A ACCELERATION READINGS AND SIM-TO-REAL

Moving Average Filtering. Since real IMU data paired with ground-truth full-body motions are small in size, following previous work [Huang et al. 2018], we place virtual IMU sensors on virtual characters driven by captured motions to synthesize IMU orientation and acceleration readings. Using the AMASS [Mahmood et al. 2019] motion dataset (a collection of smaller motion capture datasets), we create a large-scale synthetic IMU dataset for training our model.

However, synthetic and real IMU data exhibit vastly different noise profiles. Acceleration data in the real dataset are noisy, but not in the same way as the noise in the synthetic dataset, which is caused by double differentiation of mocap data (Figure 6 Top). On the other hand, orientation data are usually less noisy because they are processed by the in-sensor Kalman filter [Kalman et al. 1960]. Previous work [Huang et al. 2018; Yi et al. 2021] recognized this distribution mismatch problem and proposed to first train the model exclusively on the synthetic data and then finetune it on a smaller real dataset. This two-step solution leads to a more complex training procedure that requires careful tuning to avoid overfitting the real dataset.

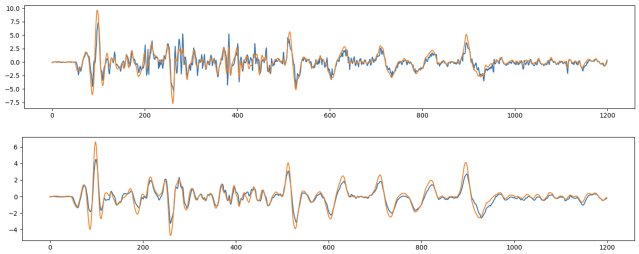

Figure 6: Example of synthesized (orange) and real (blue) acceleration data, before (top) and after (bottom) moving average filtering.

We found that simply running an average filter on both synthetic and real acceleration data (with window length of 11 in our implementation) would bring the two data sources sufficiently close to each other (Fig. 6 Bottom). We then train the model only once on the combined dataset. Combining both data sources simplifies training from two stages to one stage, and avoids the risk of catastrophic forgetting during finetuning.

In practice, filtering causes latency during real-time inference, as computing moving average requires future IMU readings. We use 5 times steps (83ms) of future readings, the same requirement as [Huang et al. 2018; Yi et al. 2021], though they require future readings as part of model input while we merely use them for filtering.

Summing Up (Integrate) Past Accelerations. Another issue we discover for non-flat terrain motions is that the sensor readings (both orientation and acceleration) during a stair step is much similar to a normal flat-ground step, especially in the case of real, noisy IMU data. Note that pelvis also accelerates up and down in a normal walking step resembling an inverted pendulum. However, if we sum up the raw IMU acceleration readings within a small window of recent history (e.g. past 0.5s), similar to "integrating" acceleration to delta velocities, we could observe a more different signal shape between stairs and normal steps. Empirically, we find adding this additional history sum features for each channel, increasing acceleration features from $\mathbb{R}^{18}$ to $\mathbb{R}^{36}$ (concatenating with filtered accelerations), improves stair recognition on real hardware.

B MODEL DETAILS

We use the AMASS dataset to generate synthetic training data following the smoothing procedure in Appendix A. It consists of over a dozen different motion capture datasets performing a variety of activities. In addition, we include 8 out of 10 subjects' data from the DIP dataset. We use pyBullet [Coumans and Bai 2016] for calculating forward kinematics during data synthesis, SBP label generation, root correction, and final visualizations. As the DIP real IMU data do not have root motion, we use a pre-trained model to label pseudo ground-truth SBPs for the DIP motions.

