WHAC: World-grounded Humans and Cameras – Supplementary Material –

Wanqi Yin[⋆,1,2], Zhongang Cai[*,1,3], Ruisi Wang[1], Fanzhou Wang[1], Chen Wei[1], Haiyi Mei[1], Weiye Xiao[1], Zhitao Yang[1], Qingping Sun[1], Atsushi Yamashita[2], Ziwei Liu[3], Lei Yang[1]

[1] SenseTime Research, [2] The University of Tokyo,
[3] S-Lab, Nanyang Technological University

A Overview

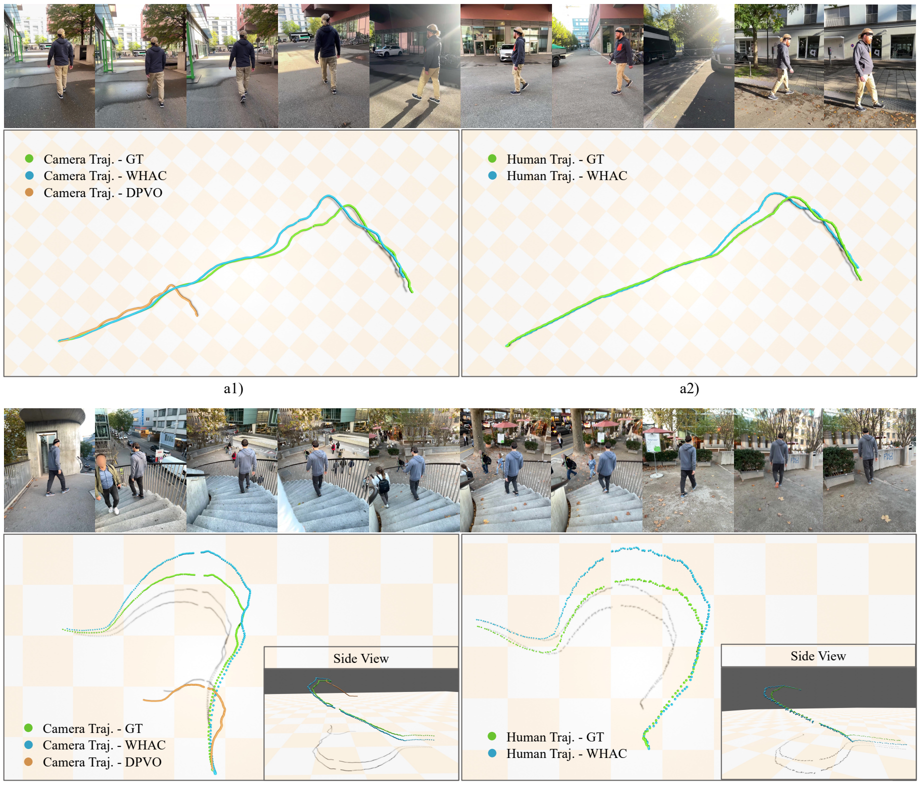
Given the space constraints in the main paper, we provide additional information and details in this supplementary material: more results visualization on the EMDB dataset and WHAC-A-Mole dataset in Sec. B; further elaboration on the MotionVelocimeter in Sec. C; the average inference speed in Sec. D; detailed training procedures in Sec. E, explanations of the adaptations applied to camera space EHPS methods during evaluations in Sec. F, the step-by-step comprehensive formulations for trajectory transformations in Sec. G and implementations of camera movements for WHAC-A-Mole in Sec. H.

B Results Visualization

In Fig. 1, we visualize the camera and human trajectories of two sequences from EMDB2 dataset. The sequences consist of 2.7k frames for and 1.1k frames respectively. Notably, WHAC demonstrates its capability to accurately recover trajectory and scale in the world space, even for lengthy sequences. Moreover, in Fig. 1b1) and Fig. 1b2), WHAC effectively captures the downward trajectory as the depicted human descends stairs. In Fig. 2, we visualize the human pose and human trajectory in hard cases from EMDB2 [3] and EgoBody [11] including rare walking poses, floating poses, pivoting on one foot, climbing stairs and truncation.

Besides human and camera trajectory estimation in the world space, we present the visualization of the camera space results in Fig. 3. This visualization includes various scenarios such as severe occlusions, close interactions, body contact between subjects, and challenging dancing poses, which serve as representative cases for the WHAC-A-Mole dataset. Even without specific training or finetuning on WHAC-A-Mole dataset, WHAC demonstrates strong abilities in handling pose estimation and depth recovery in camera space for various scenarios. However, challenges persist in the recovery of multi-human interactions and contact between body parts in the world space.

