Supplementary Material for the paper: "Decoupling Human and Camera Motion from Videos in the Wild"

A. Details of EgoBody evaluation

In Section 4.1 of the main manuscript, we present an experiment on the EgoBody dataset. Here, we provide more details about this evaluation.

We report results on the validation set of EgoBody. Regarding the estimated camera, we use DROID-SLAM [9] with ground truth intrinsics. Regarding the person of interest, we first use PHALP$^+$ [6] (which is the same with out-of-the-box PHALP, but with a more robust detection system [4]), on each sequence. Since there may be multiple people in the frame (but the dataset provides 3D ground truth only for one main person), we then associate the inferred tracklets with the person of interest with the 3D ground-truth pose. For each detected bounding box, we run a 2D keypoint detection network [10]. We run our method and our baselines [3, 11] on the detected tracklets using the same detections (bounding box, 2D keypoints) and ground-truth intrinsics. To accelerate inference, we split the original videos on sequences of 100 frames and we optimize each sequence separately. We report results using both local pose metrics, *i.e.*, PA-MPJPE [2] and global metrics that consider the global estimated trajectory across the whole reconstructed sequence. More specifically, we report results in two settings a) after aligning the predicted sequence with the ground truth sequence using Procrustes (World PA Trajectory - MPJPE), and b) after aligning the first frame of the predicted sequence with the first frame of the ground truth sequence using Procrustes (World PA First - MPJPE).

B. Details of PoseTrack tracking experiment

In Section 4.2 of the main manuscript, we present an ablation where we leverage the estimated camera and the optimized scale α for the purposes of tracking on the PoseTrack dataset [1]. Here, we give more details about this implementation.

To make a direct comparison with PHALP [6], we make minimal modifications to the main algorithm. PHALP uses four cues; appearance, pose, 2D location and nearness of the person. We did not modify the appearance and the pose cues, but only applied the effect of the camera on the location cues, *i.e.*, 2D location and nearness. More specifically, PHALP estimates the 3D location for each person detection in the camera frame, using a single-frame HMR model [2]. Given our estimated camera for each frame (*i.e.*, relative camera from [9] and estimated world scale α from our optimization), we first transform PHALP's 3D location to the world frame (*i.e.*, coordinate frame of the first video frame). Next, PHALP projects these 3D location to the image plane, keeping track of the 2D location, while also recording the depth (nearness) as a separate feature. For simplicity, we take the (X, Y, Z) location of each detection in the world frame and, a) keep the (X, Y) part of the location of each detection to represent the 2D location, (after normalizing it to $[0, 1]$, the same way that PHALP does) and b) use the Z coordinate to compute the nearness. The rest of the pipeline remains the same as PHALP. Essentially, the only difference is that the location of the people are considered in the world coordinate instead of the camera coordinate frame.

We highlight that we only make minimal adaptations to the main PHALP algorithm to demonstrate the effect of camera information for tracking, but there is further room for improvement. For example, considering that we have access to the explicit 3D location for each detection in the world frame, we could also explore tracking using 3D location as a cue, instead of splitting the position cue to 2D location and depth/nearness, but this would require modification to the PHALP's tracking parameters. Similarly, we could leverage our optimized results to compute more reliable affinity metrics on the pose, but here our goal was to decouple the benefit of the better camera from other cues, *i.e.*, our more stable pose. It would be an interesting direction for future work to integrate all these updates and implement a more robust tracking system using information for camera motion.

C. Additional implementation details

Floor specification: When multiple people are on the same floor level, our optimization becomes better constrained because all of them need to share the same floor g, meaning that the motion of more people provides constraints for the optimization of the g variable. However, in many real world videos, people are in different floor levels. In that case, when we observe that it is not possible to solve Equation **??** with a single floor variable g, we separate the people in K

clusters based on the locations of their feet, and introduce K separate floor variables g^k. The people in cluster k shares the same floor g^k and the optimization continues as usual.

Handling multiple people A distinct challenge of in-the-wild videos is properly handling multiple person tracks of undetermined length as they undergo occlusion. During the first two stages of optimization, each person's pose is optimized independently. During these stages, we only optimize the people that are visible, and mask out losses on the predictions of any frame and any track that are not visible.

