Estimating Egocentric 3D Human Pose in the Wild with Weak External Supervision **Supplementary Material**

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

1. Quantitative Results on Different Motions

In Table 2 of our main paper, we show that our method outperforms the state-of-the-art methods: Mo^{2}Cap^2 and xR-egopose. In order to further compare the performance on different types of motions, we show the quantitative comparisons on Wang *et al*. [7]'s test dataset in Table 1 and on Mo^{2}Cap^2 dataset [9] in Table 2. We show that our method outperforms all of the baselines on most types of motion in these results. Note that our method is trained on the EgoPW dataset while the focal length and distortion of the fisheye camera in the EgoPW dataset is different from the fisheye camera used in Mo^{2}Cap^2, which affects the performance of our method on the Mo^{2}Cap^2 test dataset.

Method	Mo^{2}Cap^2	xR-egopose	Ours
walking	69.68	84.20	**59.65**
running	77.88	76.78	**63.84**
crouching	**63.28**	96.86	68.87
boxing	79.37	85.74	**72.91**
dancing	82.65	94.23	**65.21**
stretching	117.7	119.9	**108.8**
waving	53.14	72.66	**44.57**
playing balls	60.95	95.30	**56.54**
open door	55.88	71.70	**49.06**
play golf	113.8	94.41	**94.29**
talking	53.93	78.10	**51.82**
shooting arrow	67.07	76.75	**60.71**
sitting	83.24	69.10	**65.06**
total (mm)	74.46	87.20	**64.87**

Table 1. The BA-MPJPE of different types of motions on the test set of Wang *et al*. [7]. Our approach outperforms Mo^{2}Cap^2 results by 9.59 mm and outperforms xR-egopose results by 22.33 mm.

2. Qualitative Results

In this section, we show more qualitative results for the in-the-wild images from the test sequence of either EgoPW in Figure 1 or Mo^{2}Cap^2 in Figure 2. These results show that our method significantly outperforms the state-of-the-art methods especially when the body parts are occluded.

3. Details and Comparisons of EgoPW dataset

The details of the EgoPW dataset and comparisons between EgoPW and other 3D pose estimation datasets are shown in Table 3. Our dataset contains 97 sequences and 318k frames in total, which is performed by 10 actors in 20 clothing styles. The actions in the EgoPW dataset includes reading magazine/newspaper, playing board games, doing a presentation, walking, sitting down, using a computer, calling on the phone, drinking water, writing on the paper, writing on the whiteboard, making tea, cutting vegetables, stretching, running, playing table tennis, playing baseball, climbing floors, dancing, opening the door and waving hands.

In Table 3, we further compare our EgoPW dataset with other datasets for external-view 3D pose estimation and egocentric view 3D pose estimation. Mo^{2}Cap^2 [9] and xR-egopose [5] provide large synthetic datasets for training the egocentric pose estimation networks. However, these datasets are synthesized and thus suffer from the domain gap with the real images. Mo^{2}Cap^2, xR-egopose and Wang *et al*. [7] also provide small test sequences with ground truth labels obtained with the mocap system. However, this dataset is not sufficient for training an egocentric pose estimation network. Our dataset contains a large amount of in-the-wild images with accurate pseudo labels generated with an optimization framework, which facilitates training the pose estimation network with in-the-wild images.

The publicly available large datasets for 3D pose estimation from an external view, like Human 3.6M [2] and MPI-INF-3DHP [3], are all collected in the studio with a multi-view mocap system. This capturing method is not able to obtain in-the-wild images and the interactions between the

Indoor	walking	sitting	crawling	crouching	boxing	dancing	stretching	waving	total (mm)
Mo^{2}Cap^2	38.41	70.94	94.31	81.90	48.55	55.19	99.34	60.92	61.40
xR-egopose	**37.35**	64.45	87.41	69.68	**45.19**	54.76	90.89	**49.41**	55.43
Ours	40.23	**60.22**	**70.88**	**62.40**	49.89	**52.41**	**82.48**	59.60	**54.78**
Outdoor	walking	sitting	crawling	crouching	boxing	dancing	stretching	waving	total (mm)
Mo^{2}Cap^2	63.10	**85.48**	96.63	92.88	96.01	68.35	123.56	**61.42**	80.64
xR-egopose	62.01	103.45	86.53	80.43	90.48	66.06	117.55	67.49	78.30
Ours	**58.06**	94.19	**85.50**	**77.61**	**83.91**	**62.56**	**111.9**	65.37	**74.55**