[⋆] Equal contributions.



a1)

a2)

b1)

b2)

Fig. 1: Visualization of world space results on the EMDB dataset. a1) and b1) depict camera trajectories, while a2) and b2) illustrate human trajectories. Notably, in sequence b, the human is descending stairs, and WHAC effectively captures the global trajectory, indicating a downward direction besides recovering the absolute trajectory scale in the world space. The grid size in the plots is 2m.

C MotionVelocimeter

We present the architecture of MotionVelocimeter in Fig. 4. The inputs to MotionVelocimeter consist of canonicalized 3D joints, which are derived from SMPL-X meshes and positioned within the canonical space of the sequence's initial frame. The model's outputs are root velocities corresponding to the previous frame within the canonical space. In contrast to the motion encoder and trajectory decoder that takes 2D keypoints as input in WHAM [7], 3D joints retain spatial information that is critical to velocity estimation in the world-frame for absolute scale recovery. Utilizing 3D joints enhances the model's ability to capture and interpret complex movement patterns, offering a more comprehensive representation of spatial and temporal dynamics within the world-grounded environment.

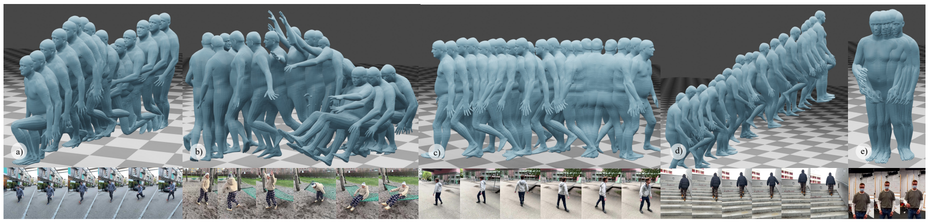


Fig. 2: Hard cases. a) Rare walking poses; b) Floating poses; c) Pivoting on one foot; d) Climbing stairs; e) Truncation.

Table 1: Inference speed of core modules in frame per second (FPS). *Denotes the inference speed without the replaceable, off-the-shelf modules.

Method	GLAMR [10]	SLAHMR [9]	PACE [4]	WHAM [7]	WHAC	WHAC*
FPS	2.4	0.04	2.1	200	165	2500

D Inference speed

As indicated in Table 1, we assess the inference speed of both the core modules (excluding real-time human detection and visual odometry) following the protocol applied in WHAM [7], as well as the inference speed without off-the-shelf modules, which only includes MotionVelocimeter and the scale recovery of human and camera trajectories. As a regression-based method, the inference speed with WHAC has notably faster inference speeds compared to optimization-based methods. Its efficiency enables it to meet the real-time speed requirement.

E Training Details

To enhance temporal consistency and smoothness in camera frame results on temporal datasets, we finetune SMPLer-X-B [1]. This involves incorporating GRUs [2] between the ViT-B backbone and the regression heads of the SMPLer-X-B model. We finetune the model on 4 × V100 GPUs using EgoBody [11], 3DPW [6], and EMDB [3] datasets. We set the minimum learning rate to 1×10^{-6} for 10 epochs, with a sequence length of 32 frames per batch.

For training MotionVelocimeter, we freeze the finetuned SMPLer-X-B model and initialize the learning rate to 1×10^{-3}. We employ a step decay learning rate scheduler, reducing the learning rate by $\gamma = 0.1$ every 2 epochs. The MotionVelocimeter is trained on 4 × V100 GPUs over 8 epochs.