We use standard loss functions for the model outputs, i.e., mean-squared error for joint rotations, mean-squared error for $\boldsymbol{v}_t$ and Cartesian elements of $\boldsymbol{c}_t$, and binary cross-entropy for binary elements of $\boldsymbol{c}_t$, (i.e. b_t). Specifically, for joint rotations,

$$\mathcal{L}_J = \|\boldsymbol{q}_t - \bar{\boldsymbol{q}}_t\|_2^2,$$

where $\bar{\boldsymbol{q}}_t$ is the ground-truth full-body joint rotations, represented as first two columns (6D) of each rotation matrix as noted previously in the main text. Similarly for root velocities,

$$\mathcal{L}_R = \|\boldsymbol{v}_t - \bar{\boldsymbol{v}}_t\|_2^2,$$

and for the SBP predictions $\boldsymbol{c}_t$,

$$\mathcal{L}_C = \sum_{i=1}^{5} \left\| \boldsymbol{r}_t^{(i)} - \bar{\boldsymbol{r}}_t^{(i)} \right\|_2^2 + \sum_{i=1}^{5} (-\bar{b}_t^{(i)} \log b_t^{(i)} - (1-\bar{b}_t^{(i)}) \log(1-b_t^{(i)})).$$

Since our model during training time predicts a whole trajectory window, we experimented with a jerk loss penalizing deviation of neighboring frames, but it did not produce visible improvements. This might be due to the fact that during test time we still only use the last prediction at each step. Instead, we pass our output through an exponential moving average filter as post processing, at the expense of slight increase in joint accuracy errors (Appendix F).

Our model is trained in PyTorch [Paszke et al. 2019] using the Adam optimizer [Kingma and Ba 2014], with a batch size of 256 and a learning rate of 0.0001 multiplied with a cosine schedule [Loshchilov and Hutter 2016]. We perform training for 1000 epochs, which takes around 6 hours with a GeForce GTX 2080Ti GPU. Once trained, our model is small enough to run at 60 fps on a 2080Ti machine, with bottleneck being the python wrapper of pyBullet. Our model uses max window size $M = 39$. It contains a total number of 3, 677, 315 parameters, comparing to 4, 798, 771 in TransPose and 10, 801, 934 in DIP.

Note that our model requires an initial full-body pose given in the first step of prediction. In practice this is always the case since the sensors need to be calibrated with a T pose before each use, as they are allowed to be slightly differently worn. See Appendix D for more details.

C TERRAIN GENERATION DETAILS

Grid size is empirically set to 0.1m, and number of grids $L \times L$ is set so that terrain is large enough. We initialize the height map $\boldsymbol{H}$ with all zero values, where zero is set to the initial root height minus a constant height w. w represents the lower bound of how low the subject could possibly reach in this capture. We assume no cluster means are below this value. If the user can provide a tighter w (e.g. starting the capture on the lowest ground plane), we can generate a more visually pleasing terrain by producing no dents lower than the specified ground plane. During real-time demos, w is set to be tight, to indicate that we know the motion will not go lower than the starting ground plane.

If there were only two SBPs with different heights, Voronoi diagram will render a 1-step stair that is infinitely wide. For aesthetics, we limit the area each new SBP can influence to 1m ×1m, which could be nevertheless still wider than the real stairs in scene. While we can arbitrarily make this influence region narrower, we note that, without additional information, both are equally *plausible*, and new SBPs can always crop the terrain narrower with more information streaming in (Video 1m40s).

For a newly active SBP, we ignore it for t_0 seconds before using it for the terrain algorithm, to allow it to settle in height. t_0 is 50 frames or the the moment SBP becomes inactive, whichever comes earlier. The pelvis SBP is only used in terrain generation if it is > 0.2m away from the feet, to avoid building terrains at the pelvis height when the subject is standing still.

D SENSOR CALIBRATION

When testing on real hardware, as the raw sensor readings are in different coordinate frames from the frame of system input, calibration is needed to obtain the offset transforms between the coordinate frames beforehand. We adopt a slightly different IMU calibration procedure from previous works that is nevertheless still straightforward to explain.

