# FRAME: Floor-aligned Representation for Avatar Motion from Egocentric Video

Andrea Boscolo Camiletto[1,2] Jian Wang[1,2] Eduardo Alvarado[1] Rishabh Dabral[1,2]
Thabo Beeler[3] Marc Habermann[1,2] Christian Theobalt[1,2]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus
[2]Saarbrücken Research Center for Visual Computing, Interaction and AI [3]Google, Switzerland

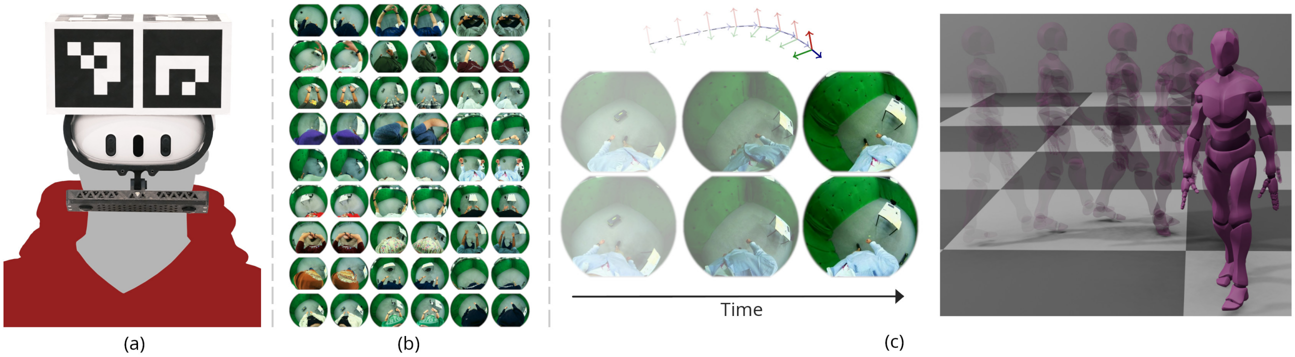


Figure 1. We introduce a large scale egocentric dataset (b) collected with a custom-made wearable capture rig (a). With this data we train FRAME, which takes as input a series of egocentric views and the VR pose tracking and predicts the skeletal motion of the user (c).

## Abstract

*Egocentric motion capture with a head-mounted body-facing stereo camera is crucial for VR and AR applications but presents significant challenges such as heavy occlusions and limited annotated real-world data. Existing methods rely on synthetic pretraining and struggle to generate smooth and accurate predictions in real-world settings, particularly for lower limbs. Our work addresses these limitations by introducing a lightweight VR-based data collection setup with on-board, real-time 6D pose tracking. Using this setup, we collected the most extensive real-world dataset for ego-facing ego-mounted cameras to date in size and motion variability. Effectively integrating this multi-modal input – device pose and camera feeds – is challenging due to the differing characteristics of each data source. To address this, we propose FRAME, a simple yet effective architecture that combines device pose and camera feeds for state-of-the-art body pose prediction through geometrically sound multimodal integration and can run at 300 FPS on modern hardware. Lastly, we showcase a novel training strategy to enhance the model's generalization capabilities. Our approach exploits the problem's geometric properties, yielding high-quality motion capture free from common artifacts in prior work. Qualitative and quantitative evaluations, along with extensive comparisons, demonstrate the effectiveness of our method. Data, code, and CAD designs will be available at vcai.mpi-inf.mpg.de/projects/FRAME.*

## 1. Introduction

Egocentric motion capture poses unique challenges, as predicting body pose without an external viewpoint introduces ambiguities and limits contextual information. Nonetheless, the demand for egocentric motion capture spans numerous applications, including virtual reality (VR), augmented reality (AR), remote collaboration, and robotic control.

In VR, body pose estimation traditionally relies on inverse kinematics [33], using only head and hand poses from the headset and controllers. However, recent trends in industry and research reflect a push towards more accurate body tracking by leveraging priors on human body motion [3, 8, 18, 19, 43], using existing forward-facing cameras [12, 24, 30, 50, 52], or incorporating additional sensing modalities, such as dedicated body-facing cameras [2, 15, 32, 41], multiple inertial measurement units (IMU) [14, 20, 31, 44, 48, 49, 51], direct time-of-flight sensors [34] and pressure-sensing insoles [45].

Among wearable alternatives, capturing a user motion via an ego-facing head-mounted camera offers compelling advantages: it provides a top-down view of the user's body and surroundings, and captures detailed cues – essential for realistic 3D avatars – including clothing and wrinkles [6].

For these reasons, many works have tackled this problem, both in monocular [36, 39–42, 46] and stereo [1, 2, 7, 21, 22, 32, 47, 53] settings, although they assume impractical sensor configurations, such as fisheye cameras mounted on protruding bases (see Fig. 2), which are not feasible for actual consumer devices. In this setting, recent works have introduced large-scale synthetic datasets [1, 2, 36, 41, 46], motion priors [41], and additional sensing modalities [38]. While these developments have significantly improved model capabilities, common limitations persist: poor generalization to in-the-wild data, temporal inconsistencies, and artifacts like body-floor penetration, and foot skating.

These limitations arise from two primary challenges: significant occlusions in the ego-view and a scarcity of real-world training data. While synthetic datasets have been instrumental, the heavy reliance on them has led to difficulties in generalizing to real-world scenarios as the domain gap is hard to bridge. The limited size of real-world datasets further constrains generalization capabilities of these models.

In this paper, we address these limitations at the very core. We introduce a large-scale dataset that is $\times 6$ bigger than what is currently available in the field, eliminating the need for synthetic data pretraining and paving the way for in-the-wild generalization. This dataset is captured using a camera positioning that closely reflects a real VR scenario, offering insights into achievable body tracking in a controller-less setup. An overview of the dataset and the recording rig can be seen in Fig. 1.

Although lightweight SLAM algorithms [4] are now widely integrated into everyday devices – such as VR headsets, robotic vacuum cleaners, and drones – current egocentric motion capture methods rely on general techniques and often overlook unique setup characteristics such as the known relative pose of the two ego-facing cameras and the device pose provided by these tracking pipelines.

To address this, we design a real-time model that can take advantage of both the camera feeds and the device 6D pose, and instead of implicitly learning their relationship with the user pose, explicitly leverages the specifics of the egocentric setup in a differentiable way. This model achieves state-of-the-art accuracy while running at 300 FPS on modern consumer hardware. We also introduce a training strategy that effectively combats overfitting and significantly enhances model generalization to unseen data.
In summary, this paper presents the following contributions:

- A lightweight sensing setup with stereo ego-mounted cameras and head tracking using on-device computations.
- An egocentric benchmark dataset with significantly greater scale and motion diversity than existing datasets.
- An egocentric motion capture architecture that explicitly leverages the setup geometry, improving MPJPE by 28% over state of the art and enabling high frame rates.
- A training strategy that substantially enhances model generalization to unseen data.

