Learning Neural Volumetric Representations of Dynamic Humans in Minutes

Chen Geng*  Sida Peng*  Zhen Xu*  Hujun Bao  Xiaowei Zhou†
State Key Laboratory of CAD&CG, Zhejiang University

Figure 1. Convergence rate of training. Given a monocular video of a human performer, our model can be learned in ~5 minutes to produce photorealistic novel view rendering, which is 100 times faster than Neural Body [58].

Abstract

This paper addresses the challenge of efficiently reconstructing volumetric videos of dynamic humans from sparse multi-view videos. Some recent works represent a dynamic human as a canonical neural radiance field (NeRF) and a motion field, which are learned from input videos through differentiable rendering. But the per-scene optimization generally requires hours. Other generalizable NeRF models leverage learned prior from datasets to reduce the optimization time by only finetuning on new scenes at the cost of visual fidelity. In this paper, we propose a novel method for learning neural volumetric representations of dynamic humans in minutes with competitive visual quality. Specifically, we define a novel part-based voxelized human representation to better distribute the representational power of the network to different human parts. Furthermore, we propose a novel 2D motion parameterization scheme to increase the convergence rate of deformation field learning. Experiments demonstrate that our model can be learned 100 times faster than previous per-scene optimization methods while being competitive in the rendering quality. Training our model on a 512 × 512 video with 100 frames typically takes about 5 minutes on a single RTX 3090 GPU. The code is available on our project page: https://zju3dv.github.io/instant_nvr.

1. Introduction

Creating volumetric videos of human performers has many applications, such as immersive telepresence, video games, and movie production. Recently, some methods [58, 93] have shown that high-quality volumetric videos can be recovered from sparse multi-view videos by representing dynamic humans with neural scene representations. However, they typically require more than 10 hours of training on
a single GPU. The expensive time and computational costs limit the large-scale application of volumetric videos. Generalizable methods [34,100] utilize learned prior from datasets of dynamic humans to reduce the training time by only fine-tuning on novel human performers. These techniques could increase the optimization speed by a factor of 2-5 at the cost of some visual fidelity.

To speed up the process of optimizing a neural representation for view synthesis of dynamic humans, we analyze the structural prior of the human body and motion, and propose a novel dynamic human representation that achieves 100x speedup during optimization while maintaining competitive visual fidelity. Specifically, to model a dynamic human, we first transform world-space points to a canonical space using a novel motion parameterization scheme and inverse linear blend skinning (LBS) [35]. Then, the color and density of these points are estimated using the canonical human model.

The innovation of our proposed representation is two-fold. First, we observe that human body parts have different levels of complexity in terms of both shape and texture. For example, the face of a human performer typically exhibits higher complexity than a flatter textured torso region, thus requiring more representational power to depict. Motivated by this, our method decomposes the canonical human body into multiple parts and represents the human body with a structured set of voxelized NeRF [47] networks to bring the convergence rate of these different parts to the same level. In contrast to a single-resolution representation, the part-based body model utilizes the human body prior to represent the human shape and texture efficiently, heuristically distributing variational representational power to human parts with spatially varying complexity.

Second, we notice that non-rigid deformation of human geometry typically occurs around a surface instead of in a volume, that is, nearby surface points on a parametric human model tend to have similar motion behavior. Thus we propose a novel motion parameterization technique that models the 3D human deformation in a 2D domain, enabling modeling of the motion field using 3D voxelized representation to accelerate the learning. This idea is similar to the displacement map and bump map [14,15] in traditional computer graphics to represent detailed deformation on a 2D texture domain. We extend the technique of displacement map [14,15] to represent human motions by restricting the originally 3D deformation field [40,56,59] on the 2D surface of a parametric human model, such as SMPL [43]. This technique allows us to use hybrid representations [48,98] to model non-rigid deformation by reducing the memory footprint, largely boosting the convergence of the field.

Experiments demonstrate that our method significantly accelerates the optimization of neural human representations while being competitive with state-of-the-art human modeling methods on rendering quality. As shown in Figure 1, our model can be trained within 5 minutes to produce a volumetric video of a dynamic human from a 100-frame monocular video of a 512 × 512 resolution on an RTX 3090 GPU.