We start from defining a few coordinate frames. Let G_n be the base (i.e. identity) frame of each of the 6 sensors (for the Xsens sensors we used, identity orientation could mean different poses per sensor). Let G_p be any fixed global frame the user specifies, whose x axis indicates the specified front, y axis corresponds to the left, and z axis corresponds to the upwards. (Note this axis definition is different from DIP and TransPose models.) Let S_t be the sensor frame, while S_0 defines the sensor frame during T-pose calibration. Let B_t be the bone frame, while B_0 defines the bone frame during T-pose calibration. We omit the sensor indices j (e.g. $G_n^{(j)}$, $S_t^{(j)}$) since calibration is agnostic to each sensor.

Using these notations, $\boldsymbol{R}_{G_n}^{S_t}$ represents the raw sensor orientation reading based from frame G_n, and $\boldsymbol{a}_{S_t}$ represents the raw acceleration reading which is always local in sensor frame. The system however expects both bone orientation and acceleration reading in G_p, i.e., $\boldsymbol{R}_{G_p}^{B_t}$ and $\boldsymbol{a}_{G_p}$. We have the following relations:

$$\boldsymbol{R}_{G_p}^{B_t} = \boldsymbol{R}_{G_p}^{G_n} \boldsymbol{R}_{G_n}^{S_t} \boldsymbol{R}_{S_t}^{B_t},$$
$$\boldsymbol{a}_{G_p} = \boldsymbol{R}_{G_p}^{G_n} \boldsymbol{R}_{G_n}^{S_t} \boldsymbol{a}_{S_t} - \bar{\boldsymbol{a}}_{S_t},$$

where we note that $\bar{\boldsymbol{a}}_{S_t}$ is the constant acceleration bias in global frame, usually just the gravitational acceleration. From these relations, it should be clear that the goal of calibration is simply to obtain $\boldsymbol{R}_{G_p}^{G_n}$ and $\boldsymbol{R}_{S_t}^{B_t}$ before each system run.

In the first calibration step, we place all sensors to align with the specified global frame so that $\boldsymbol{R}_{G_n}^{S} = \boldsymbol{R}_{G_n}^{G_p}$, and obtain $\boldsymbol{R}_{G_p}^{G_n}$ which is simply $\{\boldsymbol{R}_{G_n}^{S}\}^T$. Following [Yi et al. 2021], we keep all sensors still on ground for three seconds and take the average reading.

Next, to obtain $\boldsymbol{R}_{S_t}^{B_t}$, the user wears all six sensors and stand in a T pose, facing the same "front" as G_p. We assume that the sensor will stay static with respect to the bone throughout the entire system run, therefore $\boldsymbol{R}_{S_t}^{B_t} = \boldsymbol{R}_{S_0}^{B_0}$. Since the orientation of each bone at a standard T pose, $\boldsymbol{R}_{G_p}^{B_0}$, is known, we are able to obtain $\boldsymbol{R}_{S_0}^{B_0}$ from the T-pose raw sensor reading $\boldsymbol{R}_{G_n}^{S_0}$ using:

$$\boldsymbol{R}_{S_0}^{B_0} = \{\boldsymbol{R}_{G_n}^{S_0}\}^T \boldsymbol{R}_{G_n}^{G_p} \boldsymbol{R}_{G_p}^{B_0},$$

where same as the first step, T pose is maintained for three seconds and we use the average reading for $\boldsymbol{R}_{G_n}^{S_0}$.

E ADDITIONAL ANALYSIS

We present results of two additional experiments in this section. First to showcase how much the performance our autoregressive model will degrade over time, we repeat the quantitative experiment of Table 1 but on random 3000-frame (50s) windows of each motion, instead of 600 frames (10s). Note that since many test motions are shorter than 50s, this experiment setting may unevenly bias statistics. For brevity, the DIP model is not included in this comparison:

Table 3: Comparison of model performance on evaluation motion segments of maximum length 50 seconds. Bold numbers indicate the best performing entries.

