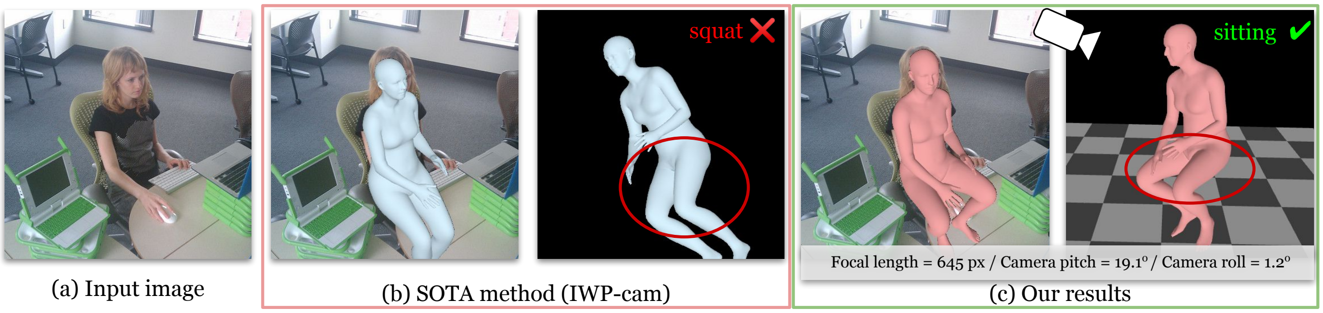
Supplementary Material for
SPEC: Seeing People in the Wild with an Estimated Camera

Anonymous Authors


squat ✗
sitting ✓
Focal length = 645 px / Camera pitch = 19.1° / Camera roll = 1.2°

(a) Input image (b) SOTA method (IWP-cam) (c) Our results

Figure 1: Comparison of two methods trained with the identical setting except for the camera models. (b) considers standard IWP-cam, while (c) is the proposed SPEC, which uses the additional camera parameters estimated by CamCalib.

In this supplementary document, we provide more information that is not covered in the main text, ranging from technical details in the method, visual examples of the proposed datasets, more qualitative results, more ablation studies as well as more analysis and discussion.

Acknowledgements: We thank Emre Aksan, Shashank Tripathi, Vassilis Choutas, Yao Feng, Priyanka Patel, Nikos Athanasiou, Yinghao Huang, Cornelia Kohler, Hongwei Yi, Dimitris Tzionas, Nitin Saini, and all Perceiving Systems department members for their feedback and the fruitful discussions. This research was partially supported by the Max Planck ETH Center for Learning Systems.

Disclosure: https://files.is.tue.mpg.de/black/CoI/ICCV2021.txt

1. Methods

1.1. Formulation of per-body translation t^b

For each body in the image, besides SMPL parameters θ, β, SPEC also estimates camera parameters (s, t_x, t_y), which is defined w.r.t. the bounding box (bbox) of the subject. Similar to [4, 7], we perform a coordinate transformation to obtain the final t^b vector w.r.t. the original full image following:

$$\begin{aligned} t^b_x &= t_x + \frac{2(c_x - w/2)}{s \cdot w_{bbox}}, \\ t^b_y &= t_y + \frac{2(c_y - h/2)}{s \cdot h_{bbox}}, \text{ and} \\ t^b_z &= \frac{2 \cdot f}{h_{bbox} \cdot s}, \end{aligned} \tag{1}$$

where (c_x, c_y) is the bbox center, w, h are originial image sizes, w_{bbox}, h_{bbox} are bbox sizes, and focal length f is estimated by CamCalib. As explained in the main text, Eq. 3, t^b is used during perspective projection $\Pi = K[R^c| - t^b]$.

1.2. Softargmax-$\mathcal{L}_2$ and Softargmax-biased-$\mathcal{L}_2$

As described in the main manuscript, we follow [22] to discretize the spaces of pitch α, roll ϕ and vertical field of view (vfov) υ into $B = 256$ bins but avoid casting it as a pure classification problem. To this end, we propose Softargmax-$\mathcal{L}_2$ loss and the biased variant. Let $\boldsymbol{\alpha} = [\alpha_1, \ldots \alpha_i, \ldots \alpha_B]$, $\boldsymbol{\phi} = [\phi_1, \ldots \phi_i, \ldots \phi_B]$, and $\boldsymbol{\upsilon} = [\upsilon_1, \ldots \upsilon_i, \ldots \upsilon_B]$ denote the center values of each of the bins, and let $\mathbf{p}^\alpha = [p^\alpha_1, \ldots p^\alpha_i, \ldots p^\alpha_B]$, $\mathbf{p}^\phi = \left[p^\phi_1, \ldots p^\phi_i, \ldots p^\phi_B\right]$, and $\mathbf{p}^\upsilon = [p^\upsilon_1, \ldots p^\upsilon_i, \ldots p^\upsilon_B]$ denote the probability mass from the fully-connected layers of each head respectively. We compute the expectation value of the probability mass as the prediction:

$$\begin{aligned} \hat{\alpha} &= \sum_i p^\alpha_i \alpha_i, \\ \hat{\phi} &= \sum_i p^\phi_i \phi_i, \text{ and} \\ \hat{\upsilon} &= \sum_i p^\upsilon_i \upsilon_i. \end{aligned} \tag{2}$$

This differentiable operation has been commonly-used in human joint detection [12, 19] to determine the peak location in a likelihood heat map, in contrast to the non-differentiable argmax operation.

For pitch α and roll ϕ angles, we apply the standard $\mathcal{L}_2$ loss between the prediction and the ground truth. To encourage underestimation of vfov υ more than overestimation, we design an asymmetric loss as depicted in Fig. 3 in the main text; formally:

$$\mathcal{L}(\hat{v}) = \begin{cases} \frac{(\hat{v}-v)^2}{(\hat{v}-v)^2+1}, & \text{if } \hat{v} <= v \\ (\hat{v}-v)^2, & \text{otherwise.} \end{cases} \tag{3}$$

We verify the benefits of these design choices in Table 1 in the main text.

1.3. Virtual ground plane

In all qualitative results, we visualize a virtual ground plane with a checkerboard, which is parallel to the xz-plane and therefore parameterized as $[0, y, 0]$. We define $y = \min\left(\mathcal{M}(\theta, \beta)[:, 2]\right)$, i.e. we place the ground plane just below the SMPL mesh.

This simple parameterization is feasible because we disentangle the camera rotation R^c from the body orientation R^b. For SOTA methods that apply the IWP-cam model, the virtual ground planes are often tilted. As a result, it requires further processing to estimate the up-vector, or conversely, the direction of gravity, making it non-trivial to integrate the reconstructed bodies for some downstream applications, e.g. scene understanding, character animation, physics simulation. SPEC, on the other hand, reconstructs bodies in the world coordinate frame with a consistent up vector $[0, 1, 0]$, which is more physically plausible when visualized together with the ground plane. See Sup. Mat. video.