## 2. Related Work

**Motion Capture using a Single Egocentric Camera.** One line of work focuses on using a single fisheye camera. xR-EgoPose [36], for instance, utilizes a dual-branch autoencoder to estimate 3D poses from 2D heatmaps. Self-pose [37] extended this approach by incorporating a joint rotation loss and refining the backbone model. $\text{Mo}^2\text{Cap}^2$ adopted a similar method, training a model to unproject 2D predictions into 3D space. Recently, Wang *et al.* [42] and EgoWholeBody [41] improved the techniques by moving predictions into a 3D volumetric space and a pixel-aligned 3D space, respectively. Although single-camera setups enable valuable 3D pose estimations, they lack sufficient context for accurate depth estimation, making accurate spatial capture inherently challenging. Therefore, stereo setups are often preferred as they provide enhanced depth information and improve spatial accuracy.

**Motion Capture using an Egocentric Stereo Camera.** EgoGlass [53] employed a UNet to predict 2D heatmaps, followed by an autoencoder to lift the stereo heatmaps into 3D space. UnrealEgo [1] took a similar approach but utilized a multi-branch autoencoder for 3D pose estimation. UnrealEgo2 [2] expanded on this by adding segmentation masks, depth prediction, structure-from-motion, and temporal refinement modules. More recent works leverage the kinematic structure of joints, such as EgoTAP [21], which uses a propagation network, and Ego3DPose [22], which compensates for limb size disparities by considering their angle relative to the camera. EgoPoseFormer [47] introduced a Pose Refinement Transformer that refines 3D estimates attending visual features in the image space.

Existing methods estimate 3D poses in either the camera coordinate system or relative to the pelvis. Camera-frame estimation benefits from known camera model parameters but lacks broader context, such as gravity-aligned axes. On the other hand, a pelvis-relative estimation cannot leverage the camera's intrinsic prior and needs an estimate of the pelvis position to be aligned with a global frame. Our approach improves on both methods by first predicting in the camera frame and then rototranslating to a floor-aligned reference frame for further refinement. This approach establishes a stable, environment-aligned reference frame that enhances lower-body accuracy and yields more realistic motion capture results.

**Datasets for Egocentric Motion Capture.** Recent years have seen the release of large-scale datasets in fields such as 3D human pose estimation [9, 16] and egocentric action recognition [10, 11], enabling significant advancements. However, in egocentric motion capture, the availability of large-scale datasets remains limited. Collecting large-scale, high-quality datasets for this task is challenging due to the need for specialized devices for markerless motion tracking and complex setups for camera movement tracking.

As a result, most works in this domain rely on synthetic datasets, which provide controlled conditions and readily available ground truth. Mo$^2$Cap$^2$ [46] introduced a dataset of 530k images rendered from two cameras, while xR-egopose [36] produced 383k images from a single camera. UnrealEgo [1] improved upon this with higher-quality assets and a more advanced rendering pipeline, generating 900k images from two cameras at 25 fps. More recently, UnrealEgo2 [2] expanded on this with more complex environments and increased the dataset size to 2.5M images. While synthetic datasets facilitate large-scale data generation with accurate labels, they suffer from domain gaps. The differences in appearance and motion characteristics between synthetic and real-world data often lead to struggles in generalization when models trained on synthetic data are applied to real-world scenarios.

Real-world data collections [2, 32, 36, 46, 53] remain limited, restricting generalization potential and diminishing benchmark reliability due to small dataset sizes. Among the largest, UnrealEgo-RW collected 260k frames matching the camera position of their synthetic dataset, while EgoGlass collected 170k frames. Both total under 2 hours of data.

Another recurring challenge of dataset collection is the tracking of the camera poses. Some previous works lack camera tracking, limiting evaluations to alignment-based metrics like Procrustes analysis [40]. When pose tracking is available, it often requires a cumbersome head-mounted checkerboard [2, 42, 46], which restricts user movement and limits recording duration, leading to unnatural and constrained motion. Moreover, although a checkerboard setup can provide accurate ground-truth device poses, it is unsuitable as input for any method because it does not represent realistic usage scenarios. Consequently, some approaches rely on computationally intensive techniques, such as SLAM coupled with segmentation pipelines to mask out the human body and estimate camera motion [2, 39], resulting in unrealistic head pose tracking incompatible with real-time applications that still lack important details like the ground level.

In contrast, our approach tracks the device pose both from external cameras using a lightweight 3D ArUco board and from onboard sensing directly from a VR device, providing realistic input for our model while also being able to align ground truth labels. This allows us to leverage real-time camera pose information, significantly simplifying the process and improving the reliability of our predictions.

# 3. SELF Dataset

To address the lack of real-world data in egocentric motion capture, we introduce a comprehensive dataset focused on body-facing stereo camera setups. Our dataset captures authentic, diverse, and challenging human movements at unprecedented scale, incorporating environmental interactions absent from prior synthetic or real-world collections.

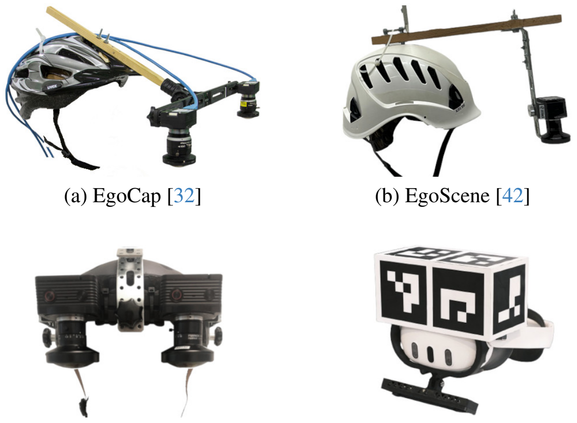

(a) EgoCap [32]
(b) EgoScene [42]
(c) UnrealEgo2 RW [2]
(d) Ours

Figure 2. Comparison of collection devices. With (a), (b), (c) an additional checkerboard must be mounted on top to align ground truth labels, making the helmet significantly top-heavy. Our device can be used for longer period of times and has a camera positioning that mimicks the most a realistic VR scenario.

## 3.1. Capture System and Recording Rig

We captured the dataset in a recording studio equipped with 120 synchronized and calibrated 4K RGB cameras, tracking the skeletal motion with markerless motion capture [5].

