Dance In the Wild: Monocular Human Animation with Neural Dynamic Appearance Synthesis (Supplementary material)

Tuanfeng Y. Wang Duygu Ceylan Krishna Kumar Singh
Adobe Research
{yangtwan,ceylan,krishsin}@adobe.com

Niloy J. Mitra
Adobe Research, University College London
nimitra@adobe.com

1. Network architecture

Our network consists of five sub-modules: 1) pose encoder $\mathbf{E_P}$; 2) motion encoder $\mathbf{E_M}$; 3) pose refinement network $\mathbf{E_{Refine}}$; 4) style generator $\mathbf{T}$; and 5) discriminator $\mathbf{D}$. The basic building block of each of the modules is a convolutional layer $\mathbf{Conv}_{\mathsf{out,k,s}}(x)$, where x is the input tensor, out is the number of output channels, k is the kernel size, s is the stride. The convolutional layer is followed by a LeakyRelu activation by default. We further define a residual block:

$$\mathbf{Res}_{\mathsf{out}}(x) = \frac{1}{\sqrt{2}}(\mathbf{Conv}_{\mathsf{out},3,2}(\mathbf{Conv}_{\mathsf{in},3,1}(x)) + \mathbf{Conv}_{\mathsf{out},1,2}(x)),$$

where in is the number of input channels from x. For $\mathbf{T}$ and $\mathbf{D}$, we adopt the exact architecture from StyleGAN2. Please refer to [4] for more details.

Pose encoder $\mathbf{E_P}$. Starting from a pose signature $\mathbb{P}_i(6 \times 512 \times 512)$, we apply $\mathbf{Conv}_{32,3,1}(\cdot) \to \mathbf{Res}_{64}(\cdot) \to \mathbf{Res}_{128}(\cdot) \to \mathbf{Res}_{256}(\cdot) \to \mathbf{Res}_{512}(\cdot)$ to generate our pose feature $\mathcal{P}_i(512 \times 32 \times 32)$.

Motion encoder $\mathbf{E_M}$. Similar to $\mathbf{E_P}$, starting from a motion signature $\mathbb{M}_i(60 \times 512 \times 512)$, we apply $\mathbf{Conv}_{64,3,1}(\cdot) \to \mathbf{Res}_{256}(\cdot) \to \mathbf{Res}_{512}(\cdot) \to \mathbf{Res}_{1024}(\cdot) \to \mathbf{Res}_{2048}(\cdot)$ to generate our motion feature $\mathcal{M}_i(2048 \times 32 \times 32)$.

Pose refinement network $\mathbf{E_{Refine}}$. We first concatenate $\mathcal{P}_i$ and $\mathcal{M}_i$ to get $\mathcal{P}_i^{int}(2560 \times 32 \times 32)$. We then apply $\mathbf{Conv}_{1024,3,1}(\cdot) \to \mathbf{Conv}_{512,3,1}(\cdot) \to \mathbf{Conv}_{512,3,1}(\cdot)$ to produce $\widetilde{\mathcal{P}_i}(512 \times 32 \times 32)$.

2. Comparison with Savitzky-Golay filter [5]

Inspired by the previous efforts on pose stablization in motion capture literature (e.g., [3]), we apply the Savitzky–Golay filter [5] on the learned pose feature sequence $\{\mathcal{P}_i\}$ where $i \in [-2, 2]$ to smooth the pose input in the feature domain, and directly use the smoothed pose feature as the input to our generator. We find that such filtering tends to produce an over-smoothed pose feature, causing the generator to produce ghost effects around thin structures, i.e., arms and hands, as show in Fig. 1.

Figure 1. Applying the Savitzky-Golay filter [5] to refine the pose features tends to over smooth the features, making it difficult for the generator the synthesize thin structures, e.g., hands.

	MSE ↓	SSIM ↑	LPIPS ↓	FID ↓
w/ Savitzky–Golay filter	0.0377	0.9709	0.1014	50.5325
Ours	**0.0347**	**0.9728**	**0.0913**	**48.8647**

Table 1. We conduct quantitative evaluation around the hand region of each frame.

In Table 1, we perform quantitative evaluation on the thin structure synthesised by our results and the baseline with Savitzky-Golay filter [5]. For each frame, we crop a square region around the right hand detected by OpenPose [1] (examples shown in Fig. 2) and apply different metrics over the generated sequence in this crop region. The numbers indicate that our refinement network synthesizes the thin structure better than performing averaging over the feature

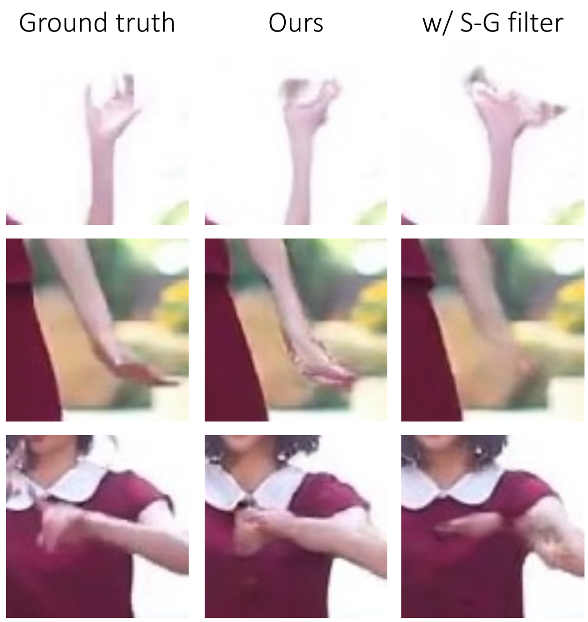

Figure 2. Examples of cropped hand regions used in the evaluations for Table 1.

space.

3. Shape prior from dense body UV prediction.

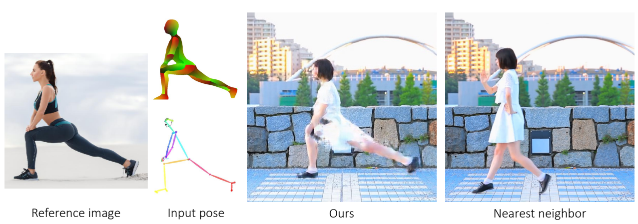

Figure 3. When the target is an extreme pose that is not covered by the train set, the difference between the shape of dense body UV prediction and the shape of the dressed body may affect the performance of the proposed method.

Since our method starts from the dense body UV map generated by Densepose [2] detector, the difference between the shape of the bare body and the shape of the dressed body may affect the performance of the proposed method, especially when the target pose is not covered in the train set. In this experiment, we show an example with the extreme pose that is very different from the dancing sequence used to train our network in Figure 3. We observe that our approach can still synthesis a reasonable appearance of the character for the target extreme pose, but such shape prior of the dense body UV prevents us from generating more plausible details. This indicates the exploration of clothes-aware dense UV (i.e., [6]) can be a promising direction for future work.

References

[1] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[2] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. 2018. 2

[3] Daniel Holden. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 1

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, volume abs/1912.04958, 2019. 1

[5] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 1

[6] Feitong Tan, Danhang Tang, Dou Mingsong, Guo Kaiwen, Rohit Pandey, Cem Keskin, Ruofei Du, Deqing Sun, Sofien Bouaziz, Sean Fanello, Ping Tan, and Yinda Zhang. Humangps: Geodesic preserving feature for dense human correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2