# EVENT-BASED EGOCENTRIC HUMAN POSE ESTIMATION IN DYNAMIC ENVIRONMENT

*Supplementary Material*

## Contents



## 1. OVERVIEW OF THE SUPPLEMENTARY MATERIAL

The supplementary material includes details on the model architecture, implementation, baselines, and synthetic dataset. We provide a video demo to obtain more qualitative results

## 2. MODEL ARCHITECTURE

**Motion Segmentation Module** The Motion Segmentation Module is a network based on U-Net [1]. U-Net consists of an encoder and a decoder, with the input being voxel grid [2] and the output being a segmentation mask. The encoder extracts features by reducing the resolution through convolutional layers, ReLU activation, and max pooling. The decoder restores the resolution using transposed convolutions and skip connections to recover detailed information.

**Monocular SLAM** We used the pre-trained model of Droid-SLAM [3] for camera pose estimation. Droid-SLAM is a frame-based method that can robustly estimate camera pose by taking inputs such as RGB and grayscale video. Although our method takes a voxel grid generated from the event cloud as input, Droid-SLAM is a pre-trained model on various datasets, reducing the domain gap.

**Optical Flow** We used the pre-trained model of ResNet-18 [4] for optical flow estimation. Although the input is a voxel grid, the optical flow reduces the domain gap due to differences in appearance because the information is reduced to a low dimension.

**HeadNet, GravityNet, and Full-Body Pose Estimation Module** HeadNet, GravityNet, and the Full-Body Pose Estimation Module are networks based on EgoEgo [5], and we implemented them based on the provided baseline method code.

## 3. IMPLEMENTATION DETAILS

**Motion Segmentation Module** We utilized an NVIDIA GeForce RTX 4090 GPU, and the Motion Segmentation Module training process reached convergence in roughly 23 hours. The training was conducted for 100 epochs with a batch size of 32, a learning rate of $1.0 \times 10^{-5}$, and AdamW [6] as the optimization algorithm.

**HeadNet** The implementation of HeadNet was based on the baseline method EgoEgo, and it was trained from scratch. An NVIDIA Quadro A6000 GPU was employed, and the training process completed within approximately 1 hour. Due to differences in input, both the baseline model and our proposed method were trained. For both, we used a batch size of 64, a learning rate of $1.0 \times 10^{-4}$, and AdamW [6] as the optimizer.

**GravityNet** GravityNet was implemented following the baseline method EgoEgo and trained from scratch. The training was performed using an NVIDIA Quadro A6000 GPU and reached convergence in about 1 hour. A batch size of 256, a learning rate of $1.0 \times 10^{-4}$, and AdamW [6] were used for optimization.

**Full-Body Pose Estimation Module** We retrained the Full-Body Pose Estimation Module based on the pre-trained model of EgoEgo. We used an NVIDIA RTX 6000 Ada Generation GPU, and the retraining of the Full-Body Pose Estimation Module converged in approximately 32 hours. The training setup included a batch size of 32, a learning rate of $1.0 \times 10^{-4}$, and AdamW [6] as the optimization technique.

## 4. SYNTHETIC DATASET

Synthetic dataset was created based on the EgoBody [7] dataset. First, event data was generated from the RGB first-person view videos of the EgoBody dataset using the event simulator DVS-Voltmeter [8]. Subsequently, voxelization [2] was performed to generate a voxel grid at 30fps.

