Scene-aware Egocentric 3D Human Pose Estimation **Supplementary Material**

Jian Wang[1,2] Diogo Luvizon[1,2] Weipeng Xu[3] Lingjie Liu[1,2]
Kripasindhu Sarkar[4] Christian Theobalt[1,2]
[1]MPI Informatics [2]Saarland Informatics Campus [3]Meta Reality Labs [4]Google
{jianwang,dluvizon,lliu,theobalt}@mpi-inf.mpg.de xuweipeng@fb.com krsarkar@google.com

1. Datasets

1.1. Our Test Dataset

In this section, we introduce the data collection process for our test dataset. All personal data in our test dataset is collected with an IRB approval. In order to estimate accurate egocentric camera poses and further obtain the ground truth human body poses under the egocentric camera perspective, we mount a calibration board on the *head*, rigidly attach it to the egocentric camera, and estimate the pose of the egocentric camera with a multi-view capturing system, as shown in Fig. 1.

Before the data collection process, we first estimate the transformation matrix $\mathbf{M}_{\text{head2ego}}$ between the calibration board and the fisheye camera with hand-eye calibration [12]. We place a second calibration board on the scene in a place where it can be seen by both the egocentric camera and the studio cameras. We then estimate the relative pose $\mathbf{M}_{\text{ego2calib}}$ between the egocentric camera and the external calibration board, the relative pose between the studio cameras and the external calibration board $\mathbf{M}_{\text{ext2calib}}$, and the relative pose between the studio cameras and the head-mounted calibration board $\mathbf{M}_{\text{ext2head}}$. We can obtain the transformation matrix $\mathbf{M}_{\text{head2ego}}$ with:

$$\mathbf{M}_{\text{head2ego}} = \mathbf{M}^{-1}_{\text{ext2head}}\mathbf{M}_{\text{ext2calib}}\mathbf{M}^{-1}_{\text{ego2calib}} \quad (1)$$

During the data collection process, we estimate the pose of the calibration board from each single view and obtain the averaged calibration board poses $\mathbf{M}_{\text{ext2head}}$ (see Fig. 1). The egocentric camera pose $\mathbf{M}_{\text{ext2ego}}$ can be obtained with:

$$\mathbf{M}_{\text{ext2ego}} = \mathbf{M}_{\text{ext2head}}\mathbf{M}_{\text{head2ego}} \quad (2)$$

With the egocentric camera pose, we can transform the ground truth pose under the studio camera coordinate system P_{ext} to the egocentric camera coordinate system P_{ego}:

$$P_{\text{ego}} = P_{\text{ext}}\mathbf{M}_{\text{ext2ego}} \quad (3)$$

Figure 1. Visualization of the data collection process for our test dataset. We detect the pose of the head-mounted calibration board rigidly attached to the egocentric camera from multiple views in the studio.

1.2. EgoGTA Dataset

Based on the GTA-IM dataset [2], we generate the synthetic EgoGTA dataset with ground truth labels of human body segmentation masks, scene depth maps, and human body poses. We first register the SMPL-X model on the 3D poses from GTA-IM following HULC [11]. Then, we use the TSDF fusion [3] to reconstruct the mesh of the scene from the depth map sequences in GTA-IM. Finally, we render the images, semantic labels, and depth maps of the scene with and without the human body using Blender [1]. We show more examples of the EgoGTA dataset in Fig. 2

1.3. EgoPW-Scene Dataset

We generate the EgoPW-Scene dataset by rendering the scene depth map for each image in the EgoPW dataset [13]. Since the scan of the background scene is not available for the EgoPW dataset, we generate the mesh of the scene from the EgoPW image sequences with SfM. In Fig. 3 we show more examples of the EgoPW-Scene dataset.