Table 2. The BA-MPJPE of different types of motions on the indoor and outdoor sequence of Mo^{2}Cap^2 dataset [9]. In the indoor sequence, our method improves the Mo^{2}Cap^2 [9] results by 6.62 mm and xR-egopose results by 0.65 mm; In the outdoor sequence, our method improves the Mo^{2}Cap^2 [9] results by 6.09 mm and xR-egopose results by 3.75 mm.

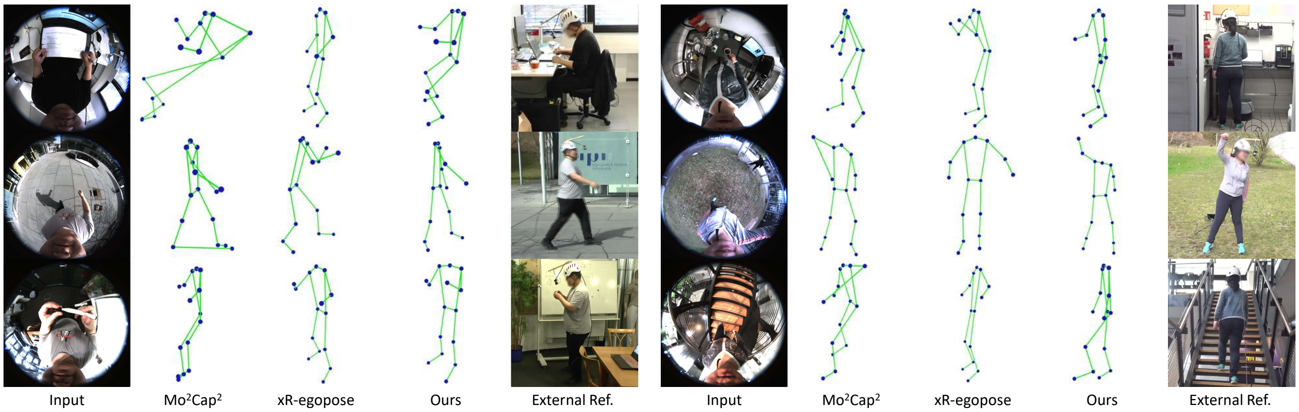

Figure 1. Qualitative comparison between our method and the state-of-the-art methods on the test images of the EgoPW dataset. From left to right: input image, Mo^{2}Cap^2 result, xR-egopose result, our result, and external image. Note that the external images are only for visualization and they are not used for predicting the pose.

human body and the environment. 3DPW [6] is a dataset collected in the in-the-wild scenes with pseudo labels obtained from a moving camera and an IMU system. This capturing method provides accurate pseudo labels for body pose with various interactions between the human body and the environment. However, this dataset only contains 51k frames, which is less than the frames in our EgoPW dataset. All of the aforementioned datasets do not contain any egocentric images and thus cannot be used for training the egocentric pose estimation networks.

4. Network Architecture

In this section, we describe the architecture of the pose estimation network and domain classifier network used in our method.

4.1. Pose Estimation Network

We use the architecture in Mo^{2}Cap^2 [9] for obtaining the 3D poses and 2D heatmaps. The pose estimation network contains a 2D module for the full-body heatmap, a 2D module for zoomed-in body heatmap, and a 3D module. The 2D module for full-body pose can be represented as an encoder-decoder network, which first gets the features $\mathcal{F}_{Full2D}$ with a Resnet-50 network [1] as the encoder and uses the features $\mathcal{F}_{Full2D}$ to predict the full-body heatmap with convolutional layers. The 2D module for zoomed-in body heatmaps has the same architecture as the former one. It takes the zoomed-in egocentric images as input and first generates features $\mathcal{F}_{Zoom2D}$ and predicts zoomed-in heatmaps from the intermediate features. The full-body heatmaps and zoomed-in heatmaps are finally averaged to get the final prediction of heatmaps $\hat{\mathcal{H}}$. The distance module takes the features from both the aforementioned 2D modules as input and predicts the distances $\hat{\mathcal{D}}$ between body joints and the camera. More details about the pose estimation network can be found in Mo^{2}Cap^2 [9].