Our TIP Model

	DIPEval	TotalCapture	DanceDB
joint angle errors (degree)	**12.33555**	**9.46942**	**15.28491**
joint position errors (cm)	**5.86926**	**5.40289**	**8.23641**
root errors in 2s (meter)		**0.08545**	**0.09504**
root errors in 5s (meter)		**0.16679**	**0.20369**
root errors in 10s (meter)		**0.20338**	**0.38935**
joint position jitter (m/s^3)	0.84848	0.80672	**1.39043**
root jitter (m/s^3)	0.64593	**0.64609**	**0.95740**

TransPose Model

	DIPEval	TotalCapture	DanceDB
joint angle errors (degree)	12.78403	11.56577	17.22182
joint position errors (cm)	6.16507	5.76287	8.35314
root errors in 2s (meter)		0.18543	0.14899
root errors in 5s (meter)		0.32042	0.28216
root errors in 10s (meter)		0.32111	0.45332
joint position jitter (m/s^3)	**0.57619**	**0.76578**	1.44662
root jitter (m/s^3)	**0.49804**	0.70235	1.29385

Reading the numbers from Table 3, degradation of model performance is minimal on longer motions, and the statistics trends between our model and TransPose remain unchanged. As a side note, the DanceDB dataset contains more short motions, making the sampling a random 50s segment more likely to cover the beginnings of motions. We therefore see both TIP and TransPose have improved *root errors in 2s* since the motions usually start from standing and are less dynamic in the first two seconds.

Second, to showcase the benefit of acceleration preprocessing, we perform an ablation study where we remove the average filtering and summation operations from our TIP system, both during training and test time.

Table 4: Ablation of model performance on evaluation motion segments of 10s. Bold numbers indicate the best performing entries.

Our TIP Model

	DIPEval	TotalCapture	DanceDB
joint angle errors (degree)	**12.09586**	**8.91642**	**15.57031**
joint position errors (cm)	**5.82242**	**5.14566**	**8.50089**
root errors in 2s (meter)		**0.08031**	0.20295
root errors in 5s (meter)		**0.1351**	0.29681
root errors in 10s (meter)		**0.19446**	0.35759
joint position jitter (m/s^3)	0.8823	**0.75075**	**1.43867**
root jitter (m/s^3)	0.66211	**0.61108**	**0.98474**

Our TIP Model, w/o Acceleration Preprocessing

	DIPEval	TotalCapture	DanceDB
joint angle errors (degree)	13.02724	9.18290	15.67625
joint position errors (cm)	6.35219	5.29268	8.58611
root errors in 2s (meter)		0.09096	**0.16825**
root errors in 5s (meter)		0.18015	**0.25855**
root errors in 10s (meter)		0.20726	**0.34444**
joint position jitter (m/s^3)	**0.85845**	0.79335	1.44460
root jitter (m/s^3)	**0.64661**	0.63560	1.00417

From Table 4, We see a visible improvement from preprocessing the raw acceleration readings on real-IMU datasets (DIPEval & TotalCapture). As expected, preprocessing is unimportant for synthesized IMU data (DanceDB).

Qualitative comparisons between our method and TransPose are presented using the following two representative motions (Figure 7, Video 3m33s). Our TIP model can generate a more stable sitting posture by making better use of its own past predictions and utilizing run-time IK correction (Figure 7 Top). Figure 7 Bottom shows that our algorithm is terrain agnostic while TransPose assumes a flat ground and uses this assumption to correct the algorithm's vertical root prediction.