2. Implementation Details

In this section, we describe the implementation details of our scene-aware egocentric pose estimation framework, including the network architectures and training procedure. Details of the scene depth estimator is shown in Sec. 2.1, which includes a human body segmentation network (Sec. 2.1.2), a depth estimation network (Sec. 2.1.1)

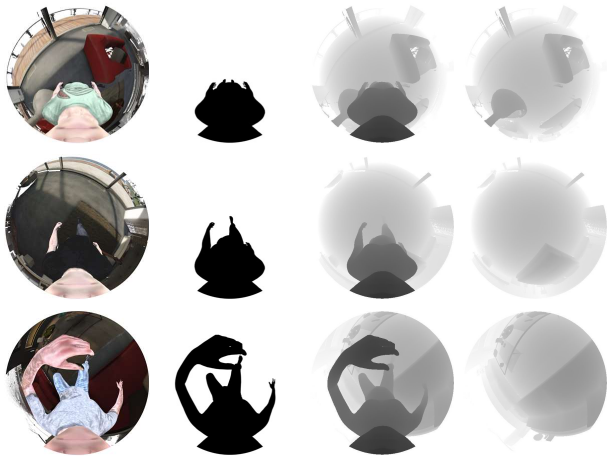

Figure 2. Example of our EgoGTA dataset.

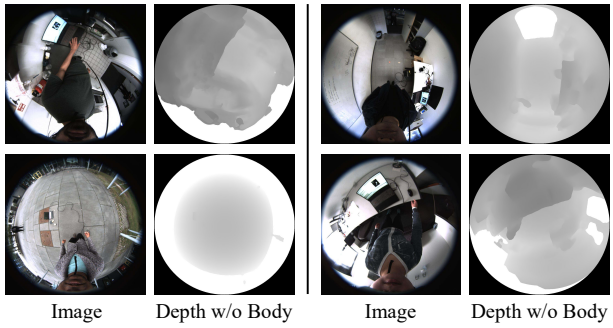

Figure 3. Example of our EgoPW-Scene dataset.

and a depth inpainting network (Sec. 2.1.3). The details of the scene-aware egocentric pose estimator are shown in Sec. 2.2.

2.1. Scene Depth Estimator

2.1.1 Depth Estimation Network with Human Body

We use the same network architecture from Hu *et al.*'s work [5] as our depth estimation network $\mathcal{D}$. The network $\mathcal{D}$ is trained on the NYU-Depth V2 dataset [8] following the training procedure from [5]. Next, the network is finetuned on the EgoGTA dataset using the Adam optimizer [6] for 40K iterations with the learning rate set to $1 \cdot 10^{-4}$, the weight decay set to $1 \cdot 10^{-4}$, the image size as 256×256, and the batch size as 16.

2.1.2 Human Body Segmentation Network

We adopt the HRNetV2-W48 network from Yuan *et al.*'s work [15] as our human body segmentation network $\mathcal{S}$. The network $\mathcal{S}$ is trained on the LIP dataset [4] following the procedure from [15]. Next, we finetune the network on the EgoGTA dataset for 2000 steps with the weight decay as $1 \cdot 10^{-3}$, the image size as 473×473, and the batch size as 32. During the finetuning step, we use the Adadelta [16] optimizer and set the learning rate of the first 3 stages in HRNet to $1 \cdot 10^{-6}$ and the learning rate of the fourth stage to 0.001.

2.1.3 Depth Inpainting Network

The depth inpainting network $\mathcal{G}$ takes the segmented depth map $\hat{D}^M$ with shape 256×256 and the human body segmentation mask $\hat{S}$ with shape 256×256 as the input and predicts the scene depth map without human body $\hat{D}^S$. We adopt the UNet [9] for the depth inpainting task. The encoder of the UNet contains one input convolutional layer with 64 output channels and 4 downsampling layers, each with 128, 256, 512, 512 output channels. Each downsampling layer consists of one 2D-maxpooling layer (kernel size 2) and two convolutional blocks. The decoder contains 4 upsampling layers, each with 256, 128, 64, 64 output channels, and one output convolutional layer with 1 output channel. Each upsampling layer consists of one 2D-bilinear interpolation layer and two convolutional blocks. Each aforementioned convolutional block contains one 2D convolutional layer (kernel size 3, stride 1, and padding 1), one batch norm layer, and one relu layer. The 1st, 2nd, 3rd and 4th input of the downsampling layers is also fed into the 4th, 3rd, 2nd and 1st input of the upsampling layer to form the skip connections in UNet.

We train the depth inpainting network on the EgoGTA and the EgoPW-Scene datasets simultaneously using the Adam optimizer [6] for 28K iterations with the learning rate as $1 \cdot 10^{-4}$, the weight decay as $1 \cdot 10^{-4}$ and batch size as 16.