4.2. Domain Classifier Γ

The domain classifier takes the intermediate features $\mathcal{F}_{Full2D}$ with shape $2048 \times 8 \times 8$ or $\mathcal{F}_{Zoom2D}$ with shape $2048 \times 8 \times 8$ as input and predicts whether the input feature is from synthetic or real image. The network contains two Resnet “bottleneck” blocks [1] with 1024 and 256 output channels and one final classification block. The classifica-

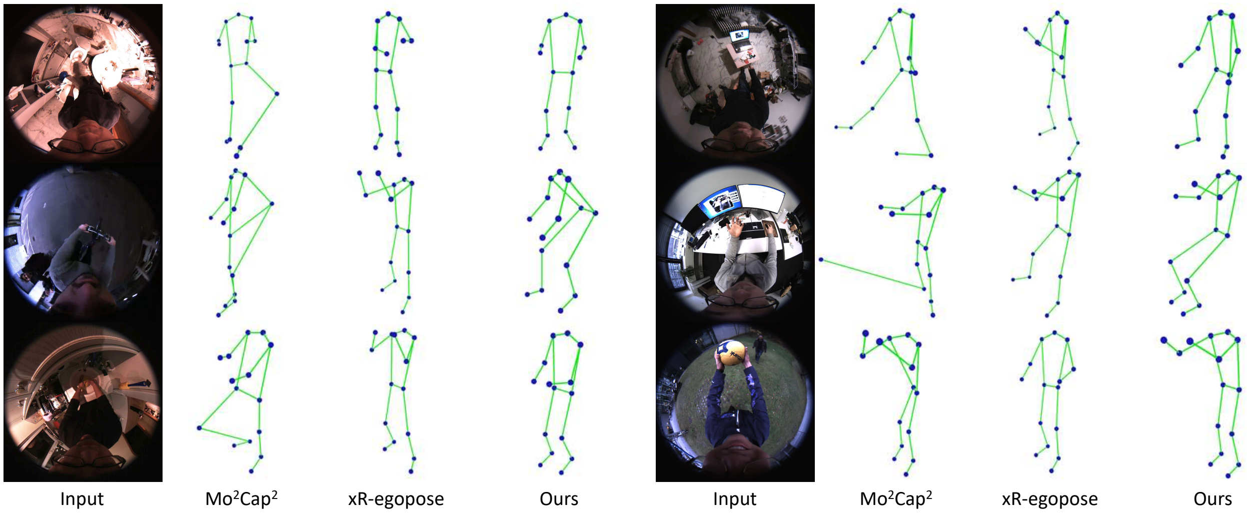

Figure 2. Qualitative comparison between our method and the state-of-the-art methods on the test images of Mo^{2}Cap^2 work. From left to right: input image, Mo^{2}Cap^2 result, xR-egopose result, and our result.

Dataset Name	Frames	Sequences	Subjects	Context	Action Types
Human 3.6M [2]	3.6M	1376	11	Studio	17
MPI-INF-3DHP [3]	1.3M	64	16	Studio	8
3DPW [6]	51k	60	18	In the wild	8
Mo2Cap2 [9]	530k	-	700	Synthetic	3000
Mo2Cap2-test	5591	2	2	Studio & in the wild	8
xR-egopose [5]	383k	-	-	Synthetic	9
xR-egopose-test	10k	-	3	Studio	6
Wang *et al*. [7]	47k	19	9	Studio	13
EgoPW	318k	97	10	In the wild	20

Table 3. Comparison between the EgoPW dataset and publicly available 3D pose estimation datasets.

tion block contains two convolutional blocks and a linear layer for the domain classification task. The first convolutional block contains one 2D convolutional layer (kernel size=4, stride=2, and padding=1), one batch norm layer, and one relu layer. The second convolutional block contains one 2D convolutional layer (kernel size=3, stride=2, and padding=1), one batch norm layer, and one relu layer. The output features of the convolutional blocks are sent to the linear layer giving the domain label prediction.