As existing VR devices lack built-in egocentric cameras or restrict access to their video feeds, we designed a rig centered on the Meta Quest 3 [29]. Unlike previous setups that relied on helmets outfitted with unrealistically protruding cameras and large checkerboards, our design is lightweight and includes a downward-facing stereo fisheye camera that closely match a realistic VR scenario (see also Fig. 2)

Our rig captures two video streams at 640×480 resolution and 30Hz. The 6D head pose is computed on-device using Quest internal SLAM algorithm. A 3D-printed ArUco board with six markers is mounted on the device solely during dataset collection to allow for ground truth alignment in the VR frame of reference, but is not used during inference.

## 3.2. Alignment and Synchronization

To ensure spatial alignment and temporal synchronization across studio cameras, egocentric cameras, and VR onboard pose tracking, we employ a calibration process.

**Calibration and Synchronization.** Each session starts with a one-minute calibration sequence in which participants perform basic motions to establish markerless capture rigging and to calculate transformations between the cameras, markers, and headset using an external checkerboard visible to both the headset and studio cameras. For synchronization, we toggle lights on and off to align the 30Hz fisheye camera clocks with the studio cameras.

**Headset Tracking Alignment.** For each external camera

$i$, we detect any visible ArUco marker $n$ as $\mathbf{d}^i_{n,t} \in \mathbb{R}^{4\times 2}$ at time $t$. With known marker dimensions and placement, and using the camera extrinsics $\mathbf{P}_i \in \mathbb{R}^{3\times 4}$ and intrinsics $\mathbf{K}_i \in \mathbb{R}^{3\times 3}$, we estimate the 6D pose of the ArUco board by solving a Perspective-n-Point problem for each camera, averaging contributions across all cameras.

To refine this estimate, we solve a global optimization problem for each frame, minimizing the sum of reprojection errors across all cameras:

$$\arg\min_{\mathbf{N}_t \in \mathrm{SE}(3)} \sum_{i=1}^{C} \sum_{n=1}^{6} \left\| \mathbf{d}^i_{n,t} - \pi\left(\mathbf{K}_i \mathbf{P}_i \mathbf{N}_t \mathbf{X}^{\text{hom}}_n\right) \right\|^2 \forall t, \qquad (1)$$

where $\mathbf{N}_t$ represents the ArUco board pose at time $t$, $C$ is the number of external cameras, $\mathbf{X}^{\text{hom}}_n$ is the known 3D position of marker $n$ in local homogeneous coordinates, and $\pi(\cdot)$ denotes the perspective projection function.

**On-Device Pose Alignment.** The VR on-device tracking provides a 6D pose independently of the studio cameras, yet in a different coordinate system and on a separate clock. To align this data with the studio reference, we note that, under ideal conditions, the VR-estimated pose should match the computed 3D ArUco Board pose once we account for the time offset, coordinate system differences, and fixed transformation between the headset and the ArUco board. Hence, we solve the following minimization problem:

$$\arg\min_{\mathbf{T}_c, \mathbf{T}_r, t_0} \sum_{t}^{T} \left\| \mathbf{N}_t - \mathbf{T}_c \mathbf{V}_{t+t_0} \mathbf{T}_r \right\|^2, \qquad (2)$$

where $\mathbf{V}_{t+t_0}$ is the VR recorded device pose at internal clock $t$, $t_0$ represents the clock offset, $\mathbf{T}_c$ is the transformation mapping the VR coordinate system to the studio frame, and $\mathbf{T}_r$ is the fixed transformation between the ArUco board and VR frames. The summation happens over the time dimension. We solve this problem with a two-stage optimization: Levenberg-Marquardt [23] for transformations and iterative grid search for the clock offset. The result of this optimization allows us to obtain the device tracked pose in the same frame of reference with our ground truth labels.

### 3.3. Dataset Structure

Our data collection involved 14 participants, each completing two sessions with distinct clothing to enhance visual diversity. Sessions begin with a one-minute calibration, followed by 50 predefined actions (20 seconds each, with 4-second pauses) spanning a variety of sports and daily activities to ensure diverse motions. Participants can see and interact with their surroundings thanks to the forward-facing cameras on the VR device.

Each session captures stereo egocentric video, 6D device pose, body poses via markerless motion capture, and feeds from a 120-camera studio setup, resulting in over 7 hours and approximately 1.6 million individual images. A separate test set, recorded with two additional participants, offers about an hour of data for unbiased evaluation and model generalization testing. Notably, our dataset is significantly larger than competing real-world datasets. A detailed comparison is provided in Tab. 1.

| Dataset | Cams | Hrs | Frames | VR Trk | Act |
|---|---|---|---|---|---|
| EgoCap [32] | 2 | 0.7 | 75k | ✗ | 8 |
| Mo2Cap2 [46] | 1 | NA | 5k | ✗ | 3 |
| xR-EP [36] | 1 | NA | 10k | ✗ | NA |
| EgoGlobal [39] | 1 | NA | 12k | ✗ | 2 |
| SceneEgo [42] | 1 | NA | 92k | ✗ | 5 |
| EgoGlass [53] | 2 | 1.6 | 173k | ✗ | 10 |
| UE-RW [2] | 2 | 1.4 | 260k | ✗ | 16 |
| **Ours** | 2 | **7.4** | **1.6M** | ✓ | 14 |

Table 1. Comparison across available egocentric human datasets that provide ground truth annotations. *Cams* stands for Egocentric Cameras, *Hrs* for Hours of video, *VR Trk* for on-device head tracking, *Act* for Actors. Note that our dataset by far outperforms prior datasets in terms of scale while also providing ground truth VR tracking and stereo camera images.

In summary, each time-frame $t$ includes two body-facing fisheye images $\mathbf{I}_t \in \mathbb{R}^{2\times 3\times H\times W}$, the onboard device pose $\mathbf{T}_{\mathrm{D}}(t) \in SE(3)$, and ground truth joint positions $\mathbf{J}_t \in \mathbb{R}^{J\times 3}$. As is the case with VR, device poses $\mathbf{T}_{\mathrm{D}}(t)$ are floor-aligned by design – a critical factor that underpins the predictive methods presented in the following section.