F DISCUSSIONS

Though we have shown clear improvement on existing challenges of temporal consistency due to ambiguity, dynamic motion coverage, and terrain coverage, our system still has a few drawbacks for future work. First, it tends to underestimate the terrain height rather than overestimate (e.g. Video 3m52s) - collecting more annotated real

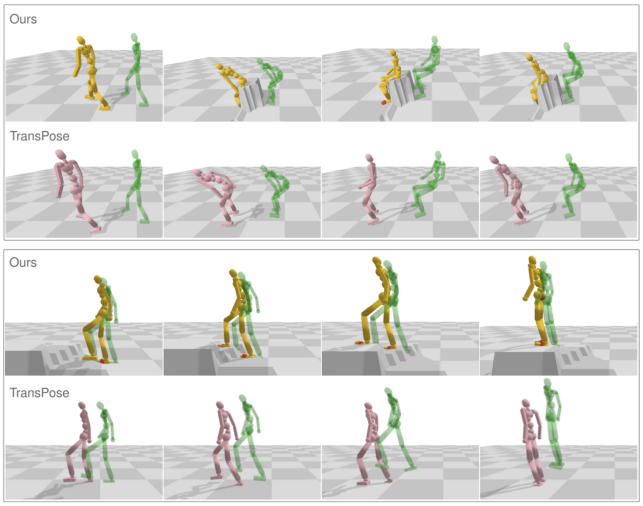

Figure 7: Motion reconstruction for sitting on a chair (from real IMU data, top) and climbing steps (from synthesized IMU data, bottom). Our character is shown in yellow, TransPose in purple and Ground-Truth motion is shown in green. The red spheres are predicted SBPs. Playing back motion on reconstructed terrains.

IMU data on various terrain types, and increasing training samples with uneven terrains through data upsampling, could both help improve terrain reconstruction. Second, terrain height estimations remain challenging since they solely depend on motion prediction, and are susceptible to sensor noises. For example, locomotion on a slightly bumpy ground versus on a flat ground is theoretically near-ambiguous given IMU's noise level (Video 4m6s). Third, our motion reconstruction quality on real hardware can degrade on motion types that are rare in training, thus affecting the quality of generated terrains (Video 4m21s). Finally, though our work does not claim contribution over the jitter level of reconstructed motions, the smoothing filter during post-processing is far from ideal and hurts our motion accuracy by effectively increasing latency.

Another very visible problem we observe is the model's bias to body types. Our synthesized data were generated from virtual characters with random heights sampled from 1.6m to 1.8m. We observed that the algorithm generalizes better to taller users than shorter ones. We hypothesize that this phenomenon is due to the magnitude of acceleration, as the model might be more easily confused by smaller signals from a shorter user. Similarly, existing real IMU datasets might have a bias in human shapes. Some personalized training and finetuning of the model may eventually be necessary for reconstructing more accurate and detailed motion for each individual user.

The terrain generated from our algorithm is "plausible" in the sense that it cannot distinguish, sorely from IMU readings, if the foot is resting on a terrain or simply staying stationary in air (Video 4m58s). An algorithm that takes the distribution of commonly seen environments into consideration could guide our system to generate more likely terrains in such ambiguous cases.

REFERENCES

Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. 2021. A Spatio-temporal Transformer for 3D Human Motion Prediction. *International Conference on 3D Vision (3DV)* (2021).

Sheldon Andrews, Ivan Huerta, Taku Komura, Leonid Sigal, and Kenny Mitchell. 2016. Real-Time Physics-Based Motion Capture with Sparse Sensors. In *Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016) (CVMP 2016)*. Article 5.

Eric R. Bachmann, Robert B. McGhee, Xiaoping Yun, and Michael J. Zyda. 2001. Inertial and Magnetic Posture Tracking for Inserting Humans into Networked Virtual Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '01)*. 9–16.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. (2020). arXiv:2005.14165 [cs.CL]

Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

Young-Woon Cha, Husam Shaik, Qian Zhang, Fan Feng, Andrei State, Adrian Ilie, and Henry Fuchs. 2021. Mobile. Egocentric Human Body Motion Reconstruction Using Only Eyeglasses-mounted Cameras and a Few Body-worn Inertial Sensors. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*.

Adrian Hilton Charles Malleson, John Collomosse. 2020. Real-Time Multi-person Motion Capture from Multi-view Video and IMUs. *International Journal of Computer Vision* 128 (06 2020).

