TransMoMo: Invariance-Driven Unsupervised Video Motion Retargeting

Supplementary Material

Zhuoqian Yang[1*] Wentao Zhu[2*] Wenyan (Wayne) Wu[3*]
Chen Qian[4] Qiang Zhou[3] Bolei Zhou[5] Chen Change Loy[6]
[1]Robotics Institute, Carnegie Mellon University [2]Peking University [3]BNRist, Tsinghua University
[4]SenseTime Research [5]CUHK [6]Nanyang Technological University
zhuoqiay@cs.cmu.edu wtzhu@pku.edu.cn
wwy15@mails.tsinghua.edu.cn qianchen@sensetime.com
zhouqiang@tsinghua.edu.cn bzhou@ie.cuhk.edu.hk ccloy@ntu.edu.sg

Abstract

This document provides supplementary information which is not elaborated in our manuscript due to space limits. Section 1 gives details about the implementation of the three stages of our method. Section 2 describes the datasets and evaluation metrics we use in our experiments. Section 3 provides some additional results of ablation study. We also present a video demo which includes more results on the project page [1].

1. Implementation details

1.1. Skeleton Extraction

We use a pretrained DensePose model [1] for skeleton extraction, missing keypoints are complemented by nearest-neighbor interpolation. The extracted skeleton sequences are smoothed using a gaussian kernel with a temporal standard deviation $\sigma = 2$. We use $N = 15$ joints for a skeleton, detailed skeleton format will be given in our Github repository.

1.2. Motion Retargeting Network

The sizes of the latent representations are $C_m = 128$, $C_s = 256$ and $C_v = 8$. Our encoders down-sample the input sequences to an eighth of its original length, therefore $M = \frac{T}{8}$. For limb-scaling, we use global and local scaling factors randomly sampled from $[0.5, 2]$ (uniformly distributed). For view perturbations we use $K = 3$. Our motion retargeting network is trained $200,000$ steps with batch size 64 and learning rate $\alpha = 0.0002$ using Adam [6] optimization algorithm. The weights of the loss terms are given as follows: $\lambda_{\text{rec}} = 10, \lambda_{\text{crs}} = 4, \lambda_{\text{adv}} = 2, \lambda_{\text{trip}} = 10, \lambda_{\text{inv}} = 2$. These parameters are determined through quantitative and qualitative experiments on a validation set.

1.3. Skeleton-to-Video Rendering

For skeleton-to-video rendering, we recorded target videos of 5 subjects as training data (none of the recorded subjects is an author of this work). We use the synthesis pipeline proposed in [4]. Each generator is trained on the target video for 40 epochs and the output size is 512×512.

2. Experimental Details

2.1. Dataset

In-the-wild dataset. For training on unlabeled web data, we collected a motion dataset named Solo-Dancer. We downloaded from YouTube 8 categories of 337 dancing videos, each one of the videos features only a single dancer. The total length of the videos add up to 11.5 hours. We then used an off-the-shelf 2D keypoints detector [3] to extract keypoints frame-by-frame to be used as our training data.

Sythesized dataset. We also perform the proposed unsupervised training pipeline on the synthetic Mixamo dataset [2] in order to quantitatively measure the transfer results with ground truth and baseline methods. The training set comprises of 32 characters, each character has 800 sequences and a total of 1.2 hours for each character.

*Equal contribution.

[1] https://yzhq97.github.io/transmomo

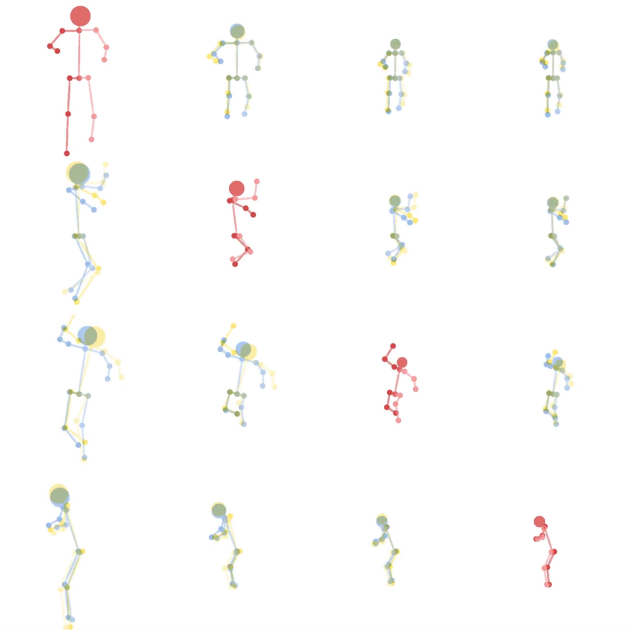

Figure 1. **Visualization of retargeting error computation with our model.** In this figure, we plotted input joint sequences (red) on the diagonal. Off the diagonal are the retargeted sequences (blue) from our model as well as the groud truth (yellow), where their overlapping areas become green. In this figure, sequences on the same row are expected to perform the same motion, while sequences on the same column are expected to share the same body structure.

2.2. Evaluation Metrics

MSE and MAE. For an inferred sequence $\hat{\mathbf{x}}$ and a groundtruth sequence $\mathbf{x}$

$$\text{MSE} = \frac{1}{2NT}\sum_{i,t}(\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t})^2$$

where i is the subscript of body joints and t is the subscript of time. Similarly,

$$\text{MAE} = \frac{1}{2NT}\sum_{i,t}|\mathbf{x}_{i,t} - \hat{\mathbf{x}}_{i,t}|$$

These two metrics are measured in the original scale of Mixamo dataset. The errors are computed after hip-alignment, as visualized in Figure 1.

FID. We calculate the Frechet Inception Distance (FID) [5] to evaluate the quality of generated frames. FID measures the perceptual distance between the generated frames and the real target frames, and smaller number represents higher visual consistency.

Figure 2. **Qualitative results of ablation study.** The first column gives the motion sources, and the other columns show corresponding results.

User study. For the quality of retargeted videos, we ask 100 volunteers to perform subjective pairwise A/B tests. For each method (4 baseline and 2 ours), we test 90 retargeted videos with the combination of 30 source and 3 target individuals. All the videos are 10 seconds in length. Participants choose which video has better motion consistency (between source videos and retargeted videos) in a pair of retargeted videos from two different methods. Source videos are also given to testers for reference. For each baseline method, 90 retargeted videos are compared 100 times by different participants against *our model*. Our model has two variants with different training sets (*i.e.*, Mixamo and SoloDancer), the results are shown in Table 1 in main paper as "User" and "User (wild)" respectively.

3. Qualitative Ablation Study

Besides testing standard MSE, we render the retargeted video for further comparison. As can be empirically observed in Fig. 2, the full model produces the results of the best quality. The cross reconstruction loss plays an essential role for disentanglement. The results without triplet loss show slightly degraded quality on the frame level. However, it is important to note that the triplet loss is used to smooth the structure and view code temporally, therefore stabilizing the generated video. The adversarial loss improves the plausibility of generated joint sequences, making them look more natural and realistic. Recall that the adversarial loss is added on randomly rotated output joint sequences to make the rotated output sequences indistinguishable from real data.

References

[1] Densepose: Dense human pose estimation in the wild. https://github.com/facebookresearch/DensePose/. 1

[2] Mixamo. https://www.mixamo.com/. 1

[3] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, pages 7297–7306, 2018. 1

[4] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody dance now. In *ICCV*, 2019. 1

[5] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014. 1