

SimPoE: Simulated Character Control for 3D Human Pose Estimation

Supplementary Material

Ye Yuan¹ Shih-En Wei² Tomas Simon² Kris Kitani¹ Jason Saragih²

¹Carnegie Mellon University ²Facebook Reality Labs

<https://www.ye-yuan.com/simpoe>

1. In-House Motion Dataset and Kinematic Pose Estimator

In-House Dataset. Our in-house motion dataset uses a more complex skeleton model (with twice as many joints, including fingers) than SMPL [7]. To recover the human skinning mesh model of each subject, we use an offline process that uses multiview 3D reconstruction to produce a person-specific skinning template with 32 bone scaling parameters based on non-rigid ICP deformation [1] and linear blend skinning [4]. For motion, we solve for 94 local joint angles (degrees of freedom) and the global 3D position of 159 joints, including finger joints, for every frame.

Kinematic Pose Estimator. Since existing kinematic pose estimators, such as VIBE [6], cannot be directly applied to the in-house dataset due to the dataset’s more complex skeletons and skinning models, we design a simple kinematic tracker (“KinPose” in the main paper) that also uses monocular inputs to produce kinematic pose estimates. The model does not have any temporal component, outputting both 2D keypoint heatmaps and joint angles frame-by-frame. These two outputs are required by our approach (SimPoE) and NeurGD [10]. Below, we detail the network architecture and training procedure of this model, which are typical for such models as the performance of SimPoE is not sensitive to these design choices.

Network Architecture. We use a 3-stage cascaded network [11] with a backbone based on ResNet-50 [2]. The output of the network at each stage is a tensor of $m + n$ channels with a spatial size that is 8x smaller than the input image, where $m = 77$ is the number of heatmap channels for 2D keypoints (a subset of the 159 joints), and $n = 94 + 6$ is the number of local joint angles and global pose dimensions. Each of the 94 angle channels is an “angle map” corresponding to a joint. The final output angle (scalar) is calculated by summing the element-wise product between an angle map and its corresponding keypoint heatmaps, where the correspondence is defined based on the skeleton. This design is a type of attention mechanism, which encourages the model to predict the angle of a joint based only on relevant image regions.

Training. At each stage of the network, we apply L_2 losses on heatmaps, 2D keypoints, 3D joint positions, and joint angles to train the model, similar to VIBE [6].

2. Additional Implementation Details

Parameter	Value
Num. of time steps	50000
Num. of epochs	2000
Num. of policy updates per epoch	10
Policy step size	5×10^{-5}
Value step size	3×10^{-4}
PPO clip ϵ	0.2
Discount factor γ	0.95
GAE coefficient λ	0.95
Reward weights $(\alpha_p, \alpha_v, \alpha_j, \alpha_k)$ (Human3.6M)	(30, 0.2, 100, 0.02)
Reward weights $(\alpha_p, \alpha_v, \alpha_j, \alpha_k)$ (In-house)	(60, 0.2, 300, 0.02)
Elements of diagonal covariance Σ (Human3.6M)	0.1
Elements of diagonal covariance Σ (In-house)	0.05
Residual force scale	500

Table 1. Hyperparameters for experiments on Human3.6M [3] and our in-house motion dataset.

For both Human3.6M and our in-house motion dataset, our method uses the same hyperparameter settings unless stated otherwise. Table 1 summarizes the hyperparameter setting.

Policy Network. The learnable parts in the policy network are the two MLPs, *i.e.*, \mathcal{U}_θ inside the kinematic refinement unit and \mathcal{V}_θ inside the control generation unit. We use ReLU activations for both \mathcal{U}_θ and \mathcal{V}_θ . The MLP \mathcal{U}_θ consists of hidden layers with size (256, 512, 256). The MLP \mathcal{V}_θ contains hidden layers with size (2048, 1024).

Policy Training. For Human3.6M, we train a single policy using data from all the training subjects and directly transfer the policy to test subjects, so it is a cross-subject experiment. For our in-house motion dataset, due to the large variation of body proportion and shape, we train a model for each subject using subject-specific data and test on separate data. All baselines are trained using the same data as our method. For learning the policy, each RL episode is constructed by randomly sampling a video segment of 200 frames from all training data. For the initial pose \mathbf{q}_1 of the character, we initialize it to the refined kinematic pose $\tilde{\mathbf{q}}_1^{(n)}$. For the initial velocity $\dot{\mathbf{q}}_1$, we set it to the kinematic velocity $\tilde{\dot{\mathbf{q}}}_1^{(n)}$ computed using finite differences. The episode is terminated when the end frame is reached or the character’s root height is 0.5 below the root height of the kinematic pose (*i.e.*, to detect if the character has lost balance). We train the policy π_θ for 2000 epochs. For each epoch, we keep collecting data by sampling RL episodes until the total number of time steps reaches 50000. The reward weighting factors $(\alpha_p, \alpha_v, \alpha_j, \alpha_k)$ are set to (30, 0.2, 100, 0.02) for Human3.6M and (60, 0.2, 300, 0.02) for the in-house dataset. For Human3.6M, we only have access to ground-truth 3D joint positions but not ground-truth joint angles, so we use the refined kinematic pose as pseudo-ground truth (for regularization) when computing rewards that need ground-truth joint angles. The elements of the policy’s diagonal covariance matrix Σ are set to 0.1 for Human3.6M and 0.05 for our in-house motion dataset. The residual forces $\boldsymbol{\eta}_t$ output by the policy is scaled by 500 before being input to the physics simulator. We use the proximal policy optimization (PPO [9]) to learn the policy π_θ . The discount factor for the Markov decision process (MDP) is 0.95. We use the generalized advantage estimator GAE(λ) [8] to compute the advantage estimate for policy gradient and the GAE coefficient λ is 0.95. At each epoch, the policy is updated 10 times using Adam [5] with a step size 5×10^{-5} . The clipping coefficient ϵ in PPO is set to 0.2. Since PPO is an actor-critic based method, it also learns a value function which mirrors the design of the policy but outputs a single value estimate. The value function is updated using Adam with a step size 3×10^{-4} whenever the policy is updated.

References

- [1] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1
- [3] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 1
- [4] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O’Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46, 2007. 1
- [5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2
- [6] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 1
- [7] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 1
- [8] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 2
- [9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2
- [10] Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, YiLi Fu, and Tao Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5349–5358, 2019. 1
- [11] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 1