4.3. Egocentric-external View Classifier Λ

Similar to the domain classifier for distinguishing synthetic and real images, the egocentric-external view classifier also takes the intermediate features $\mathcal{F}_{Full2D}$ with shape $2048 \times 8 \times 8$ or $\mathcal{F}_{Zoom2D}$ with shape $2048 \times 8 \times 8$ as input and predicts whether the input feature is from the egocentric view or the external view. The network contains two convolutional blocks, one global average pooling layer, and one final classification block. The intermediate features are firstly sent to the convolutional blocks and then generate features with shape $1024 \times 8 \times 8$. The spatial dimension of the features is eliminated with a global average pooling layer [10] to generate a feature vector with length 1024. Next, the feature vector is sent to the final classification block to predict whether the input feature is from the egocentric view or the external view. Each of the convolutional blocks consists of one 2D convolutional layer (output channel=1024, kernel size=3, stride=2, and padding=1), one batch norm layer, and one relu layer. The classification block includes one fully connected layer (output dimension=256), one batch norm layer, one relu layer, and one final fully connected layer (output dimension=2) which predicts the labels of egocentric/external views.

5. Fisheye Camera Model

In this section, we describe the fisheye camera model used in our method. The projection of a 3D point $[x, y, z]^T$

into a 2D point $[u, v]^T$ on fisheye images can be written as:

$$[u, v]^T = \frac{[x, y]^T}{\sqrt{x^2 + y^2}} \times f(\rho) \tag{1}$$

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 written as:

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

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 are described in Scaramuzza *et al*. [4].

6. Energy Function

In this section, we describe some of the terms in our objective function (Eq. 3).

$$\begin{aligned} E(\mathcal{P}^{ego}_{seq}, R_{seq}, t_{seq}) = {} & \lambda^{ego}_R E^{ego}_R + \lambda^{ext}_R E^{ext}_R + \lambda^{ego}_J E^{ego}_J \\ & + \lambda^{ext}_J E^{ext}_J + \lambda_T E_T + \lambda_B E_B \\ & + \lambda_C E_C + \lambda_M E_M \end{aligned} \tag{3}$$

In this function, E^{ext}_R, E^{ext}_J, E_C, and E_M are the external reprojection term, external 3D pose regularization term, camera pose consistency term, and camera matrix regularization term respectively which have already been described in the paper. E^{ego}_R, E^{ego}_J,E_T, and E_B are the egocentric reprojection term, egocentric pose regularization term, motion smoothness regularization term and bone length regularization term, which are the same as the corresponding terms in [7]. We also depict these terms here:

Heatmap-based Reprojection: With this term, we maximize the summed heatmap values at the reprojected 2D joint positions:

$$E_R(\mathcal{P}^{ego}_{seq}) = -\sum_{i=1}^{B} \|\mathrm{HM}_i(\Pi(\mathcal{P}^{ego}_i))\|_2^2 \tag{4}$$

where $\mathrm{HM}_i(.)$ returns the value at a pixel on $\mathcal{H}^{ego}_i$, the heatmap of i-th frame. $\Pi(.)$ refers to the projection of a 3D point with the fisheye camera model.