To summarize, our key contributions are:

• A novel part-based voxelized human representation for more efficient human body modeling.
• A 2D motion parameterization scheme for more efficient deformation field modeling.
• 100x speedup in optimization compared to previous neural human representations while maintaining competitive rendering quality.

2. Related work

Implicit neural representation and rendering. There have been many 3D scene representations, such as multi-view images [60,78], textured meshes [38,86], point clouds [1,61], and voxels [42,76]. Recently, some methods [11,41,44,51,77,97,104] propose implicit neural representations to represent scenes, which uses MLP networks to predict scene properties for any point in 3D space, such as occupancy [44,67], signed distance [41,51], and semantics [23,104]. This enables them to describe continuous and high-resolution 3D scenes. To perform novel view synthesis, neural radiance field (NeRF) [47] models scenes as implicit fields of density and color. NeRF is optimized from images with volume rendering techniques, which produces impressive image synthesis results. Many works improve NeRF in various aspects, including rendering quality [3,4], rendering speed [22,27,39,63,99], scene scale [64,82,89,94], and reconstruction quality [50,90,96]. Some methods [18,37,52,59] extend NeRF to dynamic scenes.

Human modeling. Reconstructing high-quality 3D human models is essential for synthesizing free-viewpoint videos of human performers. Traditional methods leverage multi-view stereo techniques [24,71,72] or depth fusion [13,17,80] to reconstruct human geometries, which require complicated hardware, such as dense camera arrays or depth sensors. To reduce the requirement of the capture equipment, some methods [2,67,68] train networks to learn human priors from datasets containing a large amount of 3D ground-truth human models, enabling them to infer human geometry and texture from even a single image. However, due to the limited diversity of training data, these methods do not generalize well to humans under complex poses. Recently, some methods [9,46,69,87] model the shapes of dynamic humans as implicit neural representations and attempt to optimize them from human scans. Another line of works [28,30,34,36,40,56,58,65,92,93,95,101–103,105] exploits dynamic implicit neural representations and differentiable renderers to reconstruct 3D human models from
accelerates the optimization process. This enables them to quickly fine-tune neural representations. The optimization process generally takes several hours on a modern GPU, which is time-consuming and costly to scale. Many differentiable rendering-based methods augment the approximation ability of networks by designing encoding techniques. Multiresolution hash encoding defines multiresolution feature vector arrays for a scene and uses the hash technique to assign each input coordinate a feature vector as the encoded input, which significantly improves the training speed.

### 3. Methods

This paper aims to quickly create a volumetric video from a sparse multi-view video that captures a dynamic human. We assume that the cameras are calibrated and the human pose and foreground human mask of each image are provided. Section 3.1 introduces our dynamic human model that is comprised of a part-based voxelized human representation and a dimensionality reduction motion parameterization scheme. Section 3.2 discusses how to efficiently optimize the proposed representation. Finally, Section 3.3 provides implementation details.

### 3.1. Proposed human representation

As shown in Figure 2, our dynamic human representation consists of a motion parameterization field and a part-based voxelized human model. (a) For a query point \( x \), the motion parameterization field first transforms it to the canonical space correspondence \( x_{\text{can}} \) using the inverse LBS algorithm and by parameterizing the 3D points onto 2D UV coordinates to predict the residual deformation \( \Delta x \). (b) Then, \( x_{\text{can}} \) is fed into the part-based voxelized human model in canonical space to predict and aggregate the density and color \((\sigma, c)\), where the canonical human body is decomposed into \( K \) parts, each of which is represented using an

![Figure 2. Overview of the proposed representation.](image-url)

Given a query point \( x \) at frame \( t \), we find its nearest surface point on each human part of the SMPL mesh, which gives the blend weight \( w_k \) and the UV coordinates \((u_k, v_k)\). Consider the \( k \)-th part. The motion field consists of an inverse LBS module and a residual deformation module. (a) The inverse LBS module takes body pose \( \rho \), blend weight \( w_k \), and query point \( x \) as input and outputs the transformed point \( x' \). The residual deformation module applies the multiresolution hash encoding (MHE) to \((u_k, v_k, t)\) and uses an MLP network to regress the residual translation \( \Delta x \), which is added to \( x' \) to obtain the canonical point \( x_{\text{can}} \). (b) We then feed \( x_{\text{can}} \) to networks of \( k \)-th human part to predict the density \( \sigma_k \) and color \( c_k \). With \( \{(\sigma_k, c_k)\}_{k=1}^{K} \), we select the one with the largest density as the density and color of the query point.