1.4. SMPLify-X-Cam

To integrate the estimated camera parameters from CamCalib into an optimization-based method, we use the original implementation of SMPLify-X [17] with slight modifications. We replace the IWP-cam with the estimated K and R^c as described in Sec. 3.3 of the main text. Additionally, we initialize the optimization with the output of HMR-EFT [5], instead of starting from a mean pose and a mean shape. We use the Adam optimizer with a step size of 10^{-2} for 100 steps for both the first and second stage of optimization. The results of SMPLify-X-cam are evaluated in Sec. 3.4.

2. Implementation Details

2.1. SPEC-MTP Dataset

Müller et al. [15] propose a "Mimic The Pose" (MTP) task to collect datasets of natural images with high-quality pseudo ground truth body parameters from Amazon Mechanical Turk (AMT). Each image shows a person mimicking a posed SMPL-X mesh presented to them on AMT. Müller et al. devise a three-stage optimization routine, SMPLify-XC, to fit the SMPL-X model to each image. It is based on the default SMPLify-X [17] but considers additionally the pose $\tilde{\theta}$ and self-contact $\tilde{C}$ of the presented SMPL-X mesh to constrain the optimization. The body parts in self-contact are identified by finding vertices that are close in Euclidean and far in geodesic space. Please see [15] for a detailed definition of self-contact.

We follow the MTP approach but add two distinctions in order to obtain ground truth camera parameters: (1) Instead of getting a single picture for each pose, we ask AMT subjects to record a video showing the pose from multiple viewpoints, similar to Mannequin-Challenge style [10]. (2) We also ask them to print out a calibration pattern and record a video of the grid following a detailed protocol. In addition, we ask them to measure the size of the grid and take a picture of the grid and a ruler to verify the measured values.

To fit SMPL-X pose θ and shape β, as well as camera pitch α, roll ϕ, yaw ψ, and camera translation t^c to the collected MTP videos, we extend SMPLify-XC and introduce SMPLify-XC-Cam. We follow the three-stage optimization routine. In the first stage, body pose θ is initialized as the poses of the presented mesh, $\theta = \tilde{\theta}$, and stays fixed in this stage. The objective is:

$$E(\beta, \alpha, \phi, \psi, t^c) = \lambda_M E_M + \sum_{i=1}^{F} E_{J_i}. \tag{4}$$

E_M denotes the SMPLify-XC shape loss, that takes the ground truth height and weight of a person into account.

$$E_{J_i} = \left|\left|\omega_i \gamma\big(\Pi\big(\mathcal{M}(\beta, \theta), R_i^c, K, t_i^c\big) - \mathcal{J}_{2D_i}\big)\right|\right|_2^2 \tag{5}$$

is the 2D re-projection error of a single frame i with detected 2D joints $\mathcal{J}_{2D_i}$, and camera rotation R_i^c and translation t_i^c. ω_i andγ are per joint confidence and weight, respectively. F is the number of frames per video, extracted at one frame per second.

In the second and the third stage, we freeze body shape β and refine the pose θ and camera parameters by minimizing:

$$\begin{aligned} E(\theta, \alpha, \phi, \psi, t^c) =& \lambda_{m_h} E_{m_h} + \lambda_{\tilde{\theta}} E_{\tilde{\theta}} + \lambda_{\tilde{C}} E_{\tilde{C}} + \\ & \lambda_S E_S + \sum_{i=1}^{F} E_{J_i}. \end{aligned} \tag{6}$$

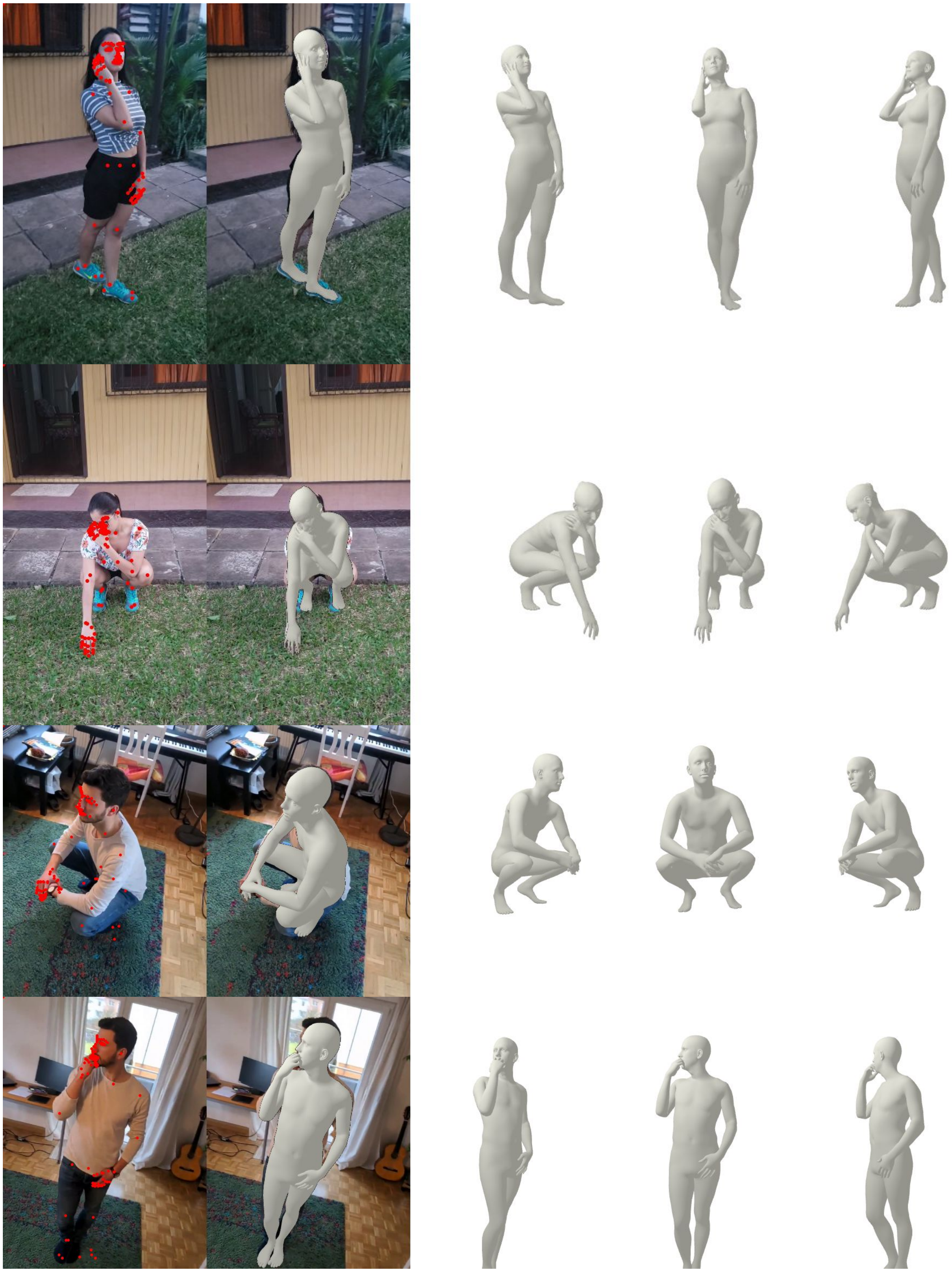
E_{m_h}, $E_{\tilde{\theta}}$, $E_{\tilde{C}}$, E_S denote the hand and presented pose priors, the presented contact loss and the general contact loss as defined in [15], respectively. Fig. 2 shows several examples of SPEC-MTP frames and the computed SMPL-X fit. SPEC-MTP is used only for evaluation.

