FlexNeRF: Photorealistic Free-Viewpoint Rendering of Moving Humans from Sparse Views (Supplementary Materials)

Vinoj Jayasundara[1]*, Amit Agrawal[2], Nicolas Heron[2], Abhinav Shrivastava[1], Larry S. Davis[1,2]

[1]University of Maryland, College Park [2]Amazon.com, Inc.

{vinoj, lsdavis}@umd.edu, {aaagrawa, heron}@amazon.com, abhinav@cs.umd.edu

Contents

1. Introduction

We present supplementary materials for our CVPR submission below. As discussed in the paper, our approach signficantly outperforms the state-of-the-art in the *Sparse* view settings. For datasets used in the paper, the average number of views across videos in *sparse* and *full* settings are as shown in Table 1.

Dataset	# of Videos	# of Views	
		Full	*Sparse*
ZJU-MoCap [4]	6	556	43
People Snapshot [2]	7	517	39
SCF	7	-	40

Table 1. The average number of views across videos in *sparse* and *full* settings for various datasets.

By default, our internal SCF dataset has *sparse* views. The videos are shorter, $8-9$ seconds on average, and do not have multiple full rotations of the human subject. After removing stationary and redundant frames, the number of input frames average to ≈ 40. Thus, results on SCF dataset are in *sparse* setting by default.

*Part of the work was done while the author was an intern at Amazon.

2. Free-viewpoint Rendering

Given a monocular video, the task of free-viewpoint rendering involves pausing at any arbitrary frame and rendering the corresponding 360^0 video path from the viewpoint. This enables to freeze the subject while viewing it from all directions. We illustrate the free-viewpoint capabilities of our method in examples below.

Figure 1 (top row) shows 5 frames of a monocular video from SCF dataset. The video features a human subject displaying rapid, complex motions. We learn the FlexNeRF model using *sparse* views (≈ 40) from the video. The bottom row shows rendering of the paused middle frame (depicted in red bounding box) by moving the camera in a 360^0 path around the subject. Notice the missing hand regions and blurred hands in HumanNeRF outputs. Our approach results in higher rendering quality.

Similarly, Figure 2 shows the free-view point rendering for ZJU-Mocap dataset, comparing HumanNeRF [6] with our approach using *sparse* views. Compared to our approach, HumanNeRF renderings show blurred clothing as well as ghosting artifacts close to arms. Please zoom in for details. Figure 3 shows the corresponding results for the dense view setting.