Videos. To represent dynamic humans, Neural Actor augments the neural radiance field with the linear blend skinning model. It additionally adopts a residual deformation field to better predict human motions. To overcome the inaccuracy of input human poses, optimize the parameters of human poses jointly with the human representations during training. These methods typically require a lengthy training process to produce high-quality human models. In contrast, we introduce a part-based voxelized human representation to model the canonical human body, which significantly accelerates the optimization process. Although have proposed part-based implicit functions, they focus on human shape modeling and do not show that the part-based representation can be used to reduce the training time.

**Accelerating the optimization of neural representations.** Many differentiable rendering-based methods optimize a separate neural representation for each scene. The optimization process generally takes several hours on a modern GPU, which is time-consuming and costly to scale. Inspired by multi-view stereo matching, some methods train a network on multi-view datasets to learn to infer radiance fields from input images. This enables them to quickly fine-tune neural representations to unseen scenes. leverage the auto-decoder to capture the scene priors for efficient fine-tuning. [5, 83] utilize meta-learning techniques to initialize network parameters, thereby improving the training speed. Some methods attempt to design scene representations that support efficient training. [21, 48, 75, 84]
Motion parameterization on 2D surface domain. To regress the canonical correspondence $x^{can}$ of a query point $x$, we first find its nearest surface point $p$ on the posed SMPL mesh. Using the strategy in [40], the blend weight $w$ and UV coordinate $(u, v)$ of surface point $p$ are obtained from the SMPL model.

Given the blend weight $w$ and UV coordinate $(u, v)$, the motion field maps the query point $x$ to the canonical space correspondence $x^{can}$. The motion field is comprised of an inverse LBS module [35] and a residual deformation module. Given a query point $x$ and blend weight $w$, we use the inverse LBS module to transform it to the unposed space, which is defined as:

$$\Phi_{LBS}(x, w, \rho) = \left( \sum_{j=1}^{N} w_{k,j} G_j \right)^{-1} x, \quad (1)$$

where $\rho$ denotes the human pose and $\{G_j\}_{j=1}^{N}$ are transformation matrices derived from $\rho$ [43]. The detailed derivation of the inverse LBS algorithm can be found in the supplementary material.

The transformed point $\Phi_{LBS}(x, w, \rho)$ is then deformed to the human surface using the residual deformation module. Specifically, the current time $t$ is first concatenated with the UV coordinate $(u, v)$ to serve as the parameterization of the query point $x$ at frame $t$. This motion parameterization is inspired by the displacement map and bump map techniques [14, 15] in the traditional Computer Graphics pipelines. It essentially reduces the dimensionality of a 4D space-time sequence down to the 3D surface-time domain utilizing the human deformation prior. Then, we apply the multiresolution hash encoding [48] $\psi_{\text{res}}$ to $(u, v, t)$ and forward the encoded input through a network $\text{MLP}_{\text{res}}$ to regress the residual $\Delta \Phi(u, v, t)$. The full human motion at frame $t$ is defined as:

$$\Delta \Phi(u, v, t) = \text{MLP}_{\text{res}}(\psi_{\text{res}}(u, v, t)), \quad (2)$$

$$x^{can}(x, w, u, v, \rho, t) = \Phi_{LBS}(x, w, \rho) + \Delta \Phi(u, v, t). \quad (3)$$

There are two main observations that inspired us to use the $(u, v, t)$ motion parameterization. First, we observe that a typical human motion happens at a surface level instead of a volumetric level. Near-surface points sharing similar UV coordinate of the parametric model shows similar motions. Utilizing this prior with a surface parameterization [14, 15], we can reduce the required 4D volumetric motion to the 3D surface-time domain, greatly decreasing the amount of information the deformation network has to learn. Based on a similar idea, [6] diffuses the surface motion to the full 3D space. Second, a naive $(x, y, z, t)$ encoding would introduce quartic memory overhead on an explicitly defined voxel structure, which is intractable to use in practice. Instead, by parameterizing the motion to $(u, v, t)$, we can reduce the memory footprint to a more practical cubic level. Experiments demonstrate that the motion parameterization scheme effectively reduces the dimensionality of the deformation field, thus greatly increasing the convergence rate of the human model.