2.2. V2V Network

The V2V network has the same architecture as the network in Moon *et al.*'s work [7]. During training, the input image is converted to 2D body pose features and further projected into a 3D volumetric space V_{body} with 32 channels. We concatenate the volumetric body feature, the volumetric representation of ground truth scene geometry V_{scene}, and their intersection V_{inter} and feed them to the V2V network. We train the network using the Adam optimizer for 24K iterations with the learning rate as $1 \cdot 10^{-3}$ and the batch size as 64.

3. Fisheye Camera Model

In this section, we describe the fisheye camera model used in our method. The projection function $\mathcal{P}$ of a 3D point $[x, y, z]^T$ into a 2D point $[u, v]^T$ on fisheye images

can be written as:

$$[u, v]^T = \frac{[x, y]^T}{\sqrt{x^2 + y^2}} \times f(\rho) \tag{4}$$

where $\rho = \arctan(z/\sqrt{x^2 + y^2})$ and $f(\rho) = \alpha_0 + \alpha_1\rho + \alpha_2\rho^2 + \alpha_3\rho^3 + \dots$ is a polynomial obtained from camera calibration.

Given a 2D point $[u, v]^T$ on the fisheye images and the distance d between the 3D point and the camera, the position of the 3D point $[x, y, z]^T$ can be obtained with the fisheye reprojection function $\mathcal{P}^{-1}$:

$$[x, y, z]^T = \frac{[u, v, f'(\rho')]^T}{\sqrt{u^2 + v^2 + (f'(\rho'))^2}} \times d \tag{5}$$

where $\rho' = \sqrt{u^2 + v^2}$ and $f'(\rho) = \alpha'_0 + \alpha'_1\rho + \alpha'_2\rho^2 + \alpha'_3\rho^3 + \dots$ is another polynomial obtained from camera calibration. The calibration of the fisheye camera and more details about the fisheye camera model can be found in Scaramuzza *et al*. [10].

4. Evaluation Metrics

In this section, we give a detailed explanation of the evaluation metrics used in our method. MPJPE is the mean of Euclidean distances for each joint in the predicted and ground truth poses. For PA-MPJPE, we rigidly align the estimated pose to the ground truth pose with Procrustes analysis and then calculate MPJPE. For BA-MPJPE, we first resize the bone length of predicted poses and ground truth poses to the bone length of a standard human skeleton. Then, we calculate the PA-MPJPE between the two resulting poses.

5. Ablation Study on Wang *et al.*'s Dataset

In this section, we run the ablation study in Sec. 4.4 in the main paper on Wang *et al.*'s dataset [14] and show the result in Table 1. Since Wang *et al.*'s dataset does not provide the ground truth scene geometry and ground truth pose annotations in the egocentric camera coordinate system, we only evaluate with PA-MPJPE and BA-MPJPE metrics.

The ablation study shows similar results on Wang *et al.*'s dataset as on our test dataset, which demonstrates similar conclusions in Sec. 4.4 in the main paper.

Method	PA-MPJPE	BA-MPJPE
EgoPW+Optimizer	79.06	63.56
EgoPW+Depth	78.41	63.75
xR-egopose+Depth	109.7	85.74
Ours w/o Depth	81.04	64.18
Ours+Depth with Body	82.98	65.09
Ours+Depth w/o Body	78.95	64.83
Ours+Depth w/o Inpainting	82.39	66.43
Ours	**76.50**	**61.92**

Table 1. Results from our method compared to different baselines.

References

[1] Blender. http://www.blender.org. 1

[2] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*. 2020. 1

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 1

[4] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 932–940, 2017. 2

[5] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[7] Gyeongsik Moon, Juyong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[8] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[10] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, 2006. 3

[11] Soshi Shimada, Vladislav Golyanik, Zhi Li, Patrick Pérez, Weipeng Xu, and Christian Theobalt. Hulc: 3d human motion capture with pose manifold sampling and dense contact guidance. In *European Conference on Computer Vision*, pages 516–533. Springer, 2022. 1

[12] Roger Y Tsai and Reimar K Lenz. Real time versatile robotics hand/eye calibration using 3d machine vision. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 554–561. IEEE, 1988. 1

[13] 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. In *CVPR*, pages 13157–13166, June 2022. 1

[14] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, and Christian Theobalt. Estimating egocentric 3d human pose in global space. *ICCV*, 2021. 3

[15] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European conference on computer vision*, pages 173–190. Springer, 2020. 2

[16] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 2