F Adaptations for camera frame EHPS methods

In Table 2 and Table 3 of the main paper, we employ world-grounded adaptations to camera frame methods such as OSX [5] and SMPLer-X [1] using the approach

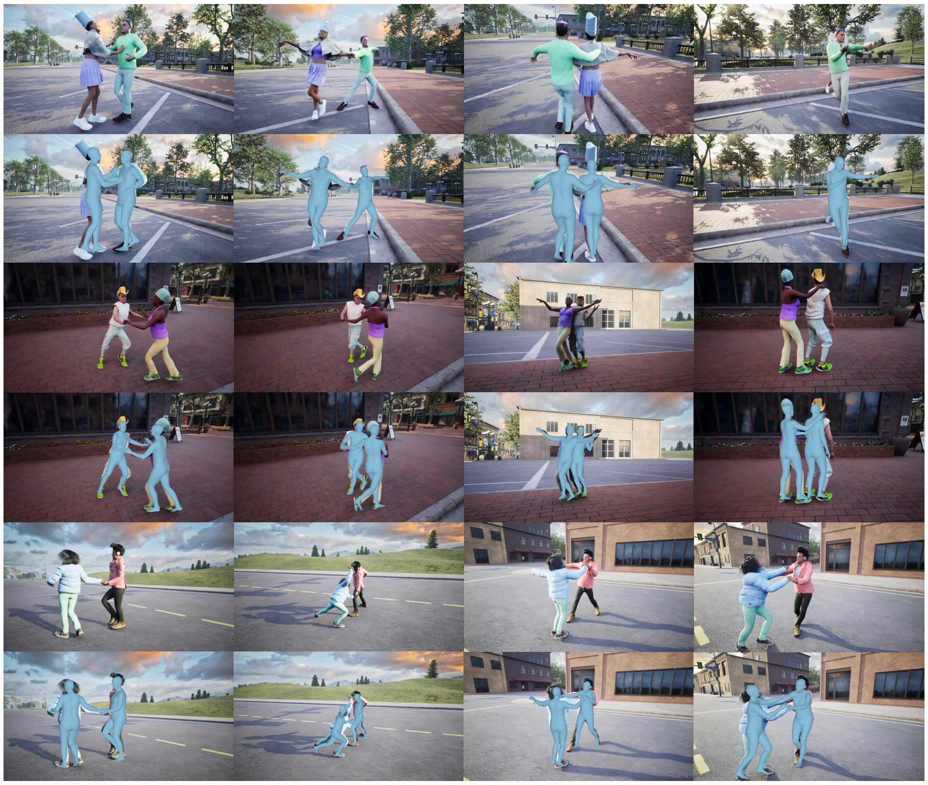


Fig. 3: Visualization of camera space results on WHAC-A-Mole dataset. Each sample comprises two rows: the first row displays the original input frames from the sequence, while the second row overlays the SMPL-X results. This visualization showcases WHAC's performance on challenging scenes, including sequences with severe occlusions, intricate human interactions, and dynamic dancing poses.

described in Sec. 3.2. This method enables the recovery of camera-space root depth, ensuring a fair comparison with other world-grounded methods. Without the adaptations, camera frame methods face limitations when compared to world-grounded methods, due to the high T-MPJPE observed in the camera frame.

G Comprehensive Formulations

Given the space constraints, we present only the finalized and general formulations in Sec. 3.3 and Sec. 3.4 of the main paper. Here, we include the comprehensive step-by-step formulations of the trajectory transformations for clarity.

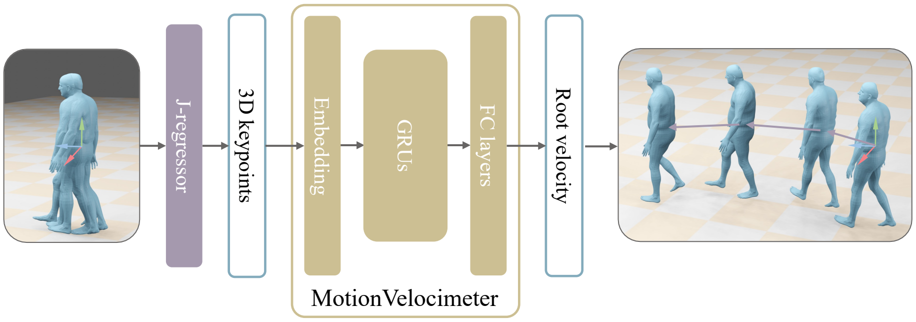


Fig. 4: Illustration of MotionVelocimeter module. The inputs are canonicalized 3D joints regressed from SMPL-X meshes, and the outputs are root velocities in the canonical space.

G.1 Canonicalization

We briefly explain the transformation for canonicalization in Eq. 6 in the main paper. Here we provide the comprehensive formulation:

$$T^{cano} = T^{cano}_{w,i} = [R^{cano}|t^{cano}] = [R^{cano}_{w,i}|t^{cano}_{w,i}], \tag{1}$$

where T^{cano} is a generalized symbol for the transformation from world coordinate system to canonical coordinate system, i denotes the i^{th} frame in the sequence.