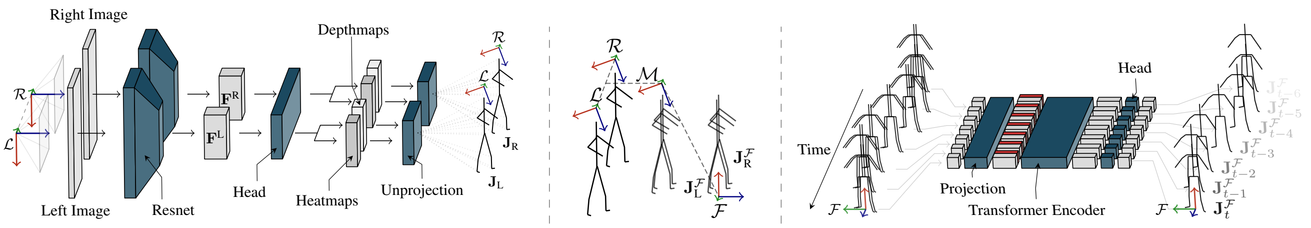
## 4. Method

To showcase the effectiveness of our dataset, we propose a method to estimate human pose $\mathbf{J}_t$ at time $t$ from an image stream, $\{\ldots, \mathbf{I}_{t-1}, \mathbf{I}_t\}$, captured from the head-mounted downward-facing cameras, along with the onboard device poses $\mathbf{T}_{\mathrm{D}}(t)$.

Our approach explicitly leverages this multimodal input, embedding global information in the model during runtime. First, we utilize the fisheye camera's intrinsic calibrations to make predictions in their local coordinate spaces for the current frame (Sec. 4.1). Next, we align the current and previous predictions to a common floor and gravity aligned coordinate system using the known relative poses of the cameras and onboard device tracking (Sec. 4.2). This alignment is crucial, as it enables the model to address challenges such as foot penetration and lower-limb inconsistencies while capturing dynamic motions over time, informed by prior predictions (Sec. 4.3). Our design decisions are motivated by real-time, on-device considerations, enabling the model to run at 300 FPS on an NVIDIA 3090 GPU.

### 4.1. Fisheye-based Pose Estimation

Building on prior methods on human pose estimation [17, 27, 28, 35], we train a network to predict a 2.5D repre-


Right Image
$\mathcal{R}$
$\mathcal{L}$
Left Image
Resnet
$\mathbf{F}^{\mathrm{R}}$
$\mathbf{F}^{\mathrm{L}}$
Head
Depthmaps
Heatmaps
Unprojection
$\mathcal{R}$
$\mathcal{L}$
$\mathbf{J}_{\mathrm{R}}$
$\mathbf{J}_{\mathrm{L}}$
$\mathcal{R}$
$\mathcal{M}$
$\mathcal{L}$
$\mathbf{J}_{\mathrm{R}}^{\mathcal{F}}$
$\mathbf{J}_{\mathrm{L}}^{\mathcal{F}}$
$\mathcal{F}$
Time
$\mathcal{F}$
Head
Projection
Transformer Encoder
$\mathcal{F}$
$\mathbf{J}_{t-6}^{\mathcal{F}}$
$\mathbf{J}_{t-5}^{\mathcal{F}}$
$\mathbf{J}_{t-4}^{\mathcal{F}}$
$\mathbf{J}_{t-3}^{\mathcal{F}}$
$\mathbf{J}_{t-2}^{\mathcal{F}}$
$\mathbf{J}_{t-1}^{\mathcal{F}}$
$\mathbf{J}_{t}^{\mathcal{F}}$

(a) (b) (c)

Figure 3. Overview of the proposed architecture. In Fig. 3a, the backbone accepts two images from the stereo camera as input and outputs two poses, one for each frame. As shown in Fig. 3b these poses are rototranslated in the frame of reference $\mathcal{F}$. In Fig. 3c our STF model accepts the history of two poses from the backbone, aligned in the most recent $\mathcal{F}$, and merges the result into a single sequence.

sentation of keypoints – comprising 2D image coordinates with their corresponding depth – and unproject them into 3D space according to our fisheye camera model. Specifically, given a pair of stereo input images $\mathbf{I}$, we use a pretrained ResNet50 [13] to extract visual features $\mathbf{F}^{\mathrm{L}}, \mathbf{F}^{\mathrm{R}} \in \mathbb{R}^{C\times \frac{H}{16} \times \frac{W}{16}}$ for each view, where $H$ and $W$ denote the height and the width of the input image. To lower the computational load, we omit the final residual block in the ResNet, obtaining as a side effect an increased spatial resolution that benefits model accuracy.

The features from both cameras are then concatenated and fed to our convolution-based head that predicts, for each camera view and each joint $i$, heatmaps and depthmaps $\mathbf{H}_i, \mathbf{D}_i \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16}}$. The heatmaps are then normalized into 2D distributions using a softmax operation, $\hat{\mathbf{H}}_i = \mathrm{softmax}(\beta_i \cdot \mathbf{H}_i)$, where $\beta_i$ is a learnable temperature parameter.
To obtain the $u_i, v_i$ pixel coordinates, we apply a soft-argmax operation [26]. We then calculate the depth $d_i$ as

$$d_i = \sum_{j,k} (\mathbf{D}_i \odot \hat{\mathbf{H}}_i)_{jk}, \tag{3}$$

where $\odot$ denotes the Hadamard product. More details on the camera model and the unprojection process are provided in the supplementary material.

With $u_i, v_i, d_i$ available for both the views, we unproject each predicted joint to 3D using the fisheye camera model, yielding pose predictions $\mathbf{J}_{\mathrm{L}}$ and $\mathbf{J}_{\mathrm{R}}$, where the subscript (L or R) denotes the camera of origin. An overview of this process can be seen in Fig. 3a.

## 4.2. Coordinate Systems Alignment

Previous methods generally could not assume device pose availability due to hardware constraints, limiting their ability to address the inherent multi-modality of the problem. In contrast, leveraging the onboard headset pose, we employ a principled approach: rather than learning the relationship between skeletal motion and the device pose, we use it explicitly to move the problem to a shared, floor-aligned frame of reference. Given the headset pose $\mathbf{T}_{\mathrm{D}}$ at any point in time and the fixed relative transformations $\mathbf{M}_{\mathrm{L}}, \mathbf{M}_{\mathrm{R}} \in SE(3)$ between the headset and each camera, we can compute the global pose of each camera $\mathbf{T}_{\mathcal{L}}, \mathbf{T}_{\mathcal{R}}$ as follows:

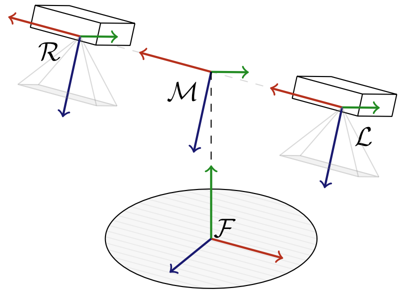


Figure 4. $\mathcal{L}$ and $\mathcal{R}$ are the left/right camera frames, $\mathcal{M}$ is the middle frame computed as the average of the two camera frames. The $x, y, z$ axes are color-coded to red, green, and blue, respectively. $\mathcal{F}$ is obtained by moving the origin of $\mathcal{M}$ at the ground level, aligning $y$ axis to be vertical, and using the projection of the $x$ axis of $\mathcal{M}$ on the horizontal plane to determine the direction of the horizontal axes of $\mathcal{F}$

$$\mathbf{T}_{\mathcal{L}} = \mathbf{T}_{\mathrm{D}} \cdot \mathbf{M}_{\mathrm{L}}, \quad \mathbf{T}_{\mathcal{R}} = \mathbf{T}_{\mathrm{D}} \cdot \mathbf{M}_{\mathrm{R}},$$

Although a global frame provides useful information, such as gravity direction and floor positioning, it remains suboptimal as both its origin and horizontal axes are arbitrary. To account for this, we define a new reference frame $\mathcal{F}$ that is aligned with the floor, as described in Fig. 4.

This allows us to compute the relative transformations, ${}^{\mathcal{F}}\mathbf{T}_{\mathcal{L}}$ and ${}^{\mathcal{F}}\mathbf{T}_{\mathcal{R}}$, of the two cameras to the floor's frame of reference and use them to align each camera's predicted 3D joints into the common, floor-aligned frame $\mathcal{F}$

$$\mathbf{J}_L^{\mathcal{F}} = {}^{\mathcal{F}}\mathbf{T}_{\mathcal{L}} \cdot \mathbf{J}_L, \quad \mathbf{J}_R^{\mathcal{F}} = {}^{\mathcal{F}}\mathbf{T}_{\mathcal{R}} \cdot \mathbf{J}_R,$$

Note that any device slipping is inherently accounted for, as onboard SLAM continuously tracks the actual headset pose, naturally capturing variations in its position relative to the head.

## 4.3. Stereo Temporal Fusion

To compute the final estimate, $\mathbf{J}_t$, we align the pose pairs from the current and the past steps to the most recent $\mathcal{F}$ as $\{\dots, \mathbf{J}_{\mathrm{L},t-1}^{\mathcal{F}}, \mathbf{J}_{\mathrm{R},t-1}^{\mathcal{F}}, \mathbf{J}_{\mathrm{L},t}^{\mathcal{F}}, \mathbf{J}_{\mathrm{R},t}^{\mathcal{F}}\}$ , yielding a complete motion sequence in a shared coordinate system. This design provides several advantages. First, by establishing a stable reference aligned with the ground, we can correct artifacts such as floating feet or ground penetration. Additionally, by knowing the direction of gravity, one can compensate for unstable poses or preserve them when consistent with the motion history from previous steps.

Having the estimated per-view, per-frame body poses in the floor's frame of reference, we finally perform a multi-view temporal fusion of the coarse pose estimates. Specifically, we input this sequence of pose pairs into our Stereo Temporal Fusion (STF) module, a transformer encoder tailored for this task. STF is structured with 8 layers, each having a feedforward dimension of 512 and 32 attention heads, and processes the previous 20 predictions sampled at 15Hz. A visualization can be seen in Fig. 3c.

The model outputs refined motion by integrating stereo predictions, floor alignment, and temporal coherence. It is worth noting that while it predicts the full motion sequence during training, our method operates in real-time at inference, outputting only the most recent frame.

### 4.4. Cross-Training Caching

Image-based pose estimation models typically improve generalization as training progresses, although performance on unseen data naturally lags behind improvements on the training set. This discrepancy negatively impacts downstream modules — such as our Stereo Temporal Fusion (STF) — which, when trained on these overly accurate backbone predictions, struggle to generalize to inputs with more realistic error distributions.

To mitigate this, we introduce a Cross-Training Caching approach inspired by k-fold cross-validation, specifically designed to replicate realistic errors during training. We divide our training set into $k$ subsets and iteratively retrain our backbone on $k-1$ subsets, caching predictions on the held-out subset. After repeating this process $k$ times, we obtain backbone predictions for the entire training set. These cached predictions, reflecting realistic errors on unseen data, are used exclusively to train the STF module, ensuring it learns robustly from inputs that closely mimic inference-time conditions. A visualization of this mechanism is shown in Fig. 5. The final model employs a backbone trained on the entire training set.

## 5. Experiments

**Implementation Details.** All input images are resized to $256 \times 256$, and we utilize the AdamW optimizer [25] with batch size 16 for training. We first train our backbone fisheye-based pose estimation module with a learning rate of $3 \cdot 10^{-5}$ and a weight decay $10^{-5}$ for 6 epochs. We apply an L2 loss on both predictions of the model in camera coordinates. We then follow the strategy explained in Sec. 4.4 to train the STF module, with a learning rate of $3 \cdot 10^{-4}$ and weight decay $10^{-4}$ for 2 epochs with the same loss, but now in global coordinates.

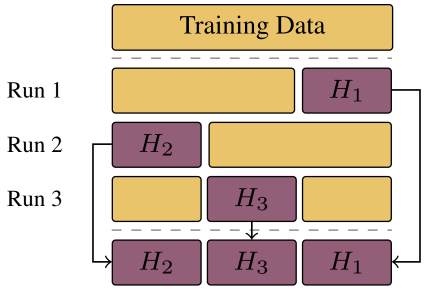


Figure 5. Visualization of the k-fold Cross Training Caching Strategy with $k = 3$. $H_i$ denotes the hold-out data for the $i$-th run.

**Dataset.** From the 14 available subjects in the FRAME dataset, we exclude two (one male and one female) from the training set to assess the generalization capability of each model. Since the cameras are already aligned with the head, we follow best practices in the field [41, 42] and do not align the body predictions with the ground truth when evaluating. We adopt a 15-keypoint skeleton consistent with the EgoScene dataset [42] and adapt the baselines to this skeleton where necessary.

**Metrics.** As in prior works, we report results in terms of mean per joint positional error (MPJPE) in millimeters. In addition, we report other complementary metrics, i.e., Procrustes-aligned MPJPE (PA-MPJPE), 3D percentage of correct keypoints (3D-PCK) within a 10cm threshold, Jitter, Non-Penetration Percentage (NPP), Mean Penetration Error (MPE), and Foot Sliding (FS). A detailed explanation on how these metrics are computed is present in the supplementary materials.

