

Supplementary Material

ReMP: Reusable Motion Prior for Multi-domain 3D Human Pose Estimation and Motion Inbetweening

Hojun Jang¹ and Young Min Kim^{1,2}

¹ Dept. of Electrical and Computer Engineering, Seoul National University

² Interdisciplinary Program in Artificial Intelligence and INMC, Seoul National University

{j12040208, youngmin.kim}@snu.ac.kr

A. Implementation Details

A.1. Motion Parameters

We define our motion parameter to be a set of 6D pose parameter and the expanded root translation transition as written below:

$$M = \left[\theta_{6D}^{\text{flat}}, \text{MLP}_{3 \rightarrow 144}(\Delta x) \right] \in \mathbb{R}^{144+144} \quad (1)$$

Using 6D pose parameters is widely known to help the network to train and inference the rotation values [1], while using a root transition instead of the absolute root position value and even expanding its dimension is not a common sense. To justify our choice, we conducted an experiment and compare motion autoencoding performance of the motion prior when the root translation type (x or Δx) and the dimension (3 or 144) change. Table A.1 verifies our choice of using the dimension-expanded Δx by showing that the reconstruction errors are the best among all choices. Increasing the dimension of Δx enhanced the motion autoencoding performance especially for the root translation estimation.

A.2. Synthetic Dataset Generation

We report additional implementation details of the synthetic dataset generation process.

Point Clouds To create a synthetic point cloud dataset, we use the Open3D [11] library, which provides various functions for handling 3D data. First, we generate synthetic depth images of a moving SMPL [6] mesh. We position a virtual camera by randomly rotating it around a vertical axis and orient it to face the root position at the midpoint of the motion sequence. After setting the camera’s position and orientation, we capture depth images of the SMPL mesh

Type	Dim.	Pose [°]	Trans. [cm]	Joint [cm]
x	3	0.92	6.06	6.96
x	144	0.93	7.58	8.31
Δx	3	0.91	2.92	3.88
Δx	144	0.91	2.26	3.68

Table A.1. Verification of using an expanded Δx for our motion parameter.

sequence at a resolution of 640×480 . We then generate a depth point cloud consisting of 1,024 points from these depth images.

For the synthetic LiDAR scan dataset, we utilize the same depth images used for generating depth point clouds. To simulate the sparsity characteristic of LiDAR scans, we downsample the depth images by taking every fifth row and column. From these sparse depth images, we randomly sample points to create LiDAR scans, each consisting of 256 points.

IMUs For the synthetic IMU sensor dataset, we follow the procedure outlined in DIP [2]. We begin by attaching six virtual IMU sensors to specific vertices of the SMPL [6] mesh. The attachment points are the left arm (vertex: 1962), right arm (vertex: 5431), left leg (vertex: 1096), right leg (vertex: 4583), head (vertex: 412), and root (vertex: 3021). Next, we animate the motion sequence to record the acceleration and orientation of the virtual sensors. The generated sensor data are then paired with the SMPL parameters to form a synthetic dataset.

A.3. Hyperparameter Setup

We show the hyperparameter settings we used to train ReMP. Tables A.2 and A.3 show the hyperparameter setup

Train		Architecture		Loss	
N_{epoch}	2,000	T	40	w_{θ}	1.0
$N_{\text{decay},1}$	1,200	r_{mask}	0.8	$w_{\Delta\theta}$	10.0
$N_{\text{decay},2}$	1,600	D_z	128	w_x	10.0
lr	1e-4	$D_{\text{Tr},z}$	256	$w_{\Delta x}$	100.0
		$D_{\text{Tr},\text{ff}}$	1024	w_J	1.0
		$N_{\text{Tr},\text{layer}}$	4	w_V	1.0
		$N_{\text{Tr},\text{head}}$	8	$w_{\text{KL}}^{\text{prior}}$	0.4

Table A.2. Hyperparameter setup for the motion prior training phase. We classify the hyperparameters into three groups, hyperparameters related to the training, architecture, and the loss weights.

Train		Architecture		Loss	
N_{epoch}	2,000	T	40	w_{θ}	1.0
$N_{\text{decay},1}$	1,200	r_{mask}	0.2	$w_{\Delta\theta}$	10.0
$N_{\text{decay},2}$	1,600	D_z	128	w_x	10.0
lr	1e-4	D'_I	128	$w_{\Delta x}$	100.0
		$D_{\text{Tr},z}$	256	w_J	1.0
		$D_{\text{Tr},\text{ff}}$	1024	w_V	1.0
		$N_{\text{Tr},\text{layer}}$	4	$w_{\text{KL}}^{\text{reuse}}$	0.1
		$N_{\text{Tr},\text{head}}$	8	w_{β}	0.1

Table A.3. Hyperparameter setup for the motion prior reusing phase. We classify the hyperparameters into three groups, hyperparameters related to the training, architecture, and the loss weights.

for the motion prior training and the reusing prior, respectively.