For the first (0^{th}) frame in the sequence:

$$R^{cano}_{w,0} = (R^{w}_{c,0} \times \theta^{c}_{go,0})^{-1} = (\theta^{c}_{go,0})^{-1}, t^{cano}_{w,0} = -p^{w}_{0}, \tag{2}$$

where $R^{w}_{c,0}$ is 0^{th} camera rotation in the world frame estimated from visual odometry, $\theta^{c}_{go,0}$ is 0^{th} global orientation estimated in the camera space, $-p^{w}_{0}$ is the pelvis joint of J^{w}_{0} for the first frame.

For every frame in the sequence:

$$R^{cano}_{w,i} = (R^{w}_{c,0} \times \theta^{c}_{go,0})^{-1} = (\theta^{c}_{go,0})^{-1}, t^{cano}_{w,i} = -p^{w}_{i}. \tag{3}$$

We use the pelvis translation $-p^{w}_{i}$ for the corresponding frame while using the camera rotation and global orientation of the first frame for the entire sequence in canonicalization. This is to retain the human rotation in the world coordinate system between frames for reliable trajectory estimation.

G.2 Derive Camera Trajectories from Human Trajectories

In Sec. 3.4 and Eq. 9 in the main paper, we briefly explain the process of deriving camera trajectories $T^{w}_{c,derived}$ from human trajectories T^{w}_{h}. We append the full formulation for this process:

$$T^{w}_{c,derived} = (T^{cano})^{-1} \times T^{cano}_{h} \times (T^{c}_{h})^{-1} = T^{w}_{cano} \times T^{cano}_{h} \times T^{h}_{c} = T^{w}_{h} \times T^{h}_{c}, \tag{4}$$

$$T_h^w = [R_h^w | t_h^w], \tag{5}$$

$$T_c^h = [R_c^h | t_c^h] = [\theta_{go,i}^c | t_{h,i}^c]^{-1}, t_c^h = -(\theta_{go,i}^c)^{-1} \times p_i^c, \tag{6}$$

where T_h^w is the output human trajectories in the world coordinate system from MotionVelocimeter, T_c^h is the inverted human root transformation in the camera coordinate system, $\theta_{go,i}^c$ and p_i^c is the i^{th} global orientation and pelvis in camera space respectively.

To further process the camera rotations $R_{c,derived}^w$ and camera moving trajectories $t_{c,derived}^w$ separately, we re-write the transformations mentioned above in the form of $T = [R|t]$:

$$T_{c,derived}^w = [R_{c,derived}^w | t_{c,derived}^w] = [R_h^w | t_h^w] \times [R_c^h | t_c^h] = [R_h^w R_c^h | R_h^w t_c^h + t_h^w], \tag{7}$$

$$R_{c,derived}^w = R_h^w \times R_c^h, t_{c,derived}^w = R_h^w \times t_c^h + t_h^w, \tag{8}$$

where $t_{c,derived}^w$ is the human-derived camera trajectory in the world coordinate system with absolute scale. The scale recovery is explained in Eq. 10 in the main paper.

G.3 Derive Human Trajectories from Camera Trajectories

In Sec. 3.4 and Eq. 12, we explain the process of deriving the human trajectories $T_{h,final}^w$ from scale-recovered camera trajectories $T_{c,final}^w$:

$$T_{c,final}^w = [R_c^w | t_{c,final}^w], \tag{9}$$

$$T_{h,final}^w = [R_{h,final}^w | t_{h,final}^w] = T_{c,final}^w \times T_h^c, \tag{10}$$

where $T_{c,final}^w$ is the scale-recovered VO-estimated camera trajectories. We empirically find that R_c^w estimated with visual odometry is accurate and can be used in the final camera trajectory $T_{c,final}^w$. The scale recovery process via Umeyama alignment [8] for $t_{c,final}^w$ is explained in Eq. 11 in the main paper. T_h^c is the human root transformation in the camera coordinate system.

By first deriving camera trajectories from MotionVelocimeter-estimated human trajectories, and followed by deriving human trajectories from scaled VO-estimated camera trajectories, we obtain human trajectory $T_{h,final}^w$ and camera trajectory $T_{c,final}^w$, both in the world coordinate system and with absolute scales.