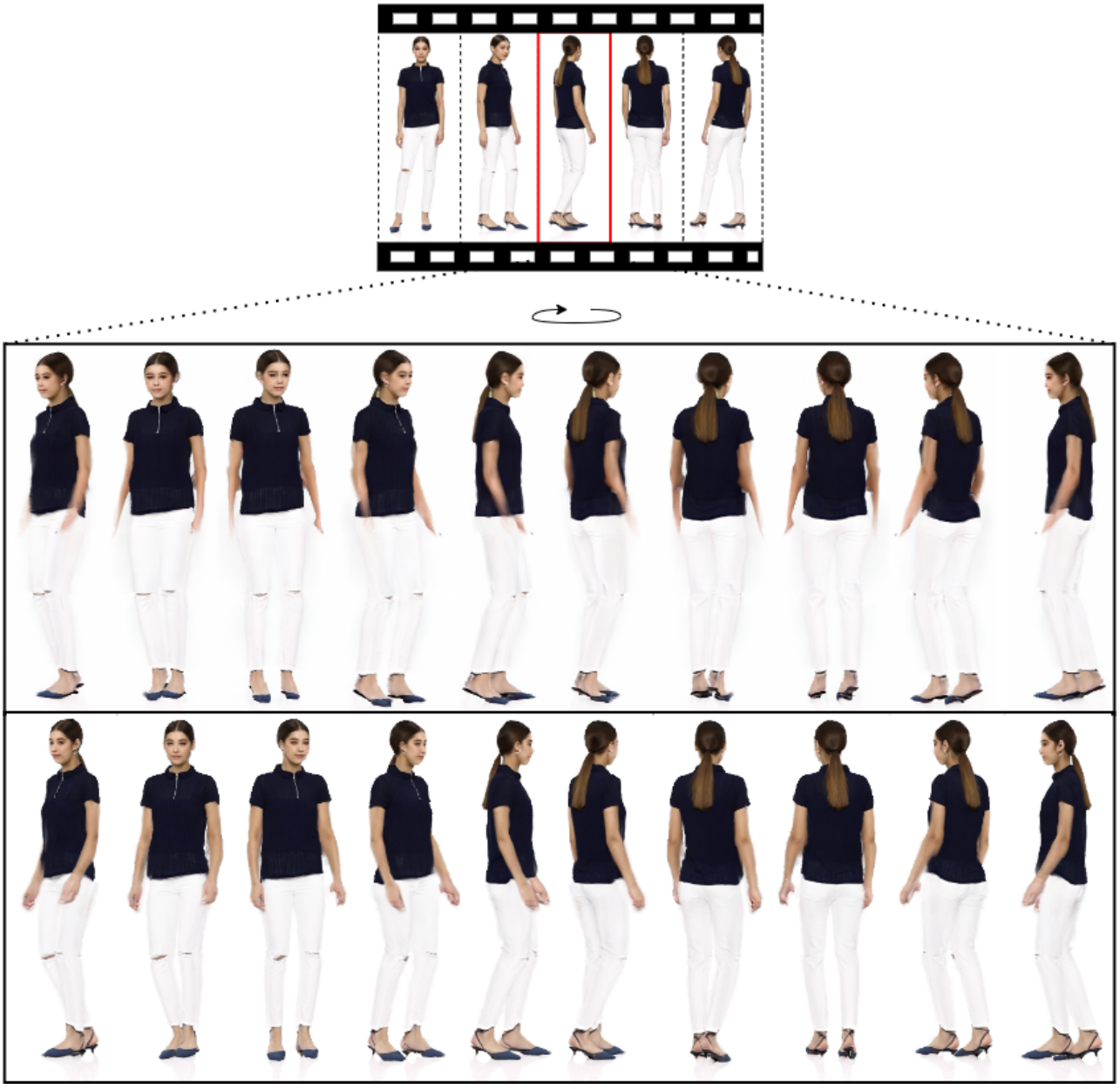

Figure 1. Example of free-viewpoint rendering on SCF dataset. (Top Row): 5 frames from an input video. (Middle Row): The middle frame (highlighted in red box) is rendered using HumanNeRF method by freezing the frame and rotating the camera around the subject. (Bottom Row): Similar to the middle row, but rendered with our approach. Notice the missing/blurred hands in HumanNerF outputs. Our approach provides higher quality renderings.