N_{epoch} is the total epoch of the training and when the training reaches the epoch of $N_{\text{decay},1}$ or $N_{\text{decay},2}$, the learning rate lr decreases to $lr/4$ and $lr/10$, respectively. We use a sequence of time length T to be 40 and mask out the sequence with the ratio of r_{mask} using a `key_padding_mask` in the transformer [8] encoder. $D_{\text{Tr},z}$, $D_{\text{Tr},\text{ff}}$, $N_{\text{Tr},\text{layer}}$, and $N_{\text{Tr},\text{head}}$ refer to the intermediate latent dimension, feedforward size, number of layers, and number of heads in the transformer network, respectively. The rest follow the notations in the main paper.

B. Experiment Details

B.1. Dataset

We use AMASS [7] to train our model. AMASS dataset is a large-scale dataset which contains SMPL parameters of more than 20 different datasets, including CMU dataset [4]. Since we test ReMP on the synthetic CMU dataset, we use the rest to train the motion prior and also the reusing part. Our motion prior learns the motion within 40 frames at 10 fps, so we split the sequence which is longer than 8 seconds into several pieces and drop the sequence shorter than

4 seconds to make each sequence to be minimum 4 seconds long. Therefore, the number of sequence we used to train our model is 17,240 and the number of sequence in the synthetic CMU dataset is 2,962.

B.2. Baselines

We use the same baselines in point cloud input scenarios for both depth images or LiDAR scans. VoteHMR [5] addresses challenges from occlusion and measurement noise in single-view point cloud measurements by segmenting the input point cloud into parts classified as different joints. Additionally, it requires point segmentation for training, introducing dependencies that our method does not require. Zuo *et al.* [12] reconstruct human body mesh surfaces from point cloud inputs. It first regresses parameters with a neural network and then refine them through optimization employing probabilistic self-supervised loss functions. The optimization step enhances robustness to outliers but incurs significant computational overhead. Both methods estimate pose parameters for individual frames, lacking temporal coherence. In contrast, Jang *et al.* [3] concurrently regress parameters for a temporal sequence, leveraging temporal information for more accurate and smooth motion. However, it does not employ efficient encoding schemes and cannot be applied to different sensor modalities.

DIP-IMU [2] is the first deep learning-based method for human pose estimation from IMU inputs, using a bidirectional RNN architecture to estimate pose parameters, but lacks the ability to estimate root translation. TransPose [10] and PIP [9] enhance results by incorporating physical constraints to recover human motion, which enables root translation estimation. TransPose estimates foot-ground contact probability and root joint velocity, while PIP includes a physics-aware motion optimizer that refines motion using a torque-controlled floating-base simulated character model and a proportional-derivative (PD) controller for better tracking accuracy and physical plausibility. Every IMU baseline excludes shape parameter estimation, as the IMU sensor input does not contain an information about body shape. Therefore, we also use ground truth shape parameter for the experiments on IMU sensors.

C. Additional Results

We provide additional results of every experiment we conducted with the supplementary video. Since we focus on the motion, videos show the result more effectively, offering better views to compare with the baselines.

References

- [1] Thorsten Hempel, Ahmed A. Abdelrahman, and Ayoub Al-Hamadi. 6d rotation representation for unconstrained head pose estimation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2496–2500, 2022. 1

- [2] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 37:185:1–185:15, Nov. 2018. First two authors contributed equally. [1](#), [2](#)
- [3] Hojun Jang, Minkwan Kim, Jinseok Bae, and Young Min Kim. Dynamic mesh recovery from partial point cloud sequence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15074–15084, October 2023. [2](#)
- [4] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015. [2](#)
- [5] Guanze Liu, Yu Rong, and Lu Sheng. VoteHmr: Occlusion-aware voting network for robust 3d human mesh recovery from partial point clouds. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 955–964, 2021. [2](#)
- [6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. [1](#)
- [7] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5442–5451, Oct. 2019. [2](#)
- [8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. [2](#)
- [9] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13167–13178, June 2022. [2](#)
- [10] Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 40(4), 08 2021. [2](#)
- [11] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. [1](#)
- [12] Xinxin Zuo, Sen Wang, Qiang Sun, Minglun Gong, and Li Cheng. Self-supervised 3d human mesh recovery from noisy point clouds. *arXiv preprint arXiv:2107.07539*, 2021. [2](#)