### 5.1. Comparison

To ensure a rigorous comparison with state-of-the-art methods, we retrain three existing methods, i.e. UnrealEgo [1], EgoGlass[53], and EgoPoseFormer[47], on our dataset according to the original settings described in the papers. If a method was initially trained on predicting the root-relative pose, we adjust it to output the pose in the camera frame, as it would require knowing ground truth information (the root position) in order to evaluate its MPJPE. The results in Tab. 2 highlight the effectiveness of our approach across multiple key metrics. Our model design achieves the lowest MPJPE among all the tested methods, while maintaining a lightweight architecture and running at 300 FPS on an NVIDIA RTX 3090, which is significatly faster than prior works. This performance is particularly noteworthy in metrics related to lower-body alignment, where the integration of device pose tracking enables precise floor alignment and prevents common artifacts such as floor penetration, leading to a perfect Non-Penetration Percentage (NPP) score. Further, our method showcases improved stability in sequen-

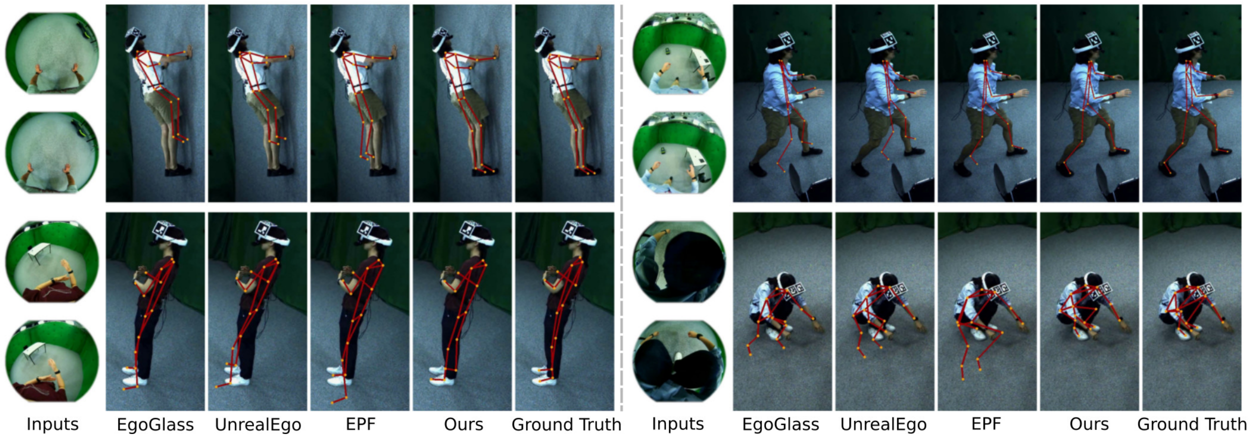


Figure 6. Qualitative comparison on challenging inputs. The predicted 3D poses are overlayed onto external reference views not used for tracking. Our qualitative results confirm that our method predicts more accurate body poses, and significantly better handles contacts with the floor and lower limbs compared to prior state-of-the-art approaches [1, 47, 53].

| Method | Inference (ms, ↓) | MPJPE (mm, ↓) | PA-MPJPE (mm, ↓) | 3D-PCK (%, ↑) | Jitter (mm, ↓) | NPP (%, ↑) | MPE (mm, ↓) | FS (cm/s, ↓) |
|---|---|---|---|---|---|---|---|---|
| Egoglass [53] | 8.97 | 105.56 | 74.11 | 61.38 | 12.60 | 52.11 | 48.16 | 12.34 |
| Unrealego [1] | 6.87 | 104.81 | 68.10 | 61.22 | 11.77 | 58.03 | 48.19 | 10.71 |
| EgoPoseFormer [47] | 14.36 | 69.18 | 41.29 | 78.98 | 9.98 | 49.45 | 47.97 | 9.29 |
| **Ours** | **2.68** | **47.53** | **35.86** | **92.56** | **4.96** | **100.0** | **0.00** | **3.47** |

Table 2. Comparison of different egocentric models. The baselines are retrained on our dataset. Our method is the only one that leverages both the camera feeds and the device tracking. Notably, we outperform prior works by a significant margin across all metrics.

tial predictions, as evidenced by reductions in, both, jitter and foot sliding. These enhancements are largely attributed to the Stereo Temporal Fusion (STF) module, which incorporates prior predictions to yield temporally smoother motions. Comparative analysis with baseline methods reveals the substantial benefits of our combined stereo-temporal fusion and floor alignment strategy and overall highlights the advantages of our multi-modal integration, achieving state-of-the-art performance across all evaluated metrics.

## 5.2. Ablation Study

We systematically evaluate the effect of different configurations on the capture performance. Tab. 3 summarizes our ablation study. Each row reflects a distinct experimental setting, incrementally incorporating model components to isolate their contribution. To better disentangle the impact of our frame alignment and the use of predictions at previous steps, we report some results where only an MLP head is used for merging the stereo views, without any information on the prediction history.

**Baseline (w/o stereo).** When only the left camera view is used without stereo information, the model lacks depth cues provided by binocular views, resulting in the highest MPJPE (87.94 mm).

**Stereo Fusion by Averaging (w/ avg).** Averaging predictions from the two stereo views, now aligned in the middle frame ($\mathcal{M}$), reduces MPJPE to 84.31 mm, due to the partial balancing out of discrepancies in each view prediction.

**Learned Stereo Fusion (w/ MLP).** Introducing an MLP to merge left and right views without any explicit alignment further decreases MPJPE to 82.19mm. This indicates that a learnable fusion can adjust to the different view biases. Explicit alignment before the MLP provides an additional improvement to 81.69mm.

**Shared Head (w/ SH).** Processing image features jointly rather than independently reduces MPJPE to 74.13mm, suggesting that stereo information can significantly help the model to resolve ambiguities in monocular views.

**Learnable SoftArgmax (w/ LSA).** Using the learnable version of softargmax decreases MPJPE to 71.31mm, hinting that joint-specific temperature can contribute to additional accuracy.

**Frame Alignment.** One of our key contributions, i.e. leveraging the device pose by rototranslating predictions into frame $\mathcal{F}$, significantly decreases the MPJPE to 59.53mm, showing the impact the frame of reference can have. As shown in Fig. 7, the most significant improvement happens in the lower limbs.

**Cross Training Caching (w/ CT).** Introducing our Cross Training (CT) brings it down to 54.08mm suggesting that efforts in mimicking unseen data error distribution in the training set increase the ability of the model to generalize.