Vasileios Choutas, Federica Bogo, Jingjing Shen, and Julien Valentin. 2021. Learning to Fit Morphable Models. *CoRR* abs/2111.14824 (2021). arXiv:2111.14824 https://arxiv.org/abs/2111.14824

Erwin Coumans and Yunfei Bai. 2016. Pybullet, a python module for physics simulation for games, robotics and machine learning. (2016).

Michael B. Del Rosario, Heba Khamis, Phillip Ngo, Nigel H. Lovell, and Stephen J. Redmond. 2018. Computationally Efficient Adaptive Error-State Kalman Filter for Attitude Estimation. *IEEE Sensors Journal* 18, 22 (2018), 9332–9342.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A Generative Model for Music. arXiv:2005.00341 [eess.AS]

Andrea Dittadi, Sebastian Dziadzio, Darren Cosker, Ben Lundell, Tom Cashman, and Jamie Shotton. 2021. Full-Body Motion From a Single Head-Mounted Device: Generating SMPL Poses From Partial Observations. In *International Conference on Computer Vision 2021*.

H. Durrant-Whyte and T. Bailey. 2006. Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine* 13, 2 (2006), 99–110. https://doi.org/10.1109/MRA.2006.1638022

E. Foxlin. 1996. Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. 185–194.

Andrew Gilbert, Matthew Trumble, Charles Malleson, Adrian Hilton, and John Collomosse. 2019. Fusing Visual and Inertial Sensors with Semantics for 3D Human Pose Estimation. *International Journal of Computer Vision* 127 (04 2019), 1–17.

Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7297–7306.

Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. 2021. Human POSEitioning System (HPS): 3D Human Pose Estimation and Self-localization in Large Scenes from Body-Mounted Sensors. In *CVPR*.

Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Christian Theobalt. 2013. Real-Time Body Tracking with One Depth Camera and Inertial Sensors. In *2013 IEEE International Conference on Computer Vision*. 1105–1112.

Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. 2020. Learned Motion Matching. *ACM Trans. Graph.* 39, 4, Article 53 (jul 2020), 13 pages. https://doi.org/10.1145/3386569.3392440

Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. *ACM TOG* 37, 6 (12 2018).

Rudolph Emil Kalman et al. 1960. A new approach to linear filtering and prediction problems [J]. *Journal of basic Engineering* 82, 1 (1960), 35–45.

Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. 2019. Learning 3D Human Dynamics from Video. In *Computer Vision and Pattern Recognition (CVPR)*.

Manuel Kaufmann, Yi Zhao, Chengcheng Tang, Lingling Tao, Christopher Twigg, Jie Song, Robert Wang, and Otmar Hilliges. 2021. EM-POSE: 3D Human Pose Estimation from Sparse Electromagnetic Trackers. In *International Conference on Computer Vision (ICCV)*.

Seong Uk Kim, Hanyoung Jang, Hyeonseung Im, and Jongmin Kim. 2021. Human motion reconstruction using deep transformer networks. *Pattern Recognition Letters* 150 (2021), 162–169.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Benoît Le Callennec and Ronan Boulic. 2006. Robust kinematic constraint detection for motion data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 281–290.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. 2021. AI Choreographer: Music Conditioned 3D Dance Generation with AIST++.

Huajun Liu, Xiaolin Wei, Jinxiang Chai, Inwoo Ha, and Taehyun Rhee. 2011. Realtime Human Motion Control with a Small Number of Inertial Sensors. In *Symposium on Interactive 3D Graphics and Games (I3D '11)*. 133–140.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM TOG* 34, 6 (Oct. 2015), 248:1–248:16.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. 2021. Dynamics-Regulated Kinematic Policy for Egocentric Pose Estimation. In *NeurIPS*.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *ICCV*. 5442–5451.