2.2. SPEC-SYN Dataset

We obtain the 3D scans and SMPL-X fits to those scans from the AGORA dataset [16]. This includes many high-quality 3D scans of clothed people with accurate SMPL-X ground truth shape and pose. We convert the SMPL-X model to the SMPL format. We then put these scans in 3D scenes and use Unreal engine [2] to generate photorealistic images with diverse fields of view (fov) and camera rotations. Fig. 3 shows several examples, with SMPL fits overlaid on the images. One can observe some perspective distortion at the image boundary in the 3rd and 4th rows, indicating large fov (small focal length); the first and the last row show examples with high camera pitch angles.

2.3. Pano360 Dataset

To generate training dataset from equirectangular panorama images, we follow the strategy of Zhu et al. [22]. We crop images from panorama images with random viewpoints and focal lengths. Fig. 4 shows a sample panorama image along with the cropped images.

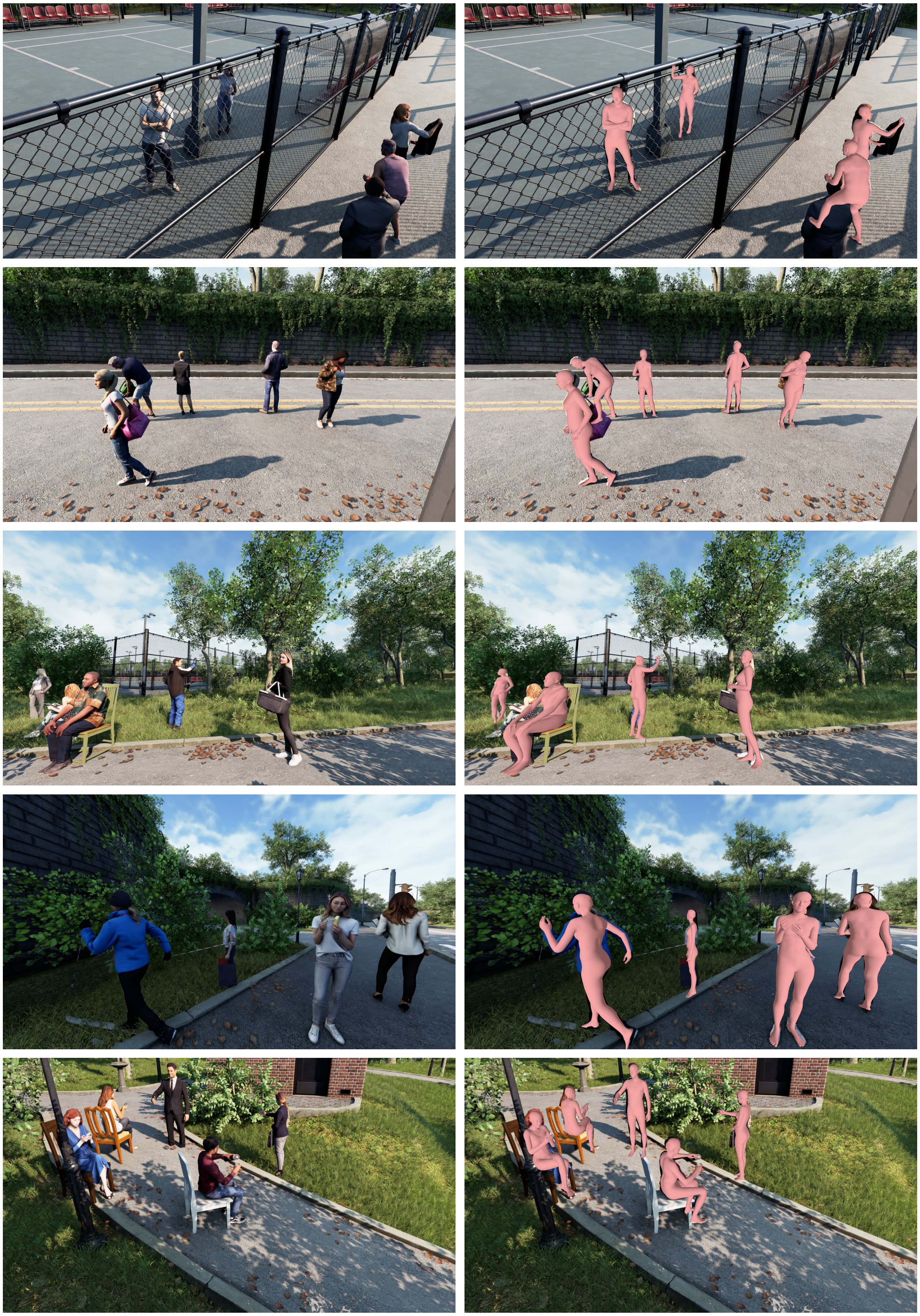


(a) Input Image / 2D keypoints

(b) SPEC-MTP annotation overlay

(c) SPEC-MTP annotation sideviews

Figure 2: **SPEC-MTP benchmark samples.**



(a) Rendered Image

(b) Annotations

Figure 3: **SPEC-SYN dataset samples.**

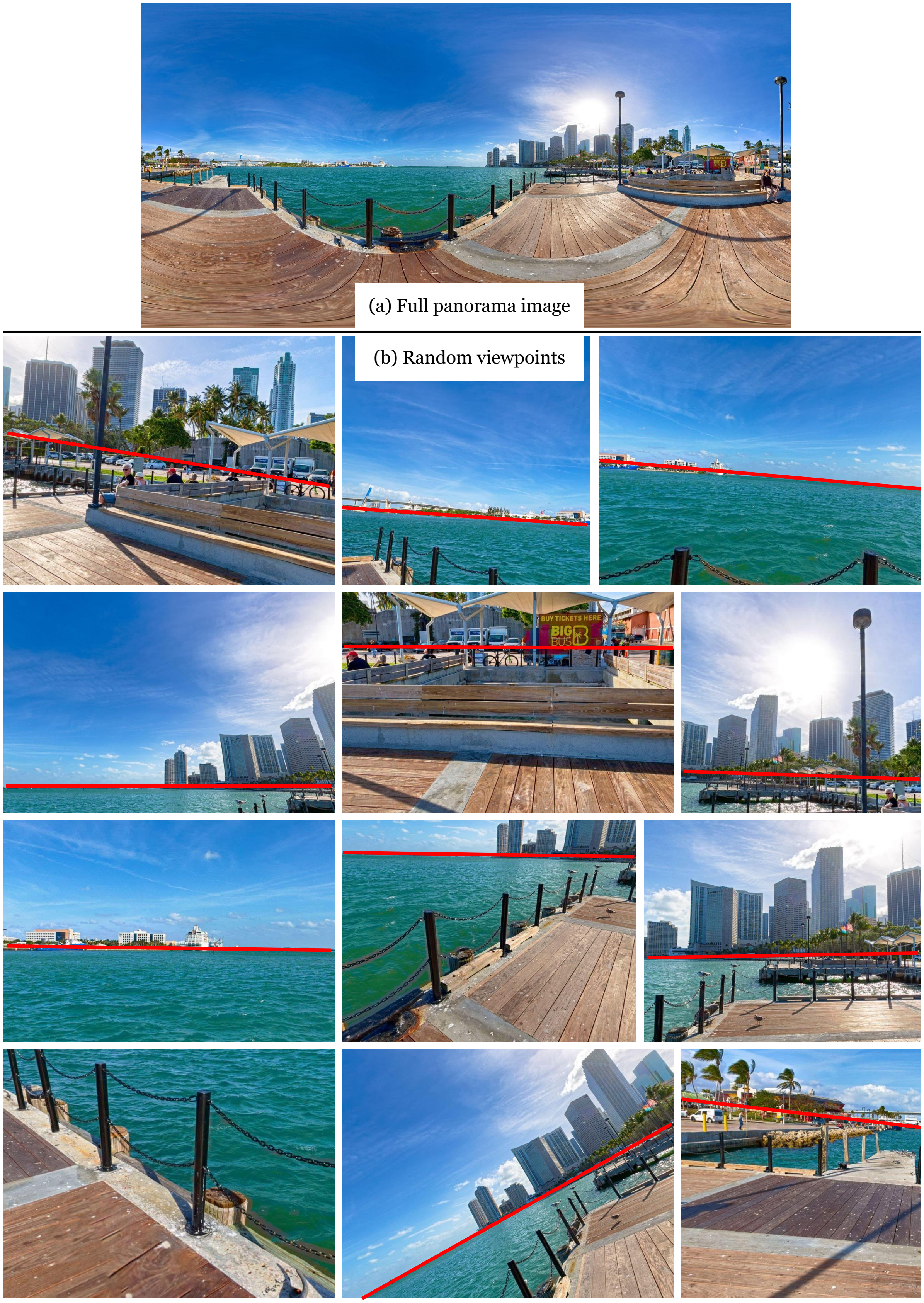


Figure 4: **Pano360 dataset.** Random viewpoints (b) from a single equirectangular panorama image (a). Horizon annotations are shown in red line.

3. Experiments

3.1. Training Datasets

In addition to SPEC-SYN dataset which is described in main text, we use datasets explained below for training.

MPI-INF-3DHP [13] is a multi-view indoor 3D human pose estimation dataset. 3D annotations are captured via a commercial markerless mocap software, therefore it is less accurate than some of the 3D datasets e.g. `Human3.6M` [3]. We use all of the training subjects S1 to S8 which makes 90K images in total.

Human3.6M [3] is an indoor, multi-view 3D human pose estimation dataset. Following previous methods, for training, we use 5 subjects (S1, S5, S6, S7, S8) which means 292K images.

COCO [11] dataset is a 2D keypoint dataset. In addition to 2D keypoint annotations, we utilize SMPLift-X-cam and CamCalib method to obtain SMPL and camera parameters annotations. We initialize the SMPLify-X-cam with SMPL fits provided by EFT [5] method.

Training Dataset Ratios. To obtain the final best performing model, we follow EFT [5] and SPIN [8] which use fixed data sampling ratios for each batch. We first train SPEC with 50% SPEC-SYN, 50% COCO for 175K steps. Then, we continue training with 20% Human3.6M, 20% MPI-INF-3DHP, with 50% SPEC-SYN, and 50% COCO for around 50K steps until convergence.

