ReFit: Recurrent Fitting Network for 3D Human Recovery Supplementary Material

Yufu Wang Kostas Daniilidis
University of Pennsylvania

This Supplementary Material includes additional details about the camera model, the network architecture, the training implementation and datasets, failure analysis, and more visualization of examples.

A. Method

A.1. Full-frame Adjusted Camera Model.

We properly adjust the reprojection of keypoints and joints to be consistent with the full-frame image formation process, for supervision when we reproject 3D joints and for feedback when we reproject keypoints (e.g., mocap markers).

Following previous work [8, 12], given a full-frame image of size (w, h) and a square bounding box of a human with center (c_x, c_y) and size b, we obtain a square-cropped image as input to the network. The network predicts a weak perspective camera $\pi = (s, t_x, t_y)$, which is converted to translation t^{full} with respect to the original camera center as

$$t_x^{full} = t_x + \frac{2(c_x - w/2)}{s \cdot b}$$
$$t_y^{full} = t_y + \frac{2(c_y - h/2)}{s \cdot b}$$
$$t_z^{full} = \frac{2 \cdot f}{s \cdot b}$$

where f is the focal length. When ground truth focal length is not available, we estimate the focal length as $f \approx \sqrt{w^2 + h^2}$ following Kissos et al. [10].

The 3D joints or 3D keypoints can be reprojected to the full-frame image using t^{full} as

$$x_{2D}^{full} = \Pi(X_{3D}^{full}) = \Pi(X_{3D} + t^{full}) \quad (1)$$

We further adjust the points back to the cropped image after full-frame reprojection with

$$x_{2D}^{crop} = (x_{2D}^{full} - c^{bbox})/s^{bbox} \quad (2)$$

where $c^{bbox} = (c_x, c_y)$ and $s^{bbox} = b$. With this camera model, the reprojections are in the crop image space, which allows us to calculate normalized reprojection error and query features from feature maps. At the same time, the reprojections are consistent with the image formation of the original full-frame image, leading to a better global rotation estimation [13].

A.2. Architecture.

We present the practical implementation of ReFit in Figure 1. We use HRNet-48 [18] as the backbone. The feature extractor outputs feature maps F and a global feature vector f^0. Instead of directly predicting an initialization Θ_0 from f^0, we reuse the update module. f^0 is concatenated with Θ_{mean} and $bbox$, and fed to the update module to predict the initialization: $\Theta_{mean} + \Delta\Theta_{mean} \rightarrow \Theta_0$.

In the feedback step, each keypoint reprojection queries a window of features. A radius of 3 (r=3) corresponds to a 7×7 window. We use bilinear interpolation to compute the entries of the window.

Each 7×7 window is first flattened and fed to a linear layer to produce a shorter feature vector of size 5, before concatenated with features from other queries to form a feedback vector of size 5K, where K is the number of keypoints. Another linear layer reduces the feedback vector from 5K to 256. These two linear layers in the feedback module prevent the feedback vector from growing too long and reduce the parameters needed in the update module. The feedback-update iterations proceed as described in the main text.

The update module outputs the 6D rotation representation [23] for the pose parameters, which are converted to rotation matrices for the SMPL model.

B. Experiments

B.1. Training Details.

Implementation. We implement ReFit in PyTorch. The input cropped image is resized to 256×256. We use the Adam [9] optimizer with a learning rate of 1e-4 and a batch size of 64. Following PARE [11], we apply standard augmentations, including random rotation and scaling, color jittering, and synthetic occlusion using objects from PASCAL [3].