Part-based voxelized human representation. Muller et al. [48] propose the multi-resolution hash encoding (MHE) to improve the approximation ability and training speed of implicit neural representations. MHE is defined on an explicit set of voxel grids of different resolutions. Given an input coordinate, it applies the hash encoding on each level and queries the corresponding voxel grid to trilinearly interpolate the feature of the input point for this level. Then, the concatenated multi-resolution feature is fed into a small MLP network to predict the target value. Note that [48] concatenates the features of multi-resolution hash encoding for the same point to mitigate the effect of hash collision, while our part-based voxelized human representation introduces spatially varying resolution to efficiently encode human parts with different complexity.

In contrast to [40, 56] which use a single neural radiance field (NeRF) to represent the canonical human model, we decompose the human body into multiple parts with different complexity and adopt a structured set of MHE-augmented NeRF with varying resolutions as the body representation. Specifically, we manually divide the human body into multiple parts based on a parametric human model (such as SMPL [43]), as shown in Figure 2. Note that other parametric human models [54, 66] can also be used in our method. We use the blend weights defined in SMPL model [43] to decompose SMPL template mesh $M = (V, E)$, where $V$ represents the vertices and $E$ represents the edges. Let the $i$-th vertex $v_i$ have the blend weight $w_i$ and for each part $k$ we define $\Omega_k$ as the set of bones that belong to this part. The detailed setting of $\Omega_k$ can be found in the supplementary material. The mesh of the $k$-th part is defined as $M_k = (V_k, E_k)$, where:

$$V_k = \{ v_i | \text{argmax } w_i \in \Omega_k \}, \quad (4)$$

$$E_k = \{ (v_i, v_j) | v_i \in V_k, v_j \in V_k \}. \quad (5)$$

To regress the density and color of a query point $x$, we first find the nearest surface point $p_k$ on each human part $M_k$ of the posed SMPL mesh. Using the strategy in [40], the blend weight $w_k$ and UV coordinate $(u_k, v_k)$ of surface point $p_k$ are obtained from the SMPL model. With $(u_k, v_k, t)$, we use the motion parameterization scheme defined in Section 3.1 to transform the query point to the space of the $k$-th human part. We redefine the parameters of the multiresolution hash encoding function $\psi_k$ for the $k$-th part.
Given the transformed point, we first apply the multiresolution hash encoding to the transformed point and then feed the encoded point \( ϕ_k(x) \) to a small NeRF network to predict the density and color. The density network \( MLP_{σ_k} \) is defined as:

\[
(σ_k, z) = MLP_{σ_k}(ϕ_k(x)),
\]

where \( σ_k \) means the density and \( z \) is a feature vector. Then, we take the feature vector \( z \) and the viewing direction \( d \) as the input for the color regression. Similar to [52], a latent embedding \( ℓ_t \) for each video frame \( t \) is introduced to model the temporally-varying appearance. The color network is defined as:

\[
c_k = MLP_{c_k}(z, d, ℓ_t).
\]

Finally, we have \( K \) predictions \( \{(σ_k, c_k)\}_{k=1}^K \). The density and color \( (σ, c) \) of the query point \( x \) is calculated based on:

\[
(σ, c) = (σ_k^*, c_k^*), \text{ where } k^* = \arg\max_k σ_k.
\]

In contrast to [40, 56] that represent the body with a single NeRF network, our part-based voxelized human representation can assign different densities of model parameters to different human parts with different complexity, thereby enabling us to efficiently distribute the representational power of the network. Experiments show that our proposed body representation significantly improves the rate of convergence.