Pose Regularization: The pose regularizer is defined to constrain the optimized pose $\mathcal{P}^{ego}_i$ to stay close to the initial pose $\widetilde{\mathcal{P}}^{ego}_i$.

$$E_J(\mathcal{P}^{ego}_{seq}, \widetilde{\mathcal{P}}^{ego}_{seq}) = \sum_{i=1}^{B} \left\| \mathcal{P}^{ego}_i - \widetilde{\mathcal{P}}^{ego}_i \right\|_2^2 \tag{5}$$

Motion Smoothness Regularization: In this term, we constrain the acceleration of each joint over the whole sequence to improve the temporal stability of the estimated poses:

$$E_T(\mathcal{P}^{ego}_{seq}) = \sum_{i=2}^{B} \left\| \nabla\mathcal{P}^{ego}_i - \nabla\mathcal{P}^{ego}_{i-1} \right\|_2^2 \tag{6}$$

where $\nabla\mathcal{P}^{ego}_i = \mathcal{P}^{ego}_i - \mathcal{P}^{ego}_{i-1}$.

Bone Length Regularization: In this term, we calculate the difference between the bone length and the average bone length to enforce the length of each bone to be consistent.

$$E_B(\mathcal{P}^{ego}_{seq}) = \sum_{i=1}^{B} \left\| L_{\mathcal{P}^{ego}_i} - \frac{1}{B}\sum_{j=1}^{B} L_{\mathcal{P}^{ego}_j} \right\|_2^2 \tag{7}$$

where the $L_{\mathcal{P}^{ego}_i}$ is the length of each bone of 3D pose $\mathcal{P}^{ego}_i$.

7. Licenses

In our paper, we have used three available assets:

- Synthetic and test dataset in Mo^{2}Cap^2 [9];
- Test dataset from Wang *et al*. [7];
- Pretrained 2D pose estimation network from Xiao *et al*. [8].

7.1. License for Synthetic and Test Dataset in Mo^{2}Cap^2

Copyright (c) 2017 MPI for Informatics

Permission is hereby granted, free of charge, to any person or company obtaining a copy of this software, dataset and associated documentation files (the ”Software”) from the copyright holders to use the Software for any non-commercial purpose. Methods and models that make use of the provided Software in any way can only be used for non-commercial purposes. Publication, redistribution and (re)selling of the Software, of modifications, extensions, and derivatives of it, and of other Software containing portions of the licensed Software, are not permitted. The Copyright holder is permitted to publically disclose and advertise the use of the Software by any licensee. The above is subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the

Software, as well as any Software including substantial portions of the Software.

If the Software is used, the licensee is required to cite the use of the following publications in any documentation or publication that results from the work:

[1] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, Christian Theobalt. Mo2Cap2: Real-time Mobile 3D Motion Capture with a Cap-mounted Fisheye Camera. IEEE TVCG Proc. VR 2019.

THE SOFTWARE IS PROVIDED ”AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

7.2. License for Test Dataset from Wang et al.

Copyright (c) 2017 MPI for Informatics

Permission is hereby granted, free of charge, to any person or company obtaining a copy of this software, dataset and associated documentation files (the ”Software”) from the copyright holders to use the Software for any non-commercial purpose. Methods and models that make use of the provided Software in any way can only be used for non-commercial purposes. Publication, redistribution and (re)selling of the Software, of modifications, extensions, and derivatives of it, and of other Software containing portions of the licensed Software, are not permitted. The Copyright holder is permitted to publically disclose and advertise the use of the Software by any licensee. The above is subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software, as well as any Software including substantial portions of the Software.

If the Software is used, the licensee is required to cite the use of the following publications in any documentation or publication that results from the work:

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

THE SOFTWARE IS PROVIDED ”AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

7.3. License for Pretrained 2D Pose Estimation Network from Xiao et al.

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ”Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ”AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

References

[1] 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. 2

[2] 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, jul 2014. 1, 3

[3] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 1, 3

[4] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cam-

eras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, 2006. 4

[5] Denis Tomè, Patrick Peluse, Lourdes Agapito, and Hernán Badino. xr-egopose: Egocentric 3d human pose from an HMD camera. In *IEEE International Conference on Computer Vision*, pages 7727–7737, 2019. 1, 3

[6] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018. 2, 3

[7] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, and Christian Theobalt. Estimating egocentric 3d human pose in global space. *ICCV*, 2021. 1, 3, 4

[8] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018. 4

[9] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, 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 Trans. Vis. Comput. Graph.*, 25(5):2093–2101, 2019. 1, 2, 3, 4

[10] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016. 3