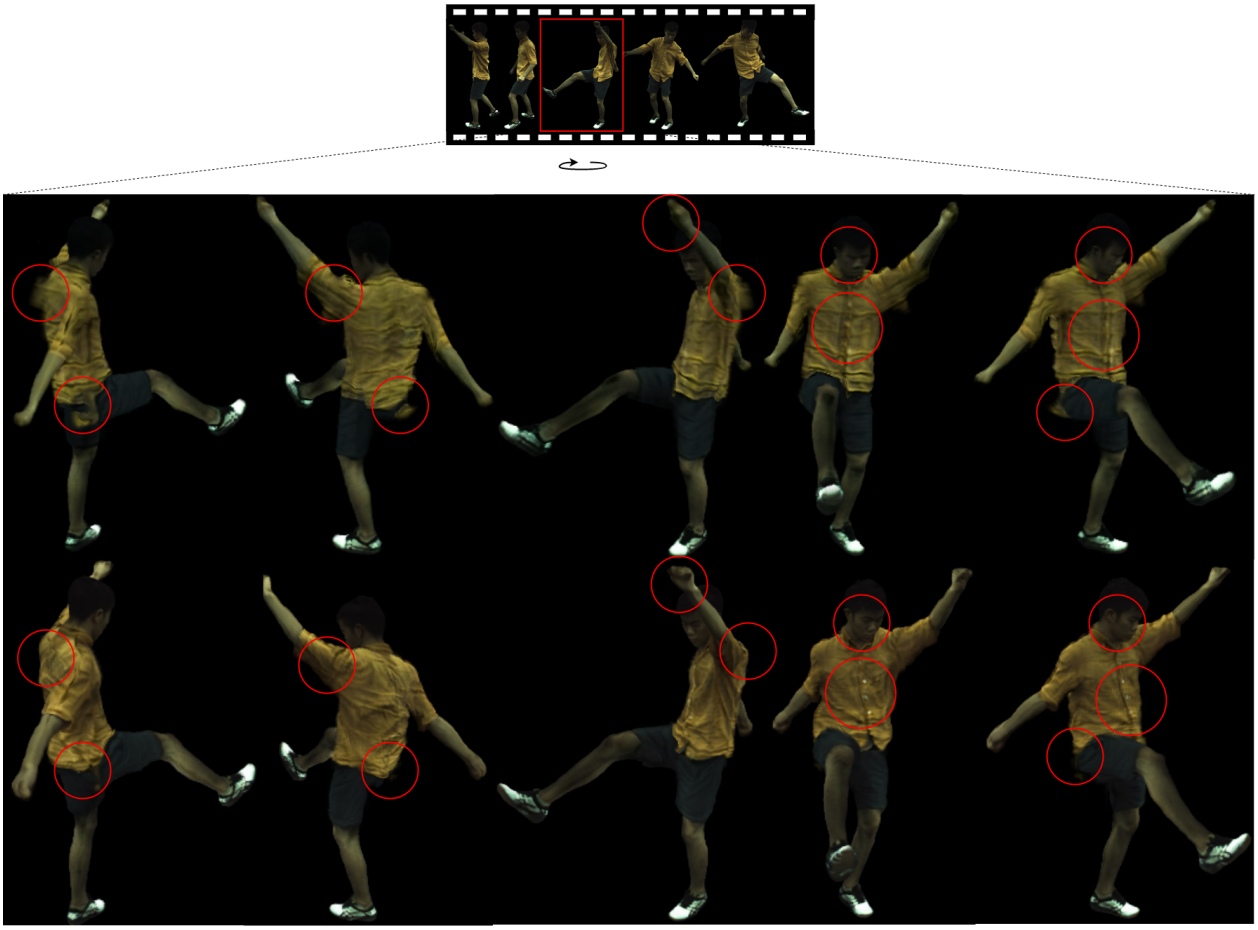

Figure 2. Example of free-viewpoint rendering on the ZJU-MoCap dataset using **sparse views**. (Top Row): 5 frames from the video. (Second Row): Middle frame is rendered with HumanNeRF by freezing the frame and rotating the camera around the subject. (Third Row): Similar to the second row, but rendered with our approach.

Figure 3. Example of free-viewpoint rendering on the ZJU-MoCap dataset using **full views**. (Top Row): 5 frames from the video. (Second Row): Middle frame is rendered with HumanNeRF by freezing the frame and rotating the camera around the subject. (Third Row): Similar to the second row, but rendered with our approach.

3. Architecture Details

Now we present details on the architecture of various networks used in our framework.

3.1. Motion Weight Volume Network

The rigid motion field x_R can be calculated with the aid of the motion weight volume $W^c(x) = CNN_{\theta_R}(x; z)$, where z is a constant latent random vector initialized with a Gaussian distribution. In this CNN illustrated in Figure 4, the output of the first fully-connected layer with 1024 units is reshaped in to 4 dimensions, followed by 5 transposed convolutions layers with LeakyReLU activation [8], producing the motion weight volume $W^c(x) \in \mathbb{R}^4$. The final layer of the CNN has $(K + 1)$ channels, where K is the total number of 3D joint locations, which is 24 in our case. As motion priors, we add an approximate Gaussian bone volume [5] computed for the chosen canonical pose to the output of the CNN.