### 3.2. Training

The proposed representation can be learned from sparse multi-view videos by minimizing the difference between rendered and observed images. The volume rendering technique [32, 47] is used to synthesize the pixel color. Give a pixel at frame \( t \), we emit a camera ray and sample points along the ray. Then, the sampled points are fed into the dynamic human representation to predict their colors and densities, which are finally accumulated into the pixel color. During each training iteration, we randomly sample an image patch from the input image and compute the Mean Squared Error (MSE) loss and perceptual loss [31] to train the model parameters, which are defined as:

\[
L_{\text{reg}} = \|\tilde{I}_P - I_P\|_2 + \|F_{\text{vgg}}(\tilde{I}_P) - F_{\text{vgg}}(I_P)\|_2,
\]

where \( \tilde{I}_P \) is the rendered image patch, \( I_P \) is the ground truth image patch, and \( F_{\text{vgg}} \) extracts image features using the pretrained VGG network [31]. Ablation study demonstrates that perceptual loss is essential for rendering quality and fast training.

In addition to the image rendering loss, two regularization techniques are used to facilitate the learning of the neural representations. First, we apply the regularizer in [4] to concentrate densities on the human surface. Second, the residual deformation field is regularized to be small and smooth. More details of the loss terms are described in the supplementary material.

### 3.3. Implementation details

We adopt the Adam optimizer [33] with a learning rate of \( 5e^{-4} \). We train our model on an RTX 3090 GPU, which takes around 5 minutes to produce photorealistic results. Our method is implemented purely with the PyTorch framework [53] to demonstrate the effectiveness of our representation. It also enables us to fairly compare with baseline methods [34, 56, 58] implemented in PyTorch. The details of the network architectures and hyper-parameters are presented in the supplementary material.

### 4. Experiments

#### 4.1. Datasets

- **ZJU-MoCap** [58] dataset is a widely-used benchmark for human modeling from videos. It provides foreground human masks and SMPL parameters. Following [93], we select 6 human subjects (377, 386, 387, 392, 393, 394) from this dataset to conduct our experiments. One camera is used for training, and the remaining cameras are used for evaluation. For each human subject, we select 1 frame every 5 frames and collect 100 frames for training. Please refer to the supplementary material for more detailed experiment settings of all characters.

- **MonoCap** dataset contains four multi-view videos collected by [57] from the DeepCap dataset [26] and the DynamicCap dataset [25]. It provides camera parameters and human masks. [57] additionally estimate the SMPL parameters for each image. We adopt the setting of training and test camera views in [57]. For each subject, 100 frames are selected for training, and we sample 1 frame every 5 frames. Detailed configurations of all sequences are described in the supplementary material.

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

**Baselines.** We compare our method with subject-specific optimization methods [56–58, 93], generalizable methods [34, 100]. All the baselines are implemented in pure PyTorch [53] for a fair comparison. Here we list only the average metric values of all selected characters on a dataset due to the size limit. We provide more detailed qualitative and quantitative comparisons in the supplementary material.

(1) **Subject-specific optimization methods.** Neural Body (NB) [58] anchors a set of latent codes to the SMPL mesh and regresses the radiance field from the posed latent codes. Animatable NeRF (AN) [56] deforms the canonical NeRF with the skeleton-driven framework and models non-rigid deformations by learning blend weight fields. [57] extend [56] with a signed distance field and pose-dependent deformation field to better model the residual deformation and geometric...
Figure 3. Qualitative results of novel view synthesis on the ZJU-MoCap and MonoCap datasets. Our method produces better rendering results while only requiring 1/100 of the training time. The bottom row lists the training time of each method.

Results on the ZJU-MoCap dataset. Table 1 compares our method with NB [58], AN [56], PixelNeRF [100], NHP [34], HN [93] and AS [57] on novel view synthesis. Our proposed representation can be optimized within around 5 minutes to produce photorealistic rendering results, while [56–58, 93] require around 10 hours to finish training and [34, 100] require 10 hours of pretraining and 1 hour of fine-tuning. [57, 93] exhibit better results than [56, 58]. However, they all require a lengthy optimization process and fail to produce reasonable renderings in only 5 minutes because their models have not converged yet. Generalizable methods [34, 100] failed to render humans with reasonable shapes under the monocular setting. Our method achieves comparable results on all of the three evaluated metrics even when only trained in minutes, which shows the effectiveness of our novel human representation. We present qualitative results of our method and baselines in Figure 3.

