Supplementary Material: Animatable Neural Implicit Surfaces for Creating Avatars from Videos

Sida Peng[1] Shangzhan Zhang[1] Zhen Xu[1] Chen Geng[1]
Boyi Jiang[2] Hujun Bao[1] Xiaowei Zhou[1]
[1]Zhejiang University [2]Image Derivative Inc

In the supplementary material, we describe how to animate the learned human model. For reproducibility, we provide implementation details, dataset details, and evaluation details. To show the effectiveness of our approach, we present more results of 3D reconstruction and image synthesis. In addition, we provide a video to describe our approach and present the qualitative results.

1. Animation

3D reconstruction. After training, AniSDF can be used to generate 3D human shapes under given human poses. For training human poses, we first construct a set of grid points by discretizing the 3D human bounding box in the observation space with a voxel size of $5mm \times 5mm \times 5mm$. Then, the grid points are transformed to the canonical space using the inverse LBS model and the displacement field, which are fed into the geometry model F_s to compute signed distances. We extract the human mesh from the signed distances with the Marching Cubes algorithm [10].

For novel human poses, we adopt another way to generate 3D human shapes, as the displacement field cannot generalize to unseen human poses. We first discretize the human bounding box in the canonical space with a voxel size of $5mm \times 5mm \times 5mm$ and evaluate the signed distances for the grid points. Then, the canonical human mesh is extracted from the signed distances based on the Marching Cubes algorithm. Blend weights of mesh vertices are obtained by retrieving blend weights of the closest surface points on the SMPL mesh under the canonical space. Given a human pose, we use the forward LBS model to deform the canonical mesh to the observation space.

Image synthesis. For training human poses, we can use "Ours-V", "Ours-S" and "Ours-S*" to render images. The color and feature fields infer the colors and feature vectors based on appearance codes of corresponding video frames. For novel human poses, we can use "Ours-S" and "Ours-S*" to render images. The color and feature fields take the appearance code at the first frame as input.

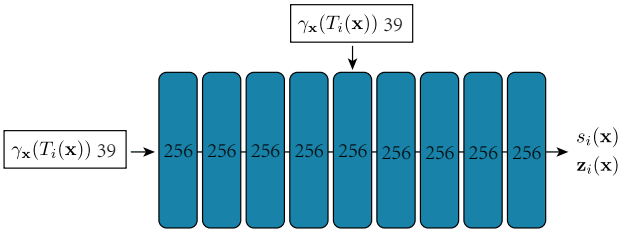

Figure 1. **Signed distance field.** All layers are linear layers with softplus activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$ as input and output the signed distance $s_i(\mathbf{x})$ and geometry feature $\mathbf{z}_i(\mathbf{x})$. The dimension of the input is shown in each block.

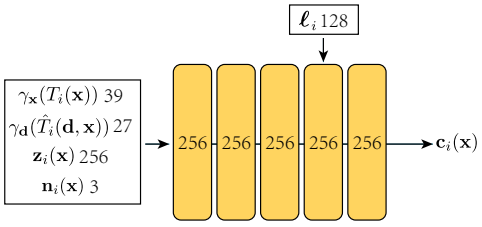

Figure 2. **Color field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$, the positional encoding of view direction $\gamma_{\mathbf{d}}(\hat{T}_i(\mathbf{d},\mathbf{x}))$, normal $\mathbf{n}_i(\mathbf{x})$, and geometry feature $\mathbf{z}_i(\mathbf{x})$ as inputs. We introduce the appearance code $\boldsymbol{\ell}_i$ in the fourth layer. The dimension of the input is shown in each block.