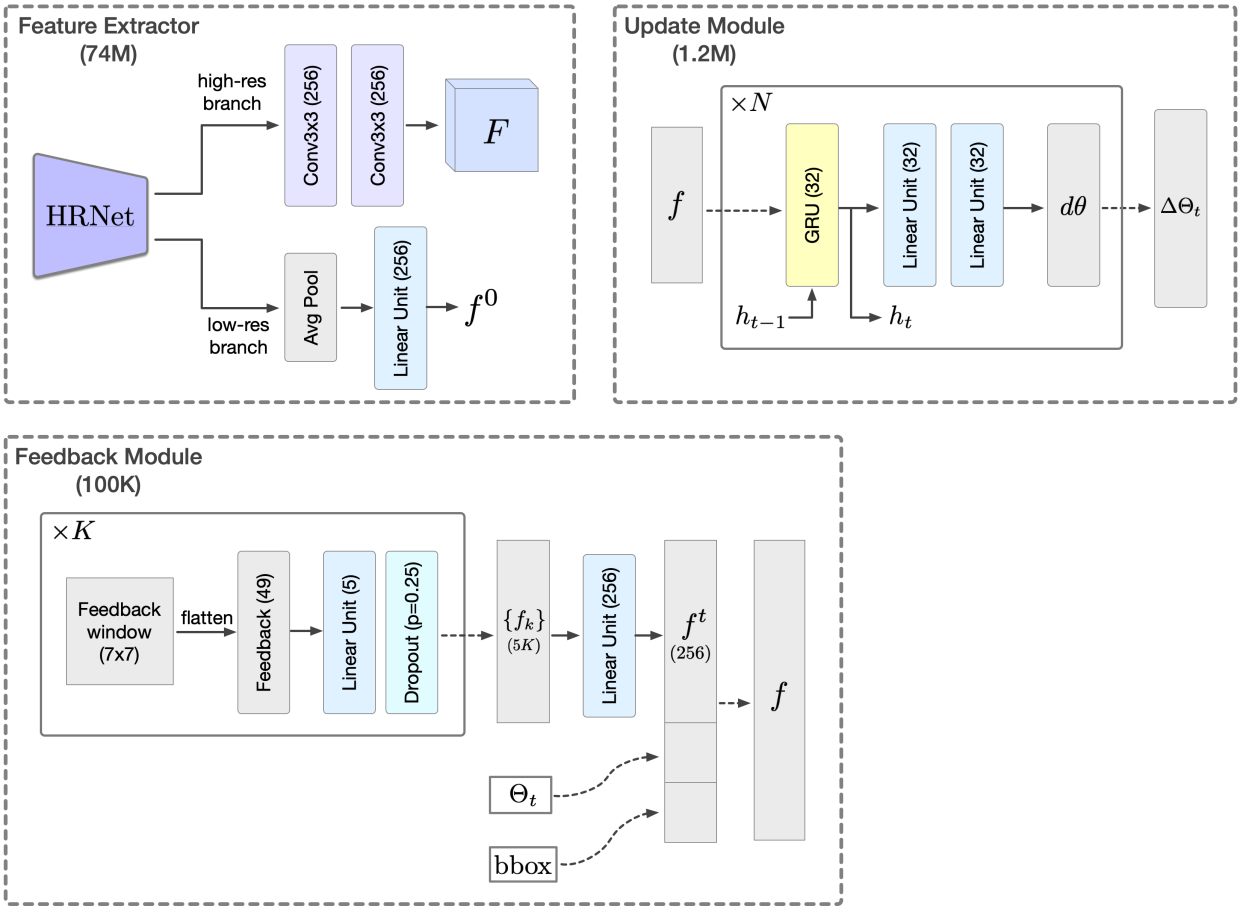

Figure 1. **ReFit Implementation.** The feature extractor takes up most of the parameters while the feedback-update modules are lightweight. We use two linear layers in the feedback module to reduce the size the feedback vector, which in turn reduces the parameter count of the GRUs in the update module.

Training schedule. Training procedure and schedule impact the final performance. While different architectures can benefit from different training schedules [11, 12, 22], this inevitably makes it harder to reproduce the experiments.