3.2. CamCalib Qualitative Results

In Fig. 5, we show the qualitative results of CamCalib. We follow [22] to visualize the estimated camera rotation by drawing the estimated horizons (red dashed lines). If the camera is pointing down (pitch angle $\alpha > 0$), the horizon should appear in the upper half of the image. Tilting to the left or right indicates the camera roll. CamCalib estimates reasonable camera parameters for most examples. We also show the failure cases in Fig. 6. We observe that they are all portrait images in which background contain little information for estimating camera parameters. We remark that despite no rich information in the background, human bodies still provide useful cues for the calibration purpose and we leave this to the future work.

3.3. SPEC MPJPE/PVE Results and Discussions

Table 1 to 3 summarize the performance of SPEC in comparison to SOTA methods on three datasets: SPEC-MTP, SPEC-SYN, and 3DPW. In addition to the three metrics, W-MPJPE, PA-MPJPE and W-PVE that are already reported in the main paper, we also include MPJPE and PVE here. The two versions of W-MPJPE and W-PVE are defined in Sec. 4.2 in the main text.

First, we observe that SPEC yields better "pure body pose" according to the improved PA-MPJPE. Moving onward, a 3DHPS method should learn not only to reconstruct body poses and shapes but also to place and orient them properly in the space. To this end, MPJPE and PVE are often considered stricter than the Procrustes-aligned counterparts as they measure additionally discrepancies in rotation. SPEC also outperforms SOTA methods [8, 9, 14, 18] in MPJPE/PVE, but yield on-par or slightly worse results than HMR*, which is an IWP-cam baseline trained under the identical setting as SPEC.

Methods	**MPJPE**	**W-MPJPE**	**PA-MPJPE**	**W-PVE**	**PVE**
GraphCMR [9]	150.9	175.1 / 166.1	94.3	205.5 / 197.3	179.9
SPIN [8]	129.4	143.8 / 143.6	79.1	165.2 / 165.3	148.2
PartialHumans [18]	150.1	158.9 / 157.6	98.7	190.1 / 188.9	177.8
I2L-MeshNet† [14]	155.5	167.2 / 167.0	99.2	199.0 / 198.1	184.4
HMR* [6]	**109.0**	142.5 / 128.8	**71.8**	164.6 / 150.7	**127.6**
SPEC	116.1	**124.3 / 124.3**	**71.8**	**147.1 / 147.1**	136.4

Table 1: Results of SOTA methods on MTP-Cam dataset. We use the implementations provided by the authors to obtain results. HMR* means that we train HMR using the same data as SPEC for fair comparison. †means we use the SMPL output of this method instead of the non-parametric mesh to be able to report W-PVE. All numbers are in *mm*.