Charles Malleson, Marco Volino, Andrew Gilbert, Matthew Trumble, John Collomosse, and Adrian Hilton. 2017. Real-time Full-Body Motion Capture from Video and IMUs. In *Int. Conf. 3D Vis.*

Deepak Nagaraj, Erik Schake, Patrick Leiner, and Dirk Werth. 2020. An RNN-Ensemble Approach for Real Time Human Pose Estimation from Sparse IMUs. In *Proceedings of the 3rd International Conference on Applications of Intelligent Systems* (Las Palmas de Gran Canaria, Spain) *(APPIS 2020)*. Article 32, 6 pages.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019), 8026–8037.

Mathis Petrovich, Michael J. Black, and Gül Varol. 2021. Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In *International Conference on Computer Vision (ICCV)*. 10985–10995.

Gerard Pons-Moll, Andreas Baak, Juergen Gall, Laura Leal-Taixé, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn. 2011. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *2011 International Conference on Computer Vision*. 1243–1250.

Gerard Pons-Moll, Andreas Baak, Thomas Helten, Meinard Müller, Hans-Peter Seidel, and Bodo Rosenhahn. 2010. Multisensor-fusion for 3D full-body human motion capture. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 663–670.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. 2021. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11488–11499.

Rokoko. n d. Rokoko https://www.rokoko.com/. Last visited: 08/26/2022.

Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. 2020. PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time. *ACM TOG* 39, 6 (12 2020).

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958. http://dl.acm.org/citation.cfm?id=2627435.2670313

Jonathan Taylor, Jamie Shotton, Toby Sharp, and Andrew Fitzgibbon. 2012. The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 103–110.

Matt Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *BMVC*.

Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, Andre Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. 2021. Transflower: Probabilistic Autoregressive Dance Generation with Multimodal Attention. *ACM Trans. Graph.* 40, 6, Article 195 (dec 2021), 14 pages. https://doi.org/10.1145/3478513.3480570

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

Vicon. n d. Vicon Motion Systems https://www.vicon.com/. Last visited: 08/26/2022.

Rachel V. Vitali, Ryan S. McGinnis, and Noel C. Perkins. 2021. Robust Error-State Kalman Filter for Estimating IMU Orientation. *IEEE Sensors Journal* 21, 3 (2021), 3561–3569.

Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. 2007. Practical Motion Capture in Everyday Surroundings. *ACM Trans. Graph.* 26, 3 (2007).

Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In *European Conference on Computer Vision (ECCV)*.

Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. 2016. Human Pose Estimation from Video and IMUs. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)* (jan 2016).

Timo von Marcard, Bodo Rosenhahn, Michael Black, and Gerard Pons-Moll. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. *Computer Graphics Forum 36(2), Proceedings of the 38th Annual Conference of the European Association for Computer Graphics (Eurographics)* (2017), 349–360.

Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. 2012. Accurate Realtime Full-Body Motion Capture Using a Single Depth Camera. *ACM Trans. Graph.* 31, 6, Article 188 (nov 2012).

Xsens. n d. Xsens https://www.xsens.com/. Last visited: 08/26/2022.

Dongseok Yang, Doyeon Kim, and Sung-Hee Lee. 2021. LoBSTr: Real-time Lower-body Pose Prediction from Sparse Upper-body Tracking Signals. *Computer Graphics Forum* (2021). https://doi.org/10.1111/cgf.142631

Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. 2022. Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xinyu Yi, Yuxiao Zhou, and Feng Xu. 2021. TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors. *ACM TOG* 40, 4 (8 2021).

Zhe Zhang, Chunyu Wang, Wenhu Qin, and Wenjun Zeng. 2020. Fusing Wearable IMUs with Multi-View Images for Human Pose Estimation: A Geometric Approach. In *CVPR*.

Zerong Zheng, Tao Yu, Hao Li, Kaiwen Guo, Qionghai Dai, Lu Fang, and Yebin Liu. 2018. HybridFusion: Real-Time Performance Capture Using a Single Depth Sensor and Sparse IMUs. In *European Conference on Computer Vision (ECCV)*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). 389–406.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5745–5753.