3.2. Temporal Deformation Network

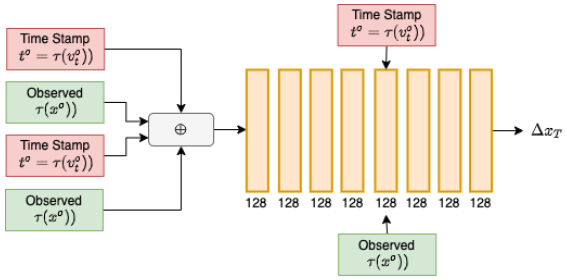

Figure 5. Temporal deformation network $MLP_{\theta_{TD}}$

The temporal deformation MLP, $MLP_{\theta_{TD}}$ consists of 8 fully connected layers as illustrated in Figure 5. Except for the last two layers, the other layers have ReLU activation [1]. It takes in the positional encoded temporal vectors $\tau(v_t^o))$ and $\tau(v_t^c))$, and the positional encoded point posi-

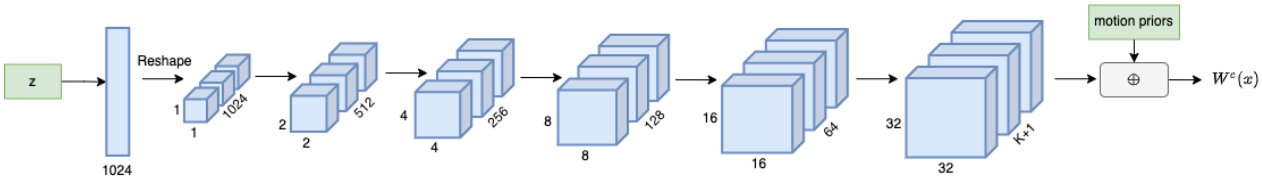

Figure 4. Motion Weight Volume Network $CNN_{\theta_R}(x; z)$.

tion vectors $\tau(x^o)$ and $\tau(x^c)$ as inputs. The observed encodings $\tau(v_t^o))$ and $\tau(x^o)$ are skip connected to the fifth layer to generate the deformation Δx_T.

3.3. Scene Representation Network

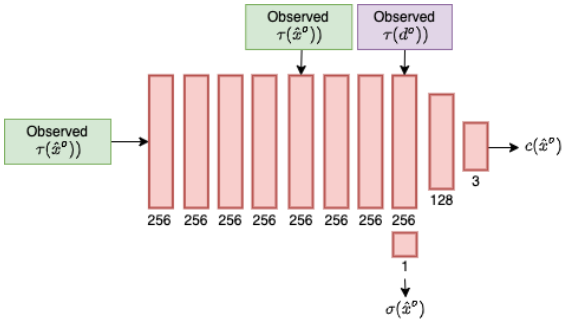

Figure 6. Scene representation network $MLP_{\theta_o}(\gamma(\hat{x}^o))$

The scene representation MLP $MLP_{\theta_o}(\gamma(\hat{x}^o))$ consists of 8 fully connected layers with 256 units and ReLU activation [1], followed by another fully connected layer with 128 and ReLU activation (Figure 6). The positional embedded observed $\hat{x}^o$ and observed directions $\hat{d}^o$ are skip connected to the fifth and eighth layers respectively. The outputs color $c(\hat{x}^o)$ and density $\sigma(\hat{x}^o)$ are obtained via fully connected layers with 3 and 1 units respectively.

4. Optimization Details

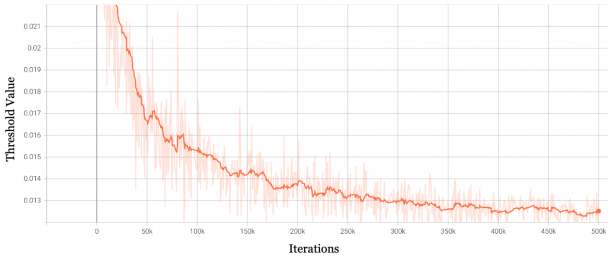

Figure 7. Learnable threshold b converges to 0 as training progresses.

Learning the threshold b: As discussed in the paper, the thresholding approach proposed for the rendering binary segmentation map employs a learnable threshold b, to ease training. For a well-learned model, this threshold should ideally be 0. Figure 7 shows that the learnable threshold b indeed converges to 0 as training progress.