Methods	**MPJPE**	**W-MPJPE**	**PA-MPJPE**	**W-PVE**	**PVE**
GraphCMR [9]	179.8	181.7 / 181.5	86.6	219.8 / 218.3	216.8
SPIN [8]	159.6	165.8 / 161.4	79.5	194.1 / 188.0	186.3
PartialHumans [18]	172.6	169.3 / 174.1	88.2	207.6 / 210.4	209.0
I2L-MeshNet† [14]	161.7	169.8 / 163.3	82.0	203.2 / 195.9	194.5
HMR* [6]	92.8	128.7 / 96.4	55.9	144.2 / 111.8	108.1
SPEC	**74.9**	**74.9 / 74.9**	**54.5**	**90.5 / 90.5**	**90.5**

Table 2: Results of SOTA methods on AGORA-cam. See Table 1 caption.

Methods	**MPJPE**	**W-MPJPE**	**PA-MPJPE**	**W-PVE**	**PVE**
GraphCMR [9]	121.2	137.8 / 129.4	69.1	158.4 / 152.1	139.3
SPIN [8]	96.9	122.2 / 116.6	59.0	140.9 / 135.8	129.8
Partial Humans [18]	126.5	139.4 / 132.9	76.9	160.1 / 152.7	144.5
I2L-MeshNet† [14]	100.0	133.3 / 119.6	60.0	154.5 / 141.2	129.5
HMR* [6]	**92.5**	119.2 / **104.0**	53.7	136.2 / **120.6**	**109.5**
SPEC	96.5	**106.4** / 106.4	**53.2**	**127.4** / 127.4	118.5

Table 3: Results of SOTA methods on 3DPW test set. See Table 1 caption.

Note that MPJPE and PVE are typically computed in the camera space. One needs to transform the ground truth bodies using the camera extrinsics provided by the datasets, and thus the error encodes dataset-specific camera information. However, existing benchmarks, e.g., MPI-INF-3DHP, Human3.6M and 3DPW are often captured with little variation in camera parameters. As a results, MPJPE and PVE cannot clearly reflect the performance of a IWP-cam method for in-the-wild scenarios where camera types and viewpoints are diverse and unknown. We advocate W-MPJPE and W-PVE because they also measure discrepancies in rotation, but unlike MPJPE/PVE, they are computed in world coordinates, assuming no access to camera information. In W-MPJPE and W-PVE, SPEC again outperforms SOTA methods [8, 9, 14, 18] and yield consistently better results than HMR* on datasets captured with diverse camera parameters – SPEC-MTP and SPEC-SYN. Even on 3DPW which is captured with single focal length value, SPEC attains improved or on-par results than HMR*.

3.4. SMPLify-X-cam Results

Table 4 and 5 summarize the performance of SMPLify-X-cam in comparison to the baseline SMPLify-X method. The same 2D keypoint detections from [1] are used for all the reported methods. We compare the default SMPLify-X with $f = 5000$ [17],