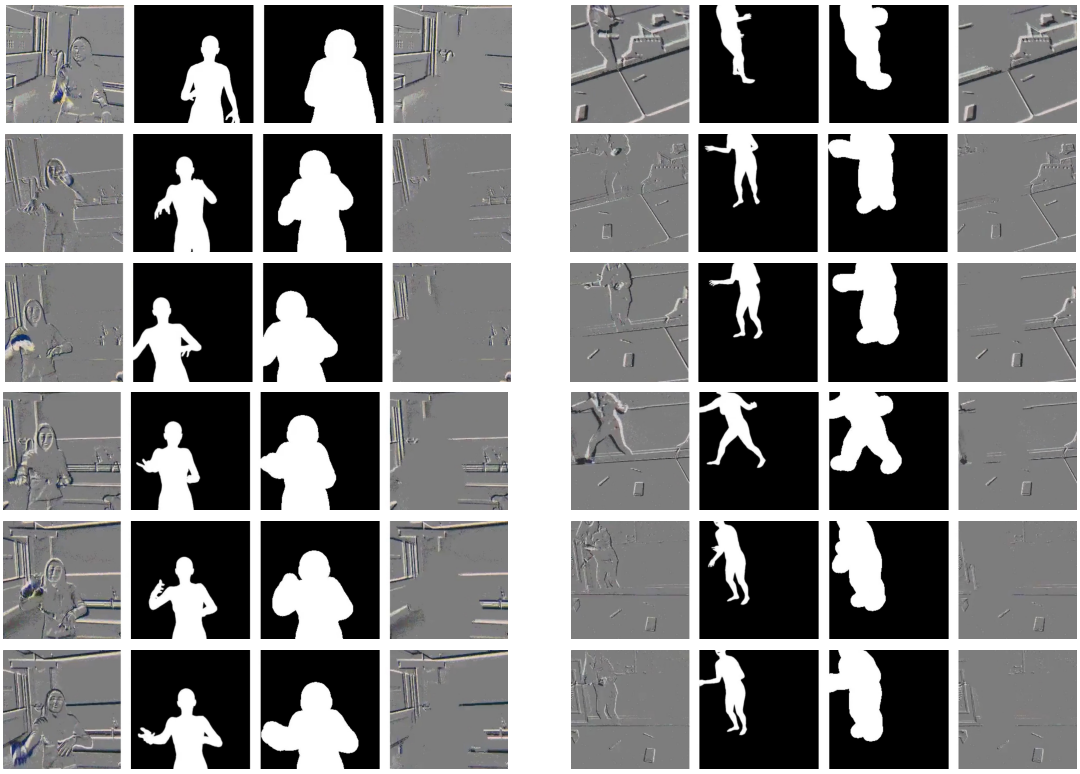

**Fig. 1**: **Synthetic Dataset and Motion Segmentation Results.** From left to right: the input voxel grid, the mask generated using the ground truth pose, the dilated mask, and the output voxel grid of the Motion Segmentation Module.

Furthermore, dynamic masks were generated by projecting the ground truth poses of the EgoBody dataset onto the first-person view videos. This process created pairs of input and ground truth masks for the dataset Fig. 1 used in the training of the Motion Segmentation Module.

## 5. VIDEO QUALITATIVE EVALUATION

To conduct a qualitative evaluation of the proposed method, we provide a video demo. The video demo compares the results of our baseline, which inputs event data into EgoEgo [5], and our proposed method, D-EventEgo. The scenes include three different environments and experiments with different subjects. Our method demonstrates results that are closer to the ground truth, such as the position of the hands when standing and the height of the head when sitting.

## 6. REFERENCES


[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[2] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Unsupervised event-based optical flow using motion compensation," in *ECCVW*, 2018.

[3] Zachary Teed and Jia Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *NeurIPS*, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[5] Jiaman Li, Karen Liu, and Jiajun Wu, "Ego-body pose estimation via ego-head pose estimation," in *CVPR*, 2023.

[6] Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, and Francesco Orabona, "Understanding adamw through proximal methods and scale-freeness," *TMLR*, 2022.

[7] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang, "Egobody: Human body shape and motion of interacting people from head-mounted devices," in *ECCV*, 2022.

[8] Songnan Lin, Ye Ma, Zhenhua Guo, and Bihan Wen, "DVS-Voltmeter: Stochastic process-based event simulator for dynamic vision sensors," in *ECCV*, 2022.