**Stereo Temporal Fusion (STF).** Incorporating the STF module, which leverages previous frames information, achieves the best performance, reducing MPJPE to 47.53mm. This highlights the importance of temporal in-

| Method | Frame | MPJPE |
|---|---|---|
| w/o stereo | $\mathcal{L}$ | 87.94 |
| w/ avg | $\mathcal{M}$ | 84.31 |
| w/ MLP | $\mathcal{L}+\mathcal{R}$ | 82.19 |
| w/ MLP+avg | $\mathcal{M}$ | 81.69 |
| w/ MLP+avg+SH | $\mathcal{M}$ | 74.13 |
| w/ MLP+avg+SH+LSA | $\mathcal{M}$ | 71.31 |
| w/ MLP+avg+SH+LSA | $\mathcal{F}$ | 59.53 |
| w/ MLP+avg+SH+LSA+CT | $\mathcal{F}$ | 54.08 |
| **Ours** | $\mathcal{F}$ | **47.53** |

Table 3. Ablation results. *w/ avg* denotes stereo merging by averaging predictions; *SH* is the Shared Head; *LSA* is Learnable Soft-Argmax; *DN* stands for Dynamic Noise; *CT* is the Cross Training caching strategy; *Ours* is our Stereo Temporal Fusion module.

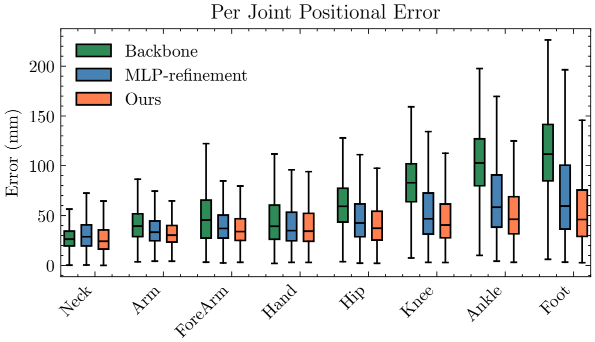


Figure 7. The distribution of errors at different stages. In green, just after the prediction in camera coordinates, where the feet are hard to estimate. In blue, we can observe the improvement provided with by the rototranslation to $\mathcal{F}$ with an MLP without the time component, while in orange we see the compounded effect of both time history and floor alignment.

formation, as the model can use past predictions to correct or refine current estimates.

Our ablations demonstrate that careful integration of stereo-temporal fusion coupled with the ability to work in a more meaningful frame of reference provides substantial gains, showing the value of each component in achieving state-of-the-art accuracy in egocentric pose estimation.

### 5.3. Generalizablity of our Floor-aligned Frame

Our floor-aligned frame is a general design, which can be easily integrated into other egocentric pose estimation methods. To illustrate the effectiveness and generalizability of this design, we select the monocular EgoWholeBody [41] model, training it solely on images from the left camera of our dataset. After obtaining initial predictions, we rototranslate it to a different frame before using an MLP to refine it. Tab. 4 demonstrates the impact of each frame choice on MPJPE. The baseline ($\mathcal{L}$) provides a starting point with 85.63mm. Aligning predictions to the middle frame ($\mathcal{M}$) yields a slight improvement, reducing MPJPE to 84.06mm. However, re-projecting the model's output into the floor frame ($\mathcal{F}$), i.e. our proposed design, results in a significant MPJPE reduction to 76.43mm, confirming that this alignment contributes directly to improved accuracy.

| Method | Refinement Frame | MPJPE |
|---|---|---|
| EgoWholeBody [41] | $\mathcal{L}$ | 85.63 |
| EgoWholeBody [41] | $\mathcal{M}$ | 84.06 |
| EgoWholeBody [41] | $\mathcal{F}$ | 76.43 |

Table 4. Impact of the frame of reference in refinement on a monocular method trained exclusively on the left camera. Note that our proposed floor-aligned frame can signficantly improve other egocentric approaches as it is a generally applicable design.

## 6. Discussion and Conclusion

**Limitations.** Current VR headsets offer rich multimodal data such as environment meshes, forward-facing cameras, eye-gaze tracking, and hand pose estimation from controllers—inputs our current method does not exploit. Leveraging these modalities can further enhance the model's ability to perceive nuanced motion and user intention. Furthermore, while our approach robustly addresses many challenging scenarios, it can struggle with inherent motion capture issues such as significant occlusions and self-contact. Integrating physics-based modeling as a way to bridge the aforementioned multimodal data, represents a promising direction to overcome these limitations by enforcing realistic physical constraints and resolving ambiguous cases.

**Conclusions.** We presented a large-scale, real-world dataset for egocentric motion capture, surpassing existing datasets in size and motion complexity. This dataset includes high-quality, on-device head tracking, providing essential information for pose estimation on any device capable of tracking its 6D pose. Our proposed method leverages known geometric transformations, such as camera and device poses, achieving precise 3D pose predictions while running efficiently at 300FPS on modern hardware. We demonstrated that our approach improves generalization and that our frame of reference choices significantly enhances performance over competing methods. Experiments validate the effectiveness of our model and the architectural decisions underlying its design. Looking ahead, we expect this motion capture model to be extended and serve as a starting point to not only estimate 3D keypoints but also physical quantities or maybe see its application in driving photorealistic avatars from egocentric signals.

**Acknowledgements.** This work was funded by the Saarbrücken Research Center for Visual Computing and Artificial Intelligence (VIA).

## References

[1] Hiroyasu Akada, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik. Unrealego: A new dataset for robust egocentric 3d human motion capture. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 6, 7

[2] Hiroyasu Akada, Jian Wang, Vladislav Golyanik, and Christian Theobalt. 3d human pose perception from egocentric stereo videos. *arXiv preprint arXiv:2401.00889*, 2023. 1, 2, 3, 4

[3] Sadegh Aliakbarian, Fatemeh Saleh, David Collier, Pashmina Cameron, and Darren Cosker. Hmd-nemo: Online 3d avatar motion generation from sparse observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9622–9631, 2023. 1

[4] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 2

[5] Captury. Captury markerless motion capture system. Accessed: 2024-10-09. 3

[6] Jianchun Chen, Jian Wang, Yinda Zhang, Rohit Pandey, Thabo Beeler, Marc Habermann, and Christian Theobalt. Egoavatar: Egocentric view-driven and photorealistic full-body avatars. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24)*, pages 1–11, New York, NY, USA, 2024. ACM. 1

[7] Hanz Cuevas-Velasquez, Charlie Hewitt, Sadegh Aliakbarian, and Tadas Baltrušaitis. Simpleego: Predicting probabilistic body pose from egocentric cameras. In *2024 International Conference on 3D Vision (3DV)*, pages 1446–1455. IEEE, 2024. 1