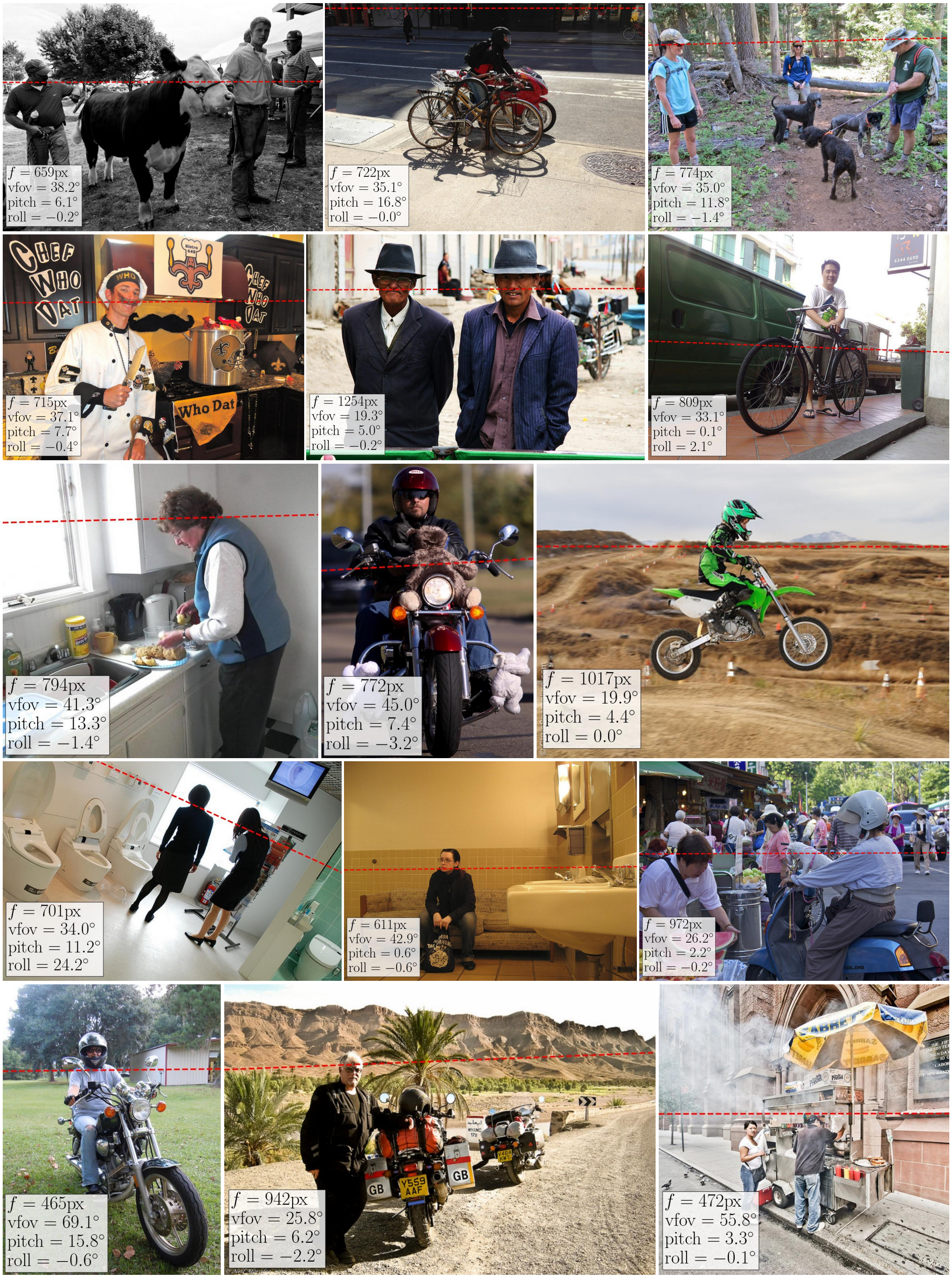


Figure 5: **CamCalib qualitative results.**

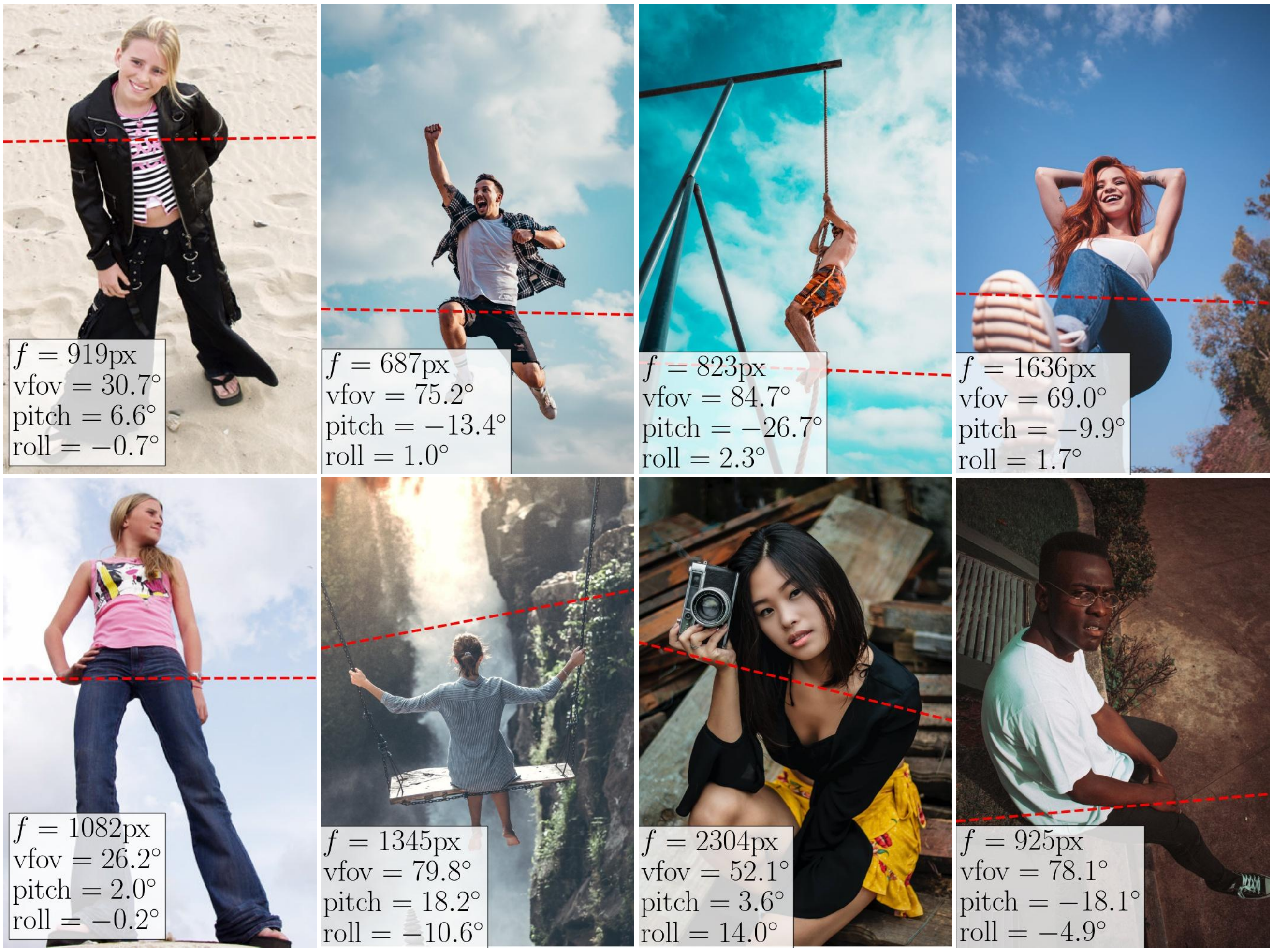


Figure 6: **CamCalib failures.**

the setting considered by Kissos et al. [7] where $f = 2200$, and the setting which uses CamCalib estimated K and R^c. We observe a consistent improvement in W-MPJPE and W-PVE when R^c is used. This is due to correct global orientation reconstruction w.r.t. world coordinates. On SPEC-SYN, using K improves the PA-MPJPE due to more accurate projective geometry. 3DPW is captured with a single camera where $f = 1962$ so it is not a good dataset with which to evaluate the effect of focal length. Even in this case, SMPLify-X-cam improves the results over the default setting and the PA-MPJPE result is on par with results using $f = 2200$. The $f = 2200$ approximation is already close to the single focal length used in 3DPW dataset $f = 1962$, consequently it does well. As a reference, the average CamCalib focal length error is 360 and 246 pixels on 3DPW and SPEC-SYN datasets, respectively.

Methods	**W-MPJPE**	**PA-MPJPE**	**W-PVE**
SMPLify-X (ground-truth f)	131.9 / 114.6	73.3	151.5 / 133.4
SMPLify-X [17] ($f = 5000$)	168.9 / 149.5	77.1	191.5 / 172.8
SMPLify-X [7] ($f = 2200$)	155.6 / 133.5	75.6	176.9 / 155.3
SMPLify-X (CamCalib K)	136.5 / 116.4	**73.0**	156.3 / 135.8
SMPLify-X-cam (CamCalib K + R^c)	**115.2 / 115.2**	73.5	**135.0 / 135.0**

Table 4: **HPS optimization with an estimated camera.** SMPLify-X-cam on the AGORA-cam dataset.

To further analyze the impact of focal length on body reconstructions, we run SMPLify-X on SPEC-SYN with focal lengths perturbed from the real ground truth values and plot the W-MPJPE trend in Fig. 7 (blue curve). We see that the quality of HPS are sensitive to underestimated focal lengths and less sensitive to overestimation, as also reported in [7, 21]. In addition, we visualize default SMPLify-X ($f = 5000$) and SMPLify-X (CamCalib K) according to the corresponding W-MPJPE in Table 4. One can see that in average, with the focal lengths from CamCalib, body reconstructions are closer to the low-error basin of small perturbations, i.e. closer to the true focal lengths. Note that, the averaged focal length in SPEC-SYN is 840 pixels, so the averaged error of 246 pixels (29%) confirms again that SMPLify-X-cam is relatively robust when the estimated focal length ranges between 0.7 and 1.3 times the true one.