2. Implementation details

Figures 1, 2, 3, 4 and 5 illustrate network architectures of signed distance field F_s, color field $F_{\mathbf{c}}$, displacement field $F_{\Delta\mathbf{x}}$, feature field $F_{\mathbf{f}}$, and 2D neural renderer, respectively. We perform positional encoding [11] to the spatial point and viewing direction. 6 frequencies are used when encoding spatial position, and 4 frequencies are used when encoding

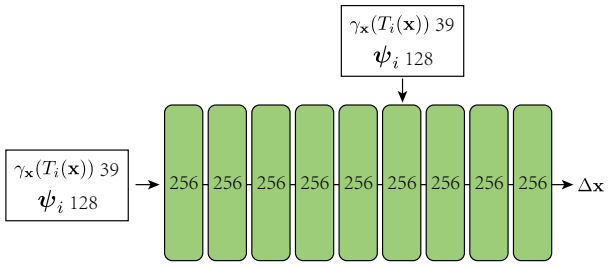

Figure 3. **Displacement field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$ and the per-frame latent code for ψ_i as input.

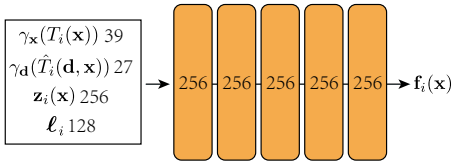

Figure 4. **Feature field.** All layers are linear layers with ReLU activations except for the final layer. The network takes the positional encoding of spatial point $\gamma_{\mathbf{x}}(T_i(\mathbf{x}))$, the positional encoding of view direction $\gamma_{\mathbf{d}}(\hat{T}_i(\mathbf{d},\mathbf{x}))$, and geometry feature $\mathbf{z}_i(\mathbf{x})$ as inputs, and output the feature vector $\mathbf{f}_i(\mathbf{x})$.

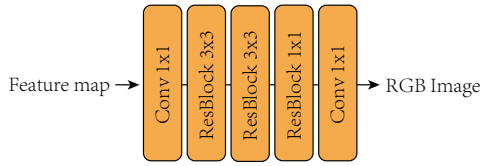

Figure 5. **2D neural renderer.** The network consists of standard ConvBlocks and ResBlocks. ALL blocks adopt the ReLU activations. The kernel size is shown in each block. The dimension of feature in the convolution operator is 256.

viewing direction. The dimensions of appearance code $\boldsymbol{\ell}_i$ and displacement field code $\boldsymbol{\psi}_i$ are 128.

Training. We take a two-stage training pipeline. First, the parameters of F_s, $F_{\mathbf{c}}$, $F_{\Delta\mathbf{x}}$, $\{\boldsymbol{\ell}_i\}$ and $\{\boldsymbol{\psi}_i\}$ are jointly optimized over the input video. Second, we fix the parameters of F_s and $\{\boldsymbol{\ell}_i\}$, and train the feature field $F_{\mathbf{f}}$ and 2D neural renderer. We use the Adam optimizer [9] for the training and set the learning rate as $5e^{-4}$, which decays exponentially to $5e^{-5}$ during the optimization. The training is conducted on one 2080 Ti GPU. We sample 1024 rays at each iteration. For a monocular video of 300 frames, both stages take around 100k iterations to converge.

3. Dataset details

Human3.6M [8] Following [12], we use three camera views for training and test on the remaining view. [12] select video clips from the action "Posing" of S1, S5, S6, S7, S8, S9, and S11. The number of training frames and test frames is described in Table 1.

subject	S1	S5	S6	S7	S8	S9	S11
training	150	250	150	300	250	260	200
test	49	127	83	200	87	133	82

Table 1. **The number of training frames and test frames of the Human3.6M dataset.**

subject	S1	S2	S3	S4	S5	S6	S7
training	69	300	70	100	100	100	70

Table 2. **The number of video frames for each subject in the SyntheticHuman dataset.**

SyntheticHuman It contains 7 animated human characters obtained from RenderPeople [3] and Mixamo [2]. We render these subjects using Blender [1, 5]. Subjects S1, S2, S3, and S4 perform rotation with A-pose, which are rendered into monocular videos. Subjects S5, S6 and S7 perform random actions, which are rendered into 4-view videos. The number of video frames is listed in Table 2.

