multi-View Neural Human Rendering (NHR) [Wu et al. 2020]. We follow the author provided code [8] and run all the experiments using the default training settings.

[4] https://github.com/NVlabs/imaginaire

[5] https://github.com/facebookresearch/NSVF
[6] https://github.com/facebookresearch/neuralvolumes
[7] https://github.com/zju3dv/neuralbody
[8] https://github.com/wuminye/NHR

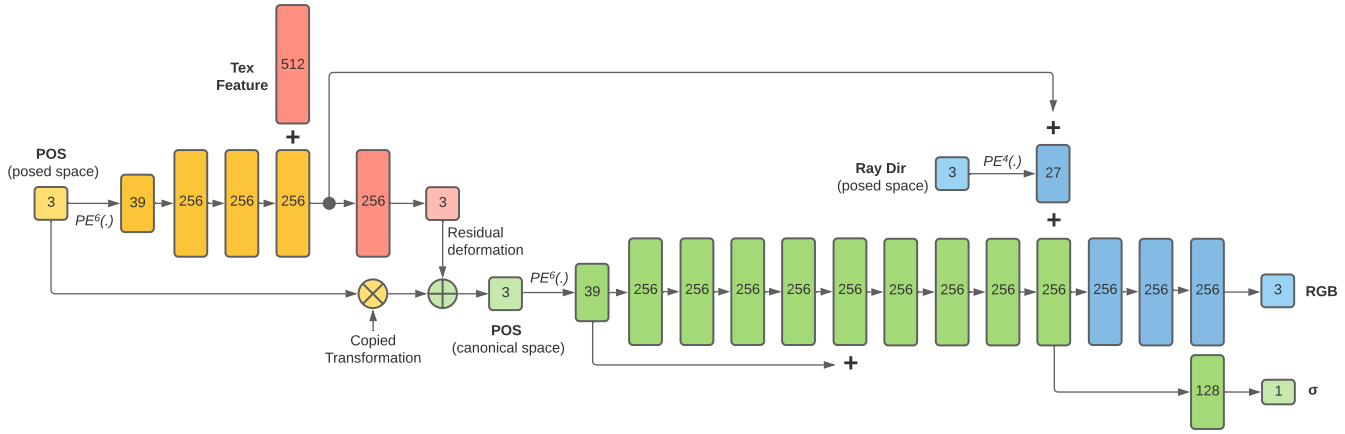

Fig. 11. A visualization of the architecture of the proposed deformation and NeRF networks. The number inside each block signifies the vector's dimension. All layers are standard fully-connected layers with ReLU activation except for the output layer where we do not use activation to predict deformation, density and colors (RGB). "+" denotes vector concatenation. The positional encoding function is defined as $\mathrm{PE}^L(\boldsymbol{x}) = \left[\boldsymbol{x}, \sin\left(2^0\boldsymbol{x}\right), \cos\left(2^0\boldsymbol{x}\right), \ldots, \sin\left(2^{L-1}\boldsymbol{x}\right), \cos\left(2^{L-1}\boldsymbol{x}\right)\right]$.

*Real-time Deep Dynamic Characters (*DDC*) [Habermann et al. 2021].* Training *DDC* has four stages: (1) EGNet was trained for 360,000 iterations with a batch size of 40, which takes 20 hours; (2) The lighting was optimized with a batch size of 4, a learning rate of 0.0001, and 30,000 iterations, which takes around 7 hours; (3) DeltaNet was trained for 360,000 iterations using a batch size of 8 and a learning rate of 0.0001 which takes 2 days. (4) TexNet was trained with a batch size of 12 and a learning rate of 0.0001 for 720,000 iterations for 4 days. These four networks were trained on 4 NVIDIA Quadro RTX 8000 with 48GB of memory.

C ADDITIONAL INFORMATION OF TRAINING DATA

The detailed information of the training data is included in Figure 12.

D1

Training Cameras: 43
Training Frames: 19500
Testing Frames: 6934

D2

Training Cameras: 100
Training Frames: 19000
Testing Frames: 7499

S1

Training Cameras: 12
Training Frames: 38194
Testing Frames: 23062

S2

Training Cameras: 11
Training Frames: 33605
Testing Frames: 14235

N1

Training Cameras: 86
Training Frames: 14619
Testing Frames: 7830

N2

Training Cameras: 79
Training Frames: 14158
Testing Frames: 7610

N3

Training Cameras: 85
Training Frames: 15398
Testing Frames: 7775

N4

Training Cameras: 80
Training Frames: 12881
Testing Frames: 7615

Fig. 12. Detailed information of eight sequences used in the experiments, containing two sequences (*D1* and *D2*) from the *DynaCap* dataset [Habermann et al. 2021] and two sequences (*S1* and *S2*) from the *DeepCap* dataset [Habermann et al. 2020] and four sequences captured by ourselves (*N1-4*). For all the images, the background has been removed and the camera parameters are given.