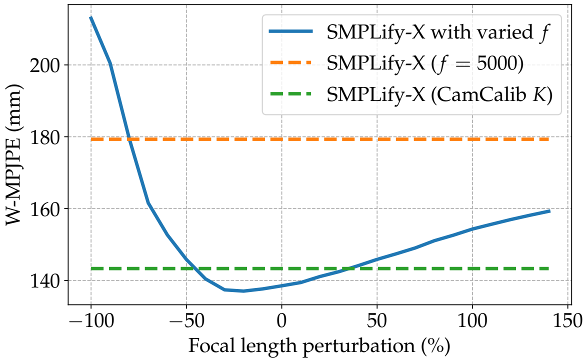


Figure 7: Sensitivity of SMPLify-X to focal length perturbation on SPEC-SYN dataset. Using CamCalib estimated f yields better results. Body reconstruction accuracy is less sensitive to larger focal lengths. Therefore, we propose Softargmax-biased-$\mathcal{L}_2$ loss.

Methods	W-MPJPE	PA-MPJPE	W-PVE
SMPLify-X (ground-truth f)	124.9 / 95.0	55.4	146.9 / 120.0
SMPLify-X [17] ($f = 5000$)	123.8 / 94.7	57.7	144.8 / 119.8
SMPLify-X [7] ($f = 2200$)	124.8 / **94.4**	**55.5**	146.7 / **119.6**
SMPLify-X (CamCalib K)	125.5 / 95.4	55.7	147.4 / 120.3
SMPLify-X-cam (CamCalib K + R^c)	**95.5** / 95.5	55.8	**120.4** / 120.4

Table 5: **HPS optimization with an estimated camera.** SMPLify-X-cam on the 3DPW validation set.

Methods	W-MPJPE	PA-MPJPE	W-PVE
HMR*	112.7 / **97.2**	62.3	133.1 / 115.6
HMR* + c	115.4 / 97.3	62.2	135.9 / 116.9
HMR* + c + f	112.1 / 96.1	61.4	134.1 / **113.6**
HMR* + c + f + R^c	102.7 / 102.7	60.2	124.0 / 124.0
SPEC	**98.1** / 98.1	**59.9**	**119.3** / 119.3

Table 6: Ablation experiments on SPEC with 3DPW validation set. c: using the image center as camera center; f and R^c: using CamCalib estimated focal length and camera rotation, respectively. All numbers are in *mm*.

3.5. Ablation study on SPEC with 3DPW

The Table 5 in the main text provides an ablation study on SPEC using the SPEC-SYN dataset, which dissects the improvement over the baseline HMR* in various aspects: using the original image center as the principal point, using the CamCalib estimated focal length, using the estimated rotations, and lastly conditioning the network with the estimated cameras (SPEC). We repeat this here on the common 3DPW benchmark as in Table 6. Despite that it is not a suitable dataset to analyze the impact of each camera parameters, we still observe that appending the estimated cameras to the image feature leads to improvement in five metrics (c.f. the last two rows), so does using the estimated focal length (HMR* + $c + f$ vs. HMR* + c).

Methods	MPJPE	PA-MPJPE
Want et al. [20]	89.7	65.2
SPEC w. 3DPW	96.4	52.7

Table 7: Comparison to Wang et al. [20]. Here both methods are trained with 3DPW training set for a fair comparison.

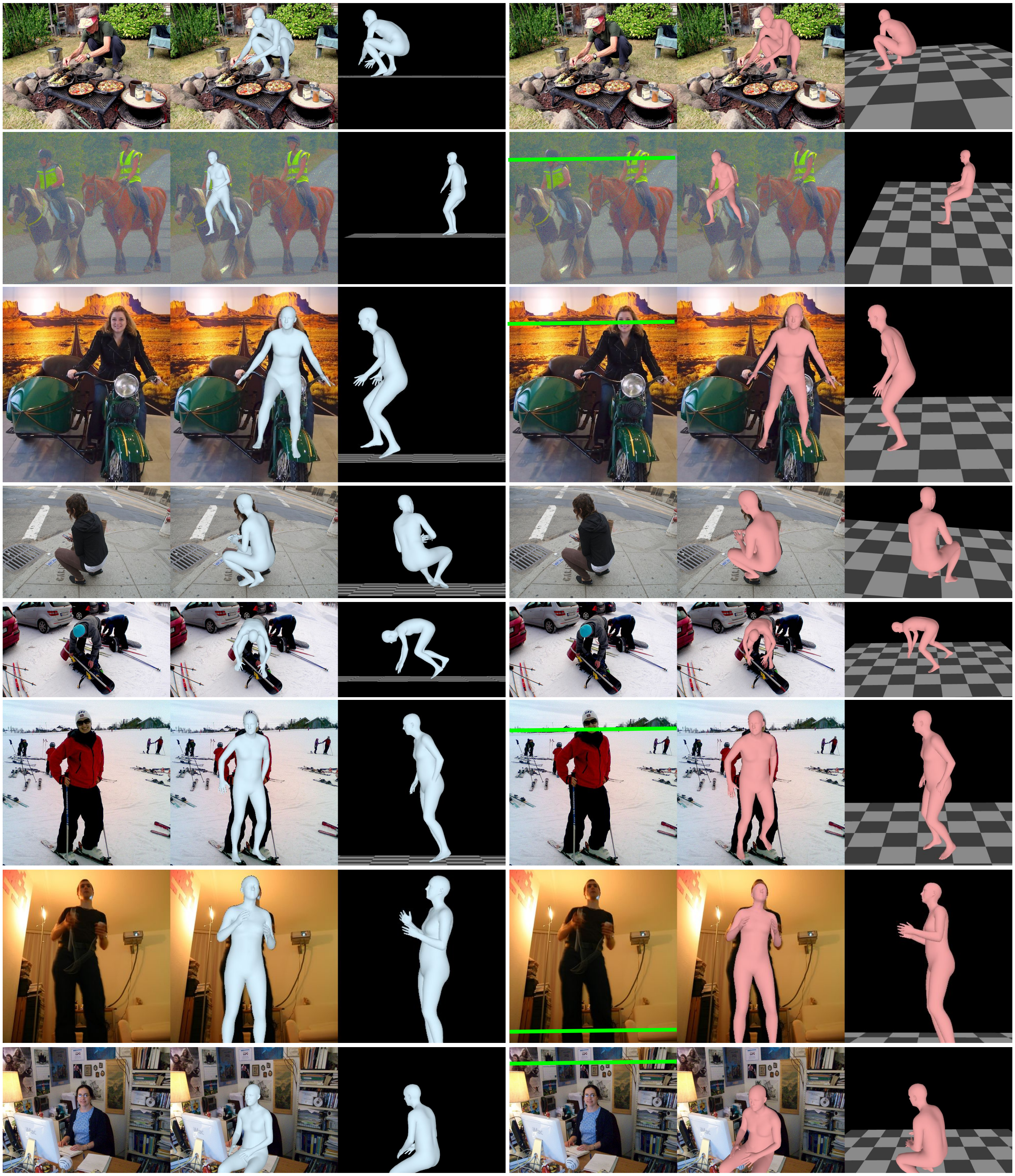
3.6. Comparison to Wang et al. [20]

Wang et al. [20] train their methods on 3DPW training set. We also train SPEC on 3DPW to make a comparison to their method. Results are denoted in Table 7. SPEC outperforms Wang et al. [20] in terms of PA-MPJPE metric, but performs poorly in MPJPE. This is due to the use of estimated camera parameters in SPEC's evaluation. We argue that SPEC would perform better with W-MPJPE metric, however a comparison is not possible since the code of Wang et al. [20] is not available.