During the last stage, optimizing all tracks in a single batch allows scale and ground contact information to be shared between people. To do this in our incremental optimization scheme (described in Section 3.4 of the main text), we store each track with respect to its *first appearance*, rather than with respect to the first frame of the video. We pad the end of each track to be T_{max}, the length of the longest track. Specifically, for each track, we store the start and end times of the track, $(t_{\text{start}}, t_{\text{end}})$, and latent vectors $z_{0:T_{\text{max}}}$. The latents of each track are contiguous in time (we infill occlusions between the first and last appearances), but do not all start or end at the same timestep.

In an optimization step at the rollout horizon τ, we roll out 10τ steps of each track $X_{0:10\tau}$, where X is the decoded latent state. We then scatter each track $X_{0:10\tau}$ into the interval $[t_{\text{start}}, \min(t_{\text{end}}, t_{\text{start}} + 10\tau)]$ of input video's timeline. That is, each state X_k synchronized to the original time t it occurred in, and remove the padded states. We then only optimize the track over the time segment containing $X_{0:10\tau}$, $[t_{\text{start}}, \min(t_{\text{end}}, t_{\text{start}} + 10\tau))]$, and mask out the frames of each track that fall outside of this interval.

The runtime of optimization grows linearly with the number of people we track. Optimizing a sequence of around 100 frames and 4 people requires around 40 minutes.

D. Robustness

One of our observations with regards to using the HuMoR motion prior [8] is that it can be challenging to optimize, especially over a long sequence. This results in our decision to optimize the pose sequences of every person in a rollout horizon, as described in the previous section. This increases the robustness of the optimization for longer sequences and it should be applicable to any motion prior that also models the transition, *e.g.*, [5].

Moreover, HuMoR assumes static camera. When used on sequences with camera movement, without modeling the camera motion as we do, it can lead to catastrophic failures in the optimization. For example, in Egobody, we observed that HuMoR fails on 30% of the sequences when we use identity (static) camera. In contrast to that, our approach, even with imperfect camera motion, *i.e.*, using the estimates from [9] as we do, leads to successful optimization in 99% of the sequences; for the rare cases where optimization of the HuMoR motion prior fails, we simply revert back to the results of the previous step where we optimize with the smoothness motion prior.

On the more challenging PoseTrack sequences, we also observe some rare optimization failures. Most of those are related to the single floor assumption and can be addressed by clustering the people in different floors, as described in Section 3.4 of the main manuscript.

E. Limitations

One of the limitation of our approach is that we rely on outputs from other methods (*e.g.*, estimated camera from [9] with approximate intrinsics for in-the-wild videos, person tracking from [6]), which sometime can propagate failures to our optimization.

For example, SfM approaches often have trouble distinguishing between translational and rotational motions, particularly with large focal length. Although our optimization can typically infer reasonable motions even with these imperfect camera estimation, an exciting future work is to jointly optimize the camera motion and human motion, which requires also updating the 3D structure.

Another failure mode is in case of identity switch errors in tracking, with the most harmful being errors that merge two different people into a single tracklet. Although we do not explicitly reason about tracklet identity during our optimization, we provide an experiment where PHALP makes better use of information about camera motion (main manuscript, Section 4.2). Future work could also solve the association problem while optimizing over people and camera's motion.

Finally, we observed some inherently challenging motions to decouple from a monocular video, *e.g.*, when people move co-linearly with the camera. In these cases, our approach can underestimate the location evolution of the people, *e.g.*, causing people to run in the same location. Please see the example in the supplemental video. In these situations, future work could consider also priors for the background scale, *e.g.*, by using monocular depth cues [7], which could help to better constrain the scale factor α.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018. 1

[2] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1

[3] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 1

[4] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 1

[5] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020. 2

[6] Jathushan Rajasegaran, Georgios Pavlakos, Angjoo Kanazawa, and Jitendra Malik. Tracking people by predicting 3D appearance, location and pose. In *CVPR*, 2022. 1, 2

[7] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*, 2020. 2

[8] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. HuMoR: 3D human motion model for robust pose estimation. In *ICCV*, 2021. 2

[9] Zachary Teed and Jia Deng. DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras. *NeurIPS*, 2021. 1, 2

[10] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *NeurIPS*, 2022. 1

[11] Ye Yuan, Umar Iqbal, Pavlo Molchanov, Kris Kitani, and Jan Kautz. GLAMR: Global occlusion-aware human mesh recovery with dynamic cameras. In *CVPR*, 2022. 1