Furthermore, recent methods [13, 14] train separate models to evaluate on 3DPW and Human3.6M because of the difference in data distribution. 3DPW consists of in-the-wild images, while Human3.6M contains only images captured in a mocap studio. So training with fewer in-the-wild images can improve performance on Human3.6M but decrease performance on 3DPW.

We use a simple schedule. The backbone network is initialized from the COCO pose detection task following prior works [11, 21], and we do not use other forms of pre-training. Instead, we directly train with all the datasets to produce a generic model, which is evaluated on 3DPW. Pose recovery in a mocap studio can be considered a special application domain, so we finetune the generic model on Human3.6M only and evaluate it on Human3.6M. The training of the generic model lasts for 50k iterations, and fine-tuning on Human3.6M uses another 50k iterations. Overall, this is a much shorter schedule compared to previous studies.

B.2. Datasets.

We provide more information about the datasets for training and evaluation. Overall, we closely follow prior works [7, 11]. We train the ReFit model with 3DPW [19], Human3.6M [6], MPI-INF-3DHP [17], COCO [15] and MPII [1]. We evaluate on 3DPW and Human3.6M.

Following prior works [7, 12], during training each batch is sampled from different datasets with the following ratio: [Human3.6M: 40%, 3DPW: 20%, MPI-INF-3DHP: 10%, COCO: 25%, and MPII: 5%].

To evaluate the 3D pose accuracy, we use the Mean Per-Joint Position Error (MPJPE), which computes the average Euclidean distance between the ground truth and the predicted joints after aligning the pelvis. The Procrustes-aligned MPJPE (PA-MPJPE) further performs general Procrustes to align the ground truth and the predicted joints before computing the position error.

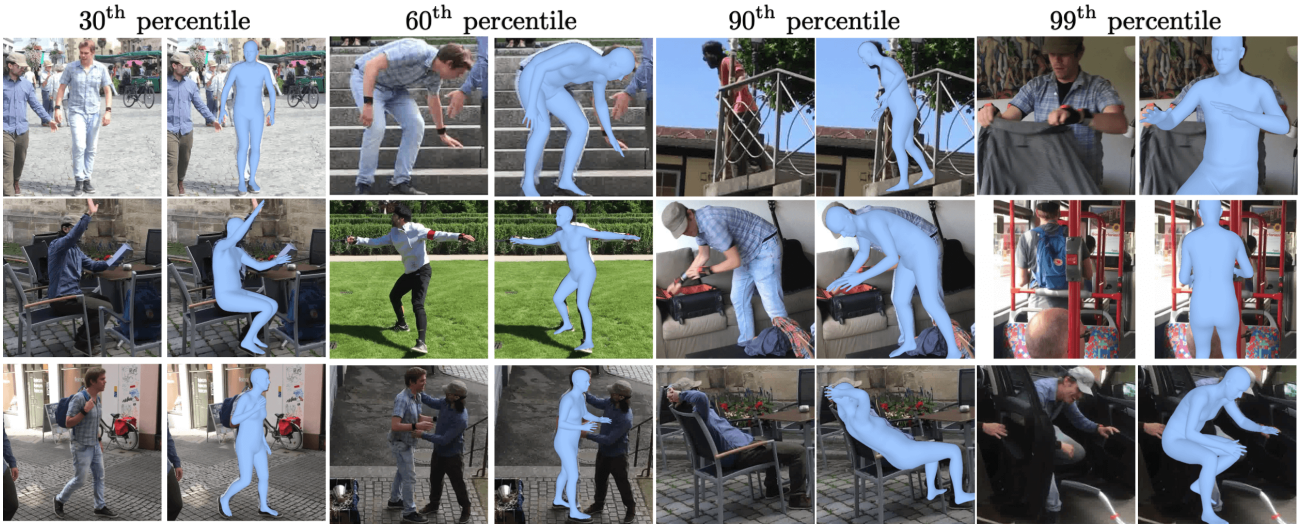

Figure 2. **More results from 3DPW.** Examples are grouped by MPJPE percentiles, with a higher percentile indicating a higher error. The MPJPE is 50.5mm, 65.4mm, 99.3mm, and 158.8mm for the four percentiles respectively.