Optimizer: We optimize the FlexNeRF model using the Adam optimizer [3] with learning rates set to 5×10^{-4} for the Scene representation network, canonical to observed transformation, and binary segmentation MLPs, and 5×10^{-5} for the rest of the networks. We use a weight decay of 0.01 as a form of regularization for parameters such as the learnable threshold b and the temporal vectors.

The FlexNeRF model is trained on a setup with 4 V100 GPUs, takes in 6 patches of 32×32 each, and produces rendered images of size 512×512. The corresponding rays are sampled 128 times each. The model convergence requires $500K$ iterations taking up to approximately $78 - 80$ hours.

5. Additional Results

Below we showcase additional quantitative results on public datasets.

5.1. Quantitative Results (ZJU-MoCap Dataset)

We present the performance breakdown of FlexNeRF on individual videos of the ZJU-MoCap dataset [4] for the *full* view setting in Table 2. We select the following subject IDs: (377, 386, 387, 392, 393, 394) compatible with [6] for comparison purposes. We observe that our approach outperforms the state-of-the-art across all videos and metrics, except for the PSNR metric on the subject 377. The performance gain of our approach is the lowest for the subject 377 in comparison to HumanNeRF, since it has the least amount of non-rigid artifacts out of all the subjects.

Method	LPIPS $\times 10^3 \downarrow$	PSNR ↑	SSIM ↑
Neural Body [4]	57.67*	24.62	0.8490
H-NeRF [7]	57.31*	26.33	0.8680
HumanNeRF [6]	36.79	28.05	0.8984
Ours	**35.63**	**28.77**	**0.9043**

Table 3. Quantitative performance comparison on the People Snapshot [2] dataset. $*$ refers to adjusted LPIPS from the values reported in [7] to fit the same scale as our experiments.

Method	Subject 377			Subject 392		
	LPIPS $\times 10^3 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\times 10^3 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
Neural Body [4]	40.95	29.11	0.9674	53.27	30.10	0.9642
HumanNeRF [6]	24.06	**30.41**	0.9743	32.12	31.04	0.9705
Ours	**23.58**	30.39	**0.9761**	**30.96**	**32.17**	**0.9769**

Method	Subject 386			Subject 393		
	LPIPS $\times 10^3 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\times 10^3 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
Neural Body [4]	46.43	30.54	0.9678	59.05	28.61	0.959
HumanNeRF [6]	28.99	33.20	0.9752	36.72	28.31	0.9603
Ours	**27.29**	**34.59**	**0.9803**	**33.85**	**29.87**	**0.9711**

Method	Subject 387			Subject 394		
	LPIPS $\times 10^3 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\times 10^3 \downarrow$	PSNR $\uparrow$	SSIM $\uparrow$
Neural Body [4]	59.47	27.00	0.9518	54.55	29.1	0.9593
HumanNeRF [6]	35.58	28.18	0.9632	32.89	30.31	0.9642
Ours	**32.49**	**29.57**	**0.9786**	**30.76**	**31.97**	**0.9709**

Table 2. Comparison of various performance metrics on the ZJU-MoCap [4] dataset.

5.2. Quantitative Results (People Snapshot Dataset)

We present additional performance comparison with H-NeRF [7] on the People Snapshot [2] dataset under the *full* setting in Table 3. It is evident that our approach surpass the state-of-the-art across all metrics.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *ArXiv*, abs/1803.08375, 2018. 4, 5

[2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8387–8397, Jun 2018. CVPR Spotlight Paper. 1, 5, 6

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[4] 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. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9050–9059, 2021. 1, 5, 6

[5] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Vid2actor: Free-viewpoint animatable person synthesis from video in the wild. *ArXiv*, abs/2012.12884, 2020. 4

[6] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16189–16199, 2022. 1, 5, 6

[7] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion. In *NeurIPS*, 2021. 5, 6

[8] Jin Xu, Zishan Li, Bowen Du, Miaomiao Zhang, and Jing Liu. Reluplex made more practical: Leaky relu. *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7, 2020. 4