MonoCap It consists of two videos "Lan" and "Marc" from DeepCap dataset [7], and two videos "Olek" and "Vlad" from DynaCap dataset [6]. "Lan" is selected from 620-th frame to 1220-th frame in the original video. "Marc" is selected from 35000-th frame to 35600-th frame. "Olek" is selected from 12300-th frame to 12900-th frame. "Vlad" is selected from 15275-th frame to 15875-th frame. Each clip has 300 frames for training and 300 frames for evaluating novel pose synthesis, respectively. We use the 0-th camera as the training view for "Lan" and "Marc". The 44-th camera is selected as the training view for "Olek". The training view of "Vlad" is the 66-th camera. We uniformly select ten cameras from the remaining cameras for test.

4. Evaluation details

We follow [13] to calculate the metrics of image synthesis. Specifically, the 3D human bounding box is first projected to produce a 2D mask. Then, we calculate the PSNR metric based on the pixels inside the 2D mask. Since the SSIM and LPIPS metrics require the image input, we compute the 2D box that bounds the 2D mask and crop the image within the box, which is used to calculate the SSIM and LPIPS metrics. For the SyntheticHuman dataset, we calculate the reconstruction metrics every 10-th frame. For the Human3.6M and MonoCap datasets, we calculate the metrics of image synthesis every 30-th frame.

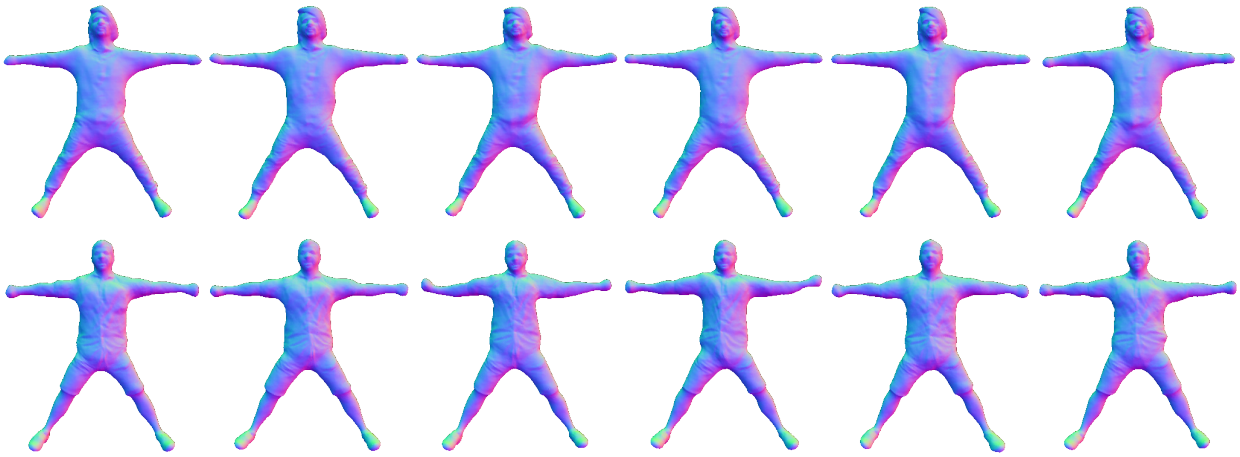

Figure 6. **Canonical geometries deformed by the displacement field on the MonoCap dataset.** We deform the canonical geometries using the learned displacement fields in different video frames. For different frames, the displacement fields produce different non-rigid deformations, including the cloth deformations and body deformations. We provide more results in the supplementary video.

Figure 7. **3D reconstruction results in the Human3.6M, MonoCap, and People-Snapshot datasets.** We reconstruct humans of Human3.6M [8] from 3-view videos. The geometries in MonoCap [6, 7] and People-Snapshot [4] are reconstructed from monocular videos.