H Implementations of camera movements

In Sec. 4.2, we briefly introduce the camera movement model and types of shots used in WHAC-A-Mole, the implementation details are explained below.

H.1 Arc shot

Arc shot adds equally-spaced keyframes to rotate the camera around the character horizontally by increasing or decreasing $\phi_c \in [\phi_{min}, \phi_{max}]$ or vertically by increasing or decreasing $\theta_c \in [\theta_{min}, \theta_{max}]$. Moreover, the angular velocity of the *arc shots* can be controlled by adjusting the $\varDelta\phi_c$ or the $\varDelta\theta_c$ between two adjacent keyframes.

H.2 Push shot

Push shot also adds equally-spaced keyframes and moves the camera towards the character by decreasing the r_c. More specifically, the fraction of the character in the view, which is more intuitive when filming, is used to indirectly decrease the r_c by

$$r_c = \begin{cases} \frac{h_{bbox}}{frac*\tan(fov/2)} & as \leqslant 1.0 \\ \frac{h_{bbox}*as}{frac*\tan(fov/2)} & as > 1.0 \end{cases}, \tag{11}$$

where the h_{bbox} is the height of the character's bounding box in the camera space, the $frac$ is the desired fraction of the character in the view, the as is the aspect ratio of the camera frame, and the fov is the field of view of the camera. Similar to the *arc shots*, the speed of the camera movement can be adjusted by setting the $\varDelta frac$ between two adjacent keyframes.

H.3 Pull shot

Pull shot is opposite to the *push shot* and moves the camera further away from the character by increasing the r_c under the control of the $frac$. Randomly sampling $frac$ in a range $[frac_{min}, frac_{max}]$ at different keyframes derives continuous pushing and pulling, which is commonly used when filming dances.

H.4 Tracking shot

Tracking shot follows the character and maintains the relative position between the camera and the character, *i.e.*, maintain the (r_c, θ_c, ϕ_c) of the camera in the human-centric spherical coordinate system. A new keyframe of the *tracking shot* is added when the overlap ratio of the character's bounding box in the current frame and in the last keyframe is greater than a threshold $\lambda_{overlap}$.

H.5 Pan shot

Pan shot rotates the camera horizontally to keep the camera looking at the character, therefore it is another way to make the camera follow the character, and it shares the same rule with the *tracking shot* to add a new keyframe.

References

1. Cai, Z., Yin, W., Zeng, A., Wei, C., Sun, Q., Yanjun, W., Pang, H.E., Mei, H., Zhang, M., Zhang, L., Loy, C.C., Yang, L., Liu, Z.: Smpler-x: Scaling up expressive human pose and shape estimation. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 11454–11468. Curran Associates, Inc. (2023)
2. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Kaufmann, M., Song, J., Guo, C., Shen, K., Jiang, T., Tang, C., Zárate, J.J., Hilliges, O.: Emdb: The electromagnetic database of global 3d human pose and shape in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14632–14643 (2023)
4. Kocabas, M., Yuan, Y., Molchanov, P., Guo, Y., Black, M.J., Hilliges, O., Kautz, J., Iqbal, U.: Pace: Human and camera motion estimation from in-the-wild videos. In: Proceedings of the International Conference on 3D Vision. pp. 397–408. IEEE (2024)
5. Lin, J., Zeng, A., Wang, H., Zhang, L., Li, Y.: One-stage 3d whole-body mesh recovery with component aware transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21159–21168 (2023)
6. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European Conference on Computer Vision. pp. 601–617 (2018)
7. Shin, S., Kim, J., Halilaj, E., Black, M.J.: Wham: Reconstructing world-grounded humans with accurate 3d motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2070–2080 (2024)
8. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis & Machine Intelligence **13**(04), 376–380 (1991)
9. Ye, V., Pavlakos, G., Malik, J., Kanazawa, A.: Decoupling human and camera motion from videos in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21222–21232 (2023)
10. Yuan, Y., Iqbal, U., Molchanov, P., Kitani, K., Kautz, J.: Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11038–11049 (2022)
11. Zhang, S., Ma, Q., Zhang, Y., Qian, Z., Kwon, T., Pollefeys, M., Bogo, F., Tang, S.: Egobody: Human body shape and motion of interacting people from head-mounted devices. In: Proceedings of the European Conference on Computer Vision. pp. 180–200. Springer (2022)