Results on the MonoCap dataset. Table 1 summarizes the quantitative comparison between our method and other baselines on the MonoCap dataset. Our model again achieves competitive visual quality while only requiring 1/100 of the training time due to our efficient part-based voxelized human representation and effective motion parameterization scheme. Figure 3 indicates that our method can produce better appearance details than [56–58]. Although [56, 58] have shown impressive rendering results given 4-view videos, they do not perform well on monocular inputs. [58] implicitly aggregates the temporal information using structured latent details of dynamic humans. [93] optimizes for a volumetric representation of the person in a canonical space along with the estimated human pose.

(2) Generalizable methods. PixelNeRF [100] trains a network to infer the radiance field from a single image. Neural Human Performer (NHP) [34] anchors image features to vertices of the SMPL mesh and aggregates temporal features using a transformer, which are decoded into a human model. For each evaluated subject (e.g. one subject of MonoCap), we first pre-train the network on the other dataset (e.g. ZJU-MoCap) and then finetune it on the evaluated subject until it converges.
Table 1. Quantitative comparison of our method and baseline methods on the ZJU-MoCap and MonoCap datasets. We use **bold text** for the best and *underlined text* for the second best metric value across methods. Our method achieves the fastest training speed and shows competitive rendering results. Note that the NHP [34] and PixelNeRF [100] are additionally pretrained for 10 hours. LPIPS* = LPIPS × 10³.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Time</th>
<th>ZJU-MoCap</th>
<th>MonoCap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PSNR ↑</td>
<td>SSIM ↑</td>
</tr>
<tr>
<td>Ours</td>
<td>*5 min</td>
<td><strong>31.01</strong></td>
<td>0.971</td>
</tr>
<tr>
<td>HumanNeRF [93]</td>
<td>*10 h</td>
<td>30.66</td>
<td>0.969</td>
</tr>
<tr>
<td>AS [57]</td>
<td>*10 h</td>
<td>30.38</td>
<td><strong>0.975</strong></td>
</tr>
<tr>
<td>AN [56]</td>
<td>*10 h</td>
<td>29.77</td>
<td>0.965</td>
</tr>
<tr>
<td>NB [58]</td>
<td>*10 h</td>
<td>29.03</td>
<td>0.964</td>
</tr>
<tr>
<td>NHP [34]</td>
<td>*1 h fine-tuning</td>
<td>28.25</td>
<td>0.955</td>
</tr>
<tr>
<td>PixelNeRF [100]</td>
<td>*1 h fine-tuning</td>
<td>24.71</td>
<td>0.892</td>
</tr>
</tbody>
</table>

Table 2. Quantitative results of ablation studies on the 377 sequence of ZJU-MoCap dataset. The description of each model can be found in Section 4.3. All models are trained for 5 minutes.

<table>
<thead>
<tr>
<th>Variant</th>
<th>PSNR</th>
<th>SSIM</th>
<th>LPIPS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td><strong>32.09</strong></td>
<td><strong>0.982</strong></td>
<td><strong>23.47</strong></td>
</tr>
<tr>
<td>Ours w/o Part</td>
<td>30.11</td>
<td>0.974</td>
<td>45.84</td>
</tr>
<tr>
<td>Ours w/o UV</td>
<td>31.40</td>
<td>0.979</td>
<td>30.99</td>
</tr>
<tr>
<td>Ours w/o Perc</td>
<td>30.55</td>
<td>0.976</td>
<td>44.33</td>
</tr>
</tbody>
</table>

Figure 4. Comparison of convergence LPIPS* and time needed for convergence among different variants of the proposed pipeline. The description of each model can be found in Section 4.3. The proposed components accelerate the training and enhance the rendering quality significantly. LPIPS* = LPIPS × 10³.

4.3. Ablation Studies

We perform ablation studies on the “377” sequence of the ZJU-MoCap dataset to analyze how the proposed components affect the performance and training speed of our method.