	P2S↓				CD↓			
	NB [13]	AN [12]	Ours + Neus	Ours	NB [13]	AN [12]	Ours + Neus	Ours
S1	1.44	2.73	0.91	**0.64**	1.39	2.02	1.09	**0.81**
S2	1.68	3.12	0.87	**0.69**	1.48	2.11	0.93	**0.74**
S3	1.52	2.41	0.80	**0.58**	1.42	1.76	0.96	**0.74**
S4	1.20	3.29	0.84	**0.58**	1.23	2.28	0.99	**0.71**
S5	1.20	2.01	0.65	**0.45**	1.14	1.60	0.68	**0.49**
S6	1.31	2.44	0.89	**0.58**	1.28	1.83	0.90	**0.60**
S7	1.61	2.36	1.42	**1.21**	1.74	2.20	1.71	**1.47**
average	1.42	2.62	0.91	**0.67**	1.38	1.97	1.04	**0.79**

Table 3. **Results of 3D reconstruction on SyntheticHuman dataset.** "Ours + Neus" means that we render AniSDF with the volume rendering scheme in Neus [14].

5. Results of 3D reconstruction

Table 3 lists the per-subject comparison on the SyntheticHuman dataset, which shows that AniSDF with the volume rendering technique in VolSDF [16] achieves the best performance of 3D reconstruction in terms of the P2S and CD metrics.

Figure 6 visualizes the non-rigid deformations captured by the displacement fields. To obtain the geometry deformed by the displacement field, we first construct a set of grid points by discretizing the human bounding box in the canonical space and transform the grid points using the displacement field. Then, we evaluate the signed distances for the transformed points and extract the human mesh with

the Marching Cubes algorithm.

Figure 7 presents the qualitative results of our reconstructed geometries. We additionally reconstruct humans in the People-Snapshot dataset [4], which captures performers rotating while holing the A-pose. The results demonstrate that our method can reconstruct high-quality geometries from monocular videos.

6. Results of image synthesis

Figure 8 presents the rendering results of subjects "Olek" and "Vlad" in the MonoCap dataset driven by complex human poses, which show that our method can synthesize photorealistic images under complex human poses. We provide more results in the supplementary video.

Figures 9 and 10 show the qualitative comparisons between our method and [12, 13, 15] on novel view synthesis of training human poses and novel human poses, respectively. Our method recovers detailed human appearance and produces more photorealistic images than other methods.

References

[1] Blender. https://www.blender.org/. 2

[2] Mixamo. https://www.mixamo.com/. 2

[3] Renderpeople. https://renderpeople.com/. 2

[4] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *CVPR*, 2018. 3, 4

[5] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 2

[6] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. In *SIGGRAPH Asia*, 2021. 2, 3

[7] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *CVPR*, 2020. 2, 3

[8] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2013. 2, 3

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

[10] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH*, 1987. 1

[11] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[12] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*, 2021. 2, 3, 4

[13] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021. 2, 3, 4

[14] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 3

[15] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. In *CVPR*, 2020. 4

[16] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv preprint arXiv:2106.12052*, 2021. 3

Figure 8. **Results of image synthesis under complex human poses in the MonoCap dataset.** Our model is trained on short video clips and can generalize to complex human poses. More results can be found in the supplementary video.