3.7. Qualitative results of SPEC

In Fig. 8, we show several qualitative results from SPEC. One can observe that SPEC yields on-par or more physically plausible reconstructed bodies than the baseline that is trained with the identical setting. For more clear illustration, please see the 360° visualizations in Sup. Mat. video.

The failure cases of SPEC are shown in Fig. 9. We observe that some examples share similar traits as those in Fig. 5: portrait images with limited background information. The error of SPEC can be partially attributed to the error from CamCalib. Other scenarios include rarely-seen viewpoints or poses that are not observed in the training data.



(a) Input Image (b) HMR* - front (c) HMR* - side (d) CamCalib (e) SPEC - front (e) SPEC - side

Figure 8: **SPEC qualitative results.**

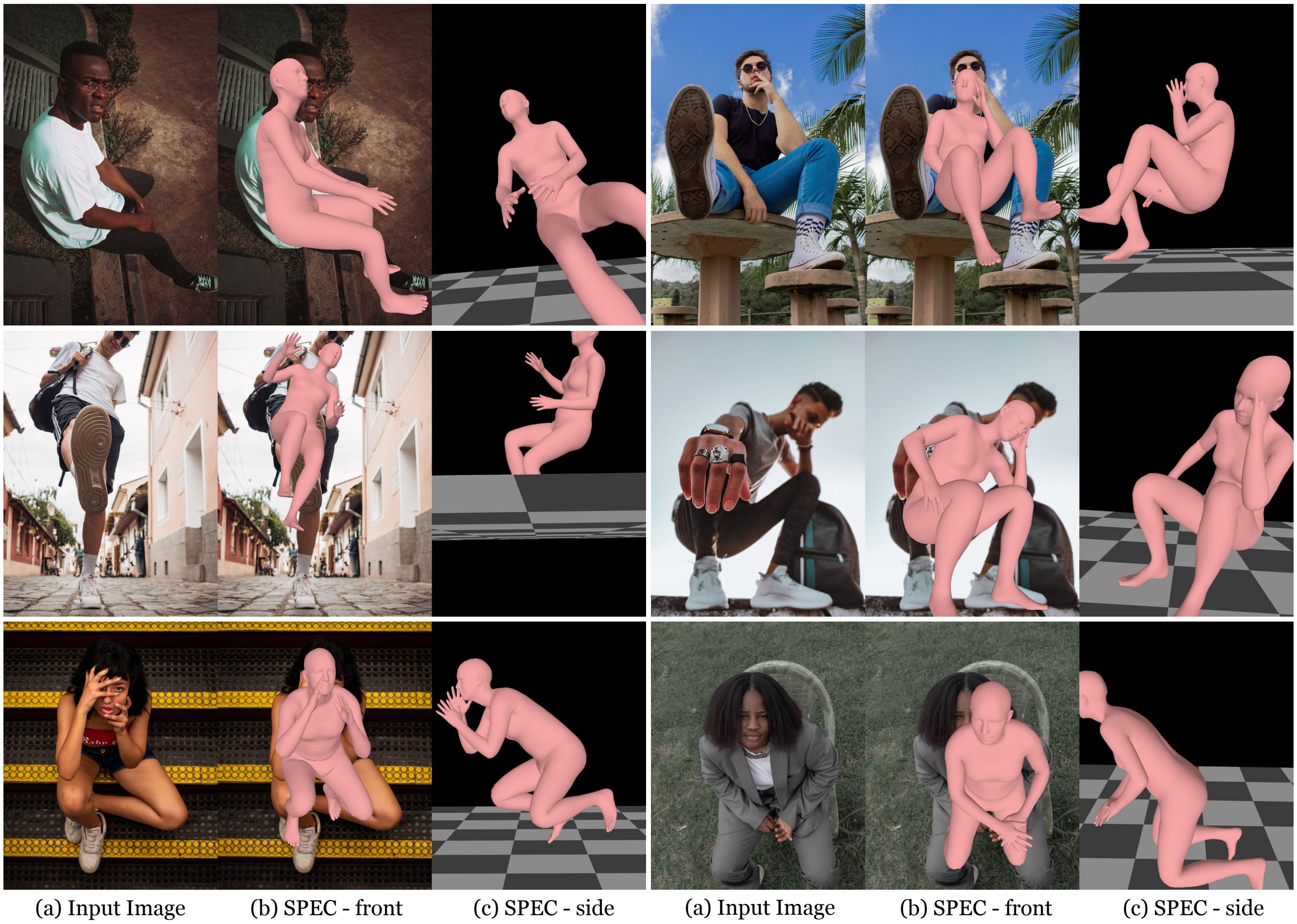


Figure 9: **SPEC failures.**

References

[1] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 6

[2] Epic Games. Unreal engine. https://www.unrealengine.com, 2021. 2

[3] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. In *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2014. 6

[4] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[5] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. *arXiv preprint arXiv:2004.03686*, 2020. 2, 6

[6] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 6

[7] Imry Kissos, Lior Fritz, Matan Goldman, Omer Meir, Eduard Oks, and Mark Kliger. Beyond weak perspective for monocular 3d human pose estimation. In Adrien Bartoli and Andrea Fusiello, editors, *European Conference on Computer Vision*, pages 541–554, Cham, 2020. Springer International Publishing. 1, 8, 9

[8] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *International Conference on Computer Vision*, 2019. 6

[9] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[10] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4521–4530, 2019. 2

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 6

[12] 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. 1

[13] 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 *International Conference on 3DVision*, 2017. 6

[14] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, 2020. 6

[15] Lea Müller, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black. On self contact and human pose. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[16] Priyanka Patel, Chun-Hao P. Huang, Joachim Tesch, David T. Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in geography optimized for regression analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2

[17] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 6, 8, 9

[18] Chris Rockwell and David F. Fouhey. Full-body awareness from partial observations. In *European Conference on Computer Vision*, 2020. 6

[19] 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. 1

[20] Zhe Wang, Daeyun Shin, and Charless C Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *European Conference on Computer Vision*, 2020. 9

[21] Frank Yu, Mathieu Salzmann, Pascal Fua, and Helge Rhodin. PCLs: Geometry-aware neural reconstruction of 3d pose with perspective crop layers. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 8

[22] Rui Zhu, Xingyi Yang, Yannick Hold-Geoffroy, Federico Perazzi, Jonathan Eisenmann, Kalyan Sunkavalli, and Manmohan Chandraker. Single view metrology in the wild. In *European Conference on Computer Vision*, 2020. 1, 2, 6