4.3.1 Ablation Studies on Proposed Components.

Table 2(a) lists the quantitative result of ablation studies on our proposed components. All models are trained for 5 minutes. “Ours w/o Part” represents the canonical human body with a single MHE-augmented NeRF [48] network, which drastically degrades the rate of convergence. To keep the comparison fair, this variant of our method has a similar number of parameters (302M) to ours (286M). However, this change leads to a significant decrease in PSNR of 1.98dB because of its unwise design of considering all parts equally complex and wasting representational power. In “Ours w/o UV”, the residual deformation network MLPres takes hash encoded (x, t) as input, which observed a PSNR degradation from 32.09dB to 31.40dB with the same training time because of severe hash collision and limited resolution. “Ours w/o Perc” does not adopt the perceptual loss during training, which in turn increases the LPIPS distance. This comparison illustrates the importance of perceptual loss for visual fidelity. Figure 4 and Figure 5 provide a more intuitive qualitative comparison among variants of the proposed method.

Analysis of the part-based voxelized human representation. MHE [48] defines a multiresolution hash table of trainable features to embed input coordinates to a high-dimension space. We find that simply increasing the size...
of the hash table does not always result in better performance with the same training time, because a bigger hash table leads to higher memory consumption and increases the time of each training iteration. The proposed part-based voxelized human representation allows us to adapt the hash table size according to the complexity of the human part, allowing us to efficiently represent the human body. Table 2(a) demonstrates the effectiveness of the part-based voxelized human representation. To further validate this representation, we design two variants that use the hash tables of size $2^{15}$ and $2^{20}$ in all human parts respectively. Table 2(c) summarizes the ablation studies, indicating that varying the model parameters in human parts improves the performance.

Analysis of the motion parameterization scheme. Table 2(a) shows that our model works better when the residual deformation network MLP$_{res}$ takes parameterized 3D surface-time $(u, v, t)$ coordinates as input, compared with taking 4D space-time $(x, t)$ as input. Note that $(u, v, t)$ makes MHE much more memory efficient than $(x, t)$. To further validate the effectiveness of our motion parameterization, we additionally design three variants of the MLP$_{res}$ input. (1) PE: positional encoded $(x, t)$. (2) XYZ-Code: hash encoded $x$ and a per-frame learnable latent code [57]. (3) XYZ-Pose: hash encoded $x$ and the pose parameter [40, 93]. When taking the positional encoded $(x, t)$ as input, we use a larger network for MLP$_{res}$. The results in Table 2(b) indicate that hash encoded $(u, v, t)$ achieves the best performance.

Analysis of Robustness. To evaluate the robustness of the proposed system, we measure the time needed to achieve an evaluation PSNR of 30 for five times on the “377” sequence. This results in a training time with a mean value of 76.00s and a standard derivation of 13.56s, showing the stability of the proposed method.

5. Limitations

Although our method can quickly reconstruct high-quality human models from videos, it has some limitations to be further addressed. First, our method currently relies on accurate SMPL parameters, which could be difficult to obtain under in-the-wild settings. It is interesting to utilize the techniques in [79, 93] to optimize the human pose parameters along with the training of human avatars. Second, we can only reconstruct foreground dynamic humans, while dynamic scenes typically include foreground and background entities. It might be plausible to combine our method with ST-NeRF [29] or MultiNB [74] to quickly reconstruct dynamic scenes containing foreground and background objects.

6. Conclusion

We introduce a novel dynamic human representation that can be quickly optimized from videos and used for generating free-viewpoint videos of the human performer. This representation consists of a part-based voxelized human model in the canonical space and a motion parameterization scheme that transforms points from the world space to the canonical space. The part-based voxelized human model decomposes the human body into multiple parts and represents each part with an MHE-augmented NeRF network, which efficiently distributes network representational power and significantly improves the training speed. When predicting the motion of a query point, the motion field reparameterizes the point coordinate as 2D surface-level UV coordinate, which effectively reduces the dimensionality of motion the network is required to model, resulting in a boost in convergence rate. Experiments demonstrate that our proposed representation can be optimized at 1/100 of the time of previous methods while still maintaining competitive rendering quality. We show that given a 100-frame monocular video of a $512 \times 512$ resolution, our method can produce photorealistic free-viewpoint videos in minutes on an RTX 3090 GPU.

Acknowledgements. This work was supported by the Key Research Project of Zhejiang Lab (No. K2022PG1BB01), NSFC (No. 62172364), and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.
References


In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14314–14323, 2021. 2, 4, 5, 6, 7


[104] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In ICCV, 2021. 2