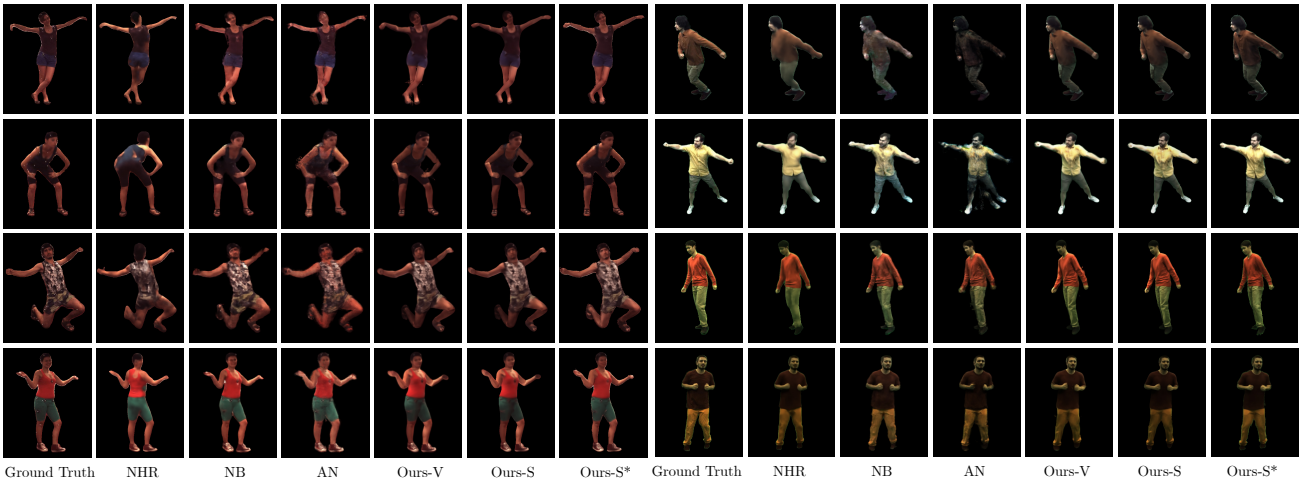

Figure 9. **Novel view synthesis of training human poses in the Human3.6M and MonoCap datasets.** Our method has better performance on image synthesis. "Ours-S*" renders more appearance details. Zoom in for details.

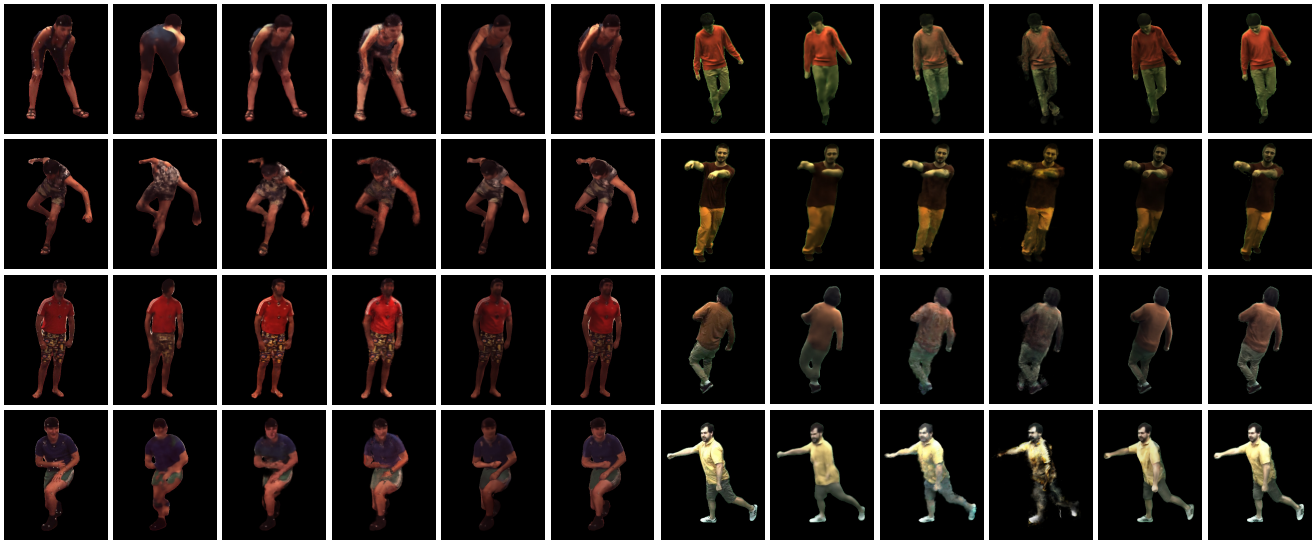

Figure 10. **Novel view synthesis of novel human poses in the Human3.6M and MonoCap datasets.** The rendered images of our method have a higher visual quality. Zoom in for details.