[8] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023. 1

[9] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa*, and Jitendra Malik*. Humans in 4D: Reconstructing and tracking humans with transformers. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[10] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2

[11] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 2

[12] Vladimir Guzov, Yifeng Jiang, Fangzhou Hong, Gerard Pons-Moll, Richard Newcombe, C Karen Liu, Yuting Ye, and Lingni Ma. Hmd$^2$: Environment-aware motion generation from single egocentric head-mounted device. *arXiv preprint arXiv:2409.13426*, 2024. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[14] 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, 2018. First two authors contributed equally. 1

[15] Apple Inc. Apple vision pro. `https://www.apple.com/apple-vision-pro/`, 2023. Accessed: 2024-09-05. 1

[16] 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, 2014. 2

[17] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 4

[18] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *Proceedings of European Conference on Computer Vision*. Springer, 2022. 1

[19] Jiaxi Jiang, Paul Streli, Manuel Meier, Andreas Fender, and Christian Holz. Egoposer: Robust real-time ego-body pose estimation in large scenes. *arXiv preprint arXiv:2308.06493*, 2023. 1

[20] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W. Winkler, and C. Karen Liu. Transformer inertial poser: Real-time human motion reconstruction from sparse imus with simultaneous terrain generation. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 1

[21] Taeho Kang and Youngki Lee. Attention-propagation network for egocentric heatmap to 3d pose lifting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 842–851, 2024. 1, 2

[22] Taeho Kang, Kyungjin Lee, Jinrui Zhang, and Youngki Lee. Ego3dpose: Capturing 3d cues from binocular egocentric views. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 1, 2

[23] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944. 4

[24] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17142–17151, 2023. 1

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 6

[26] Diogo C Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *Computers & Graphics*, 85:15–22, 2019. 5

[27] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *Acm transactions on graphics (tog)*, 36(4):1–14, 2017. 4

[28] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *Acm Transactions On Graphics (TOG)*, 39(4):82–1, 2020. 4

[29] Meta. Meta quest 3, 2023. Accessed: 2024-10-09. 3

[30] Inc. Meta Platforms. Inside-out body tracking and generative legs. https://developer.oculus.com/blog/inside-out-body-tracking-and-generative-legs/?locale=it_IT, 2023. Accessed: 2024-09-05. 1

[31] Vimal Mollyn, Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. Imuposer: Full-body pose estimation using imus in phones, watches, and earbuds. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2023. Association for Computing Machinery. 1

[32] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. Egocap: Egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 1, 3, 4

[33] RootMotion. Final ik. https://assetstore.unity.com/packages/tools/animation/final-ik-14290, 2024. Unity Asset Store plugin. 1

[34] Denys Rozumnyi, Nadine Bertsch, Othman Sbai, Filippo Arcadu, Yuhua Chen, Artsiom Sanakoyeu, Manoj Kumar, Catherine Herold, and Robin Kips. Xr-mbt: Multi-modal full body tracking for xr through self-supervision with learned depth point cloud registration. *arXiv preprint arXiv:2411.18377*, 2024. 1

[35] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European conference on computer vision (ECCV)*, pages 529–545, 2018. 4

[36] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7728–7738, 2019. 1, 2, 3, 4

[37] Denis Tome, Thiemo Alldieck, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino, and Fernando De la Torre. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6794–6806, 2020. 2

[38] Matt Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *2017 British Machine Vision Conference (BMVC)*, 2017. 2

[39] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, and Christian Theobalt. Estimating egocentric 3d human pose in global space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11500–11509, 2021. 1, 3, 4

[40] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, Diogo Luvizon, and Christian Theobalt. Estimating egocentric 3d human pose in the wild with external weak supervision. *CVPR*, 2022. 3

[41] Jian Wang, Zhe Cao, Diogo Luvizon, Lingjie Liu, Kripasindhu Sarkar, Danhang Tang, Thabo Beeler, and Christian Theobalt. Egocentric whole-body motion capture with egowholebody and diffusion-based motion refinement. *arXiv preprint arXiv:2311.16495*, 2023. 1, 2, 6, 8

[42] Jian Wang, Diogo Luvizon, Weipeng Xu, Lingjie Liu, Kripasindhu Sarkar, and Christian Theobalt. Scene-aware egocentric 3d human pose estimation. *CVPR*, 2023. 1, 2, 3, 4, 6

[43] Alexander Winkler, Jungdam Won, and Yuting Ye. Questsim: Human motion tracking from sparse sensors with simulated avatars. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 1

[44] Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott Delp, and C. Karen Liu. Diffusion inertial poser: Human motion reconstruction from arbitrary sparse imu configurations, 2023. 1

[45] Erwin Wu, Rawal Khirodkar, Hideki Koike, and Kris Kitani. Soleposer: Full body pose estimation using a single pair of insole sensor. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2024. Association for Computing Machinery. 1

[46] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. Mo$^2$Cap$^2$ : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019. 1, 2, 3, 4

[47] Chenhongyi Yang, Anastasia Tkach, Shreyas Hampali, Linguang Zhang, Elliot J Crowley, and Cem Keskin. Egoposeformer: A simple baseline for egocentric 3d human pose estimation. *arXiv preprint arXiv:2403.18080*, 2024. 1, 2, 6, 7

[48] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[49] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Vladislav Golyanik, Shaohua Pan, Christian Theobalt, and Feng Xu.

Egolocate: Real-time motion capture, localization, and mapping with sparse body-mounted sensors. *ACM Transactions on Graphics (TOG)*, 42(4), 2023. 1

[50] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10082–10092, 2019. 1

[51] Yu Zhang, Songpengcheng Xia, Lei Chu, Jiarui Yang, Qi Wu, and Ling Pei. Dynamic inertial poser (dynaip): Part-based motion dynamics learning for enhanced human pose estimation with sparse inertial sensors. *arXiv preprint arXiv:2312.02196*, 2023. 1

[52] Amy Zhao, Chengcheng Tang, Lezi Wang, Yijing Li, Mihika Dave, Lingling Tao, Christopher D. Twigg, and Robert Y. Wang. Egobody3m: Egocentric body tracking on a vr headset using a diverse dataset. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXIX*, page 375–392, Berlin, Heidelberg, 2024. Springer-Verlag. 1

[53] Dongxu Zhao, Zhen Wei, Jisan Mahmud, and Jan-Michael Frahm. Egoglass: Egocentric-view human pose estimation from an eyeglass frame. In *2021 International Conference on 3D Vision (3DV)*, pages 32–41, 2021. 1, 2, 3, 4, 6, 7