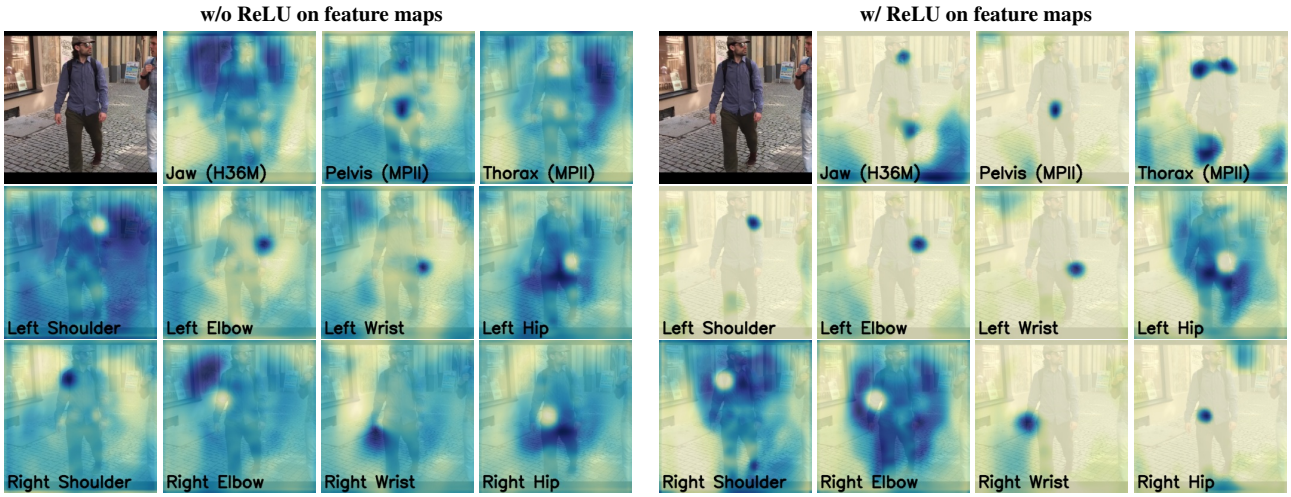

Figure 3. **Visualization of feature maps**. We show the feature maps from the semantic keypoint model. Model trained with a ReLU operator on the feature maps produces "cleaner" features but slightly lower benchmark performance, likely because the strictly positive features are less expressive.

Human3.6M is a multi-view, indoor-captured 3D human pose dataset. It includes 2D and 3D joint annotations of several subjects performing various actions. In addition, we use the SMPL parameters recovered using MoSH [16], provided by Kanazawa et al. [8], as additional supervision. Following prior works, we use subjects S1, S5, S6, S7, and S8 for training, and use S9 and S11 for evaluation. Furthermore, we evaluate Multi-view ReFit on S9 and S11 using the calibrated multi-view images.

MPI-INF-3DHP is an indoor multi-view dataset. It provides 2D and 3D joint annotations, but the 3D joints are recovered in a markerless setting. Additionally, we use the SMPL parameters provided by Kolotouros et al. [12], which is from multi-view fitting.

3DPW is an in-the-wild dataset providing 2D joints, 3D joints and SMPL parameters annotations. The SMPL parameters are recovered from IMU sensing and 2D videos. We use this dataset for training and evaluation. Moreover, we perform ablations on it because it contains diverse settings most relevant for the target applications.

COCO is a large object recognition dataset that also provides 2D keypoint annotations for human subjects in the wild. Additionally, Joo et al. [7] proposes EFT to recover pseudo-ground truth SMPL parameters for this dataset.

MPII is a human pose dataset providing 2D keypoint annotations of humans in the wild. We also use the pseudo-ground truth SMPL parameters from EFT for this dataset.

BEDLAM [2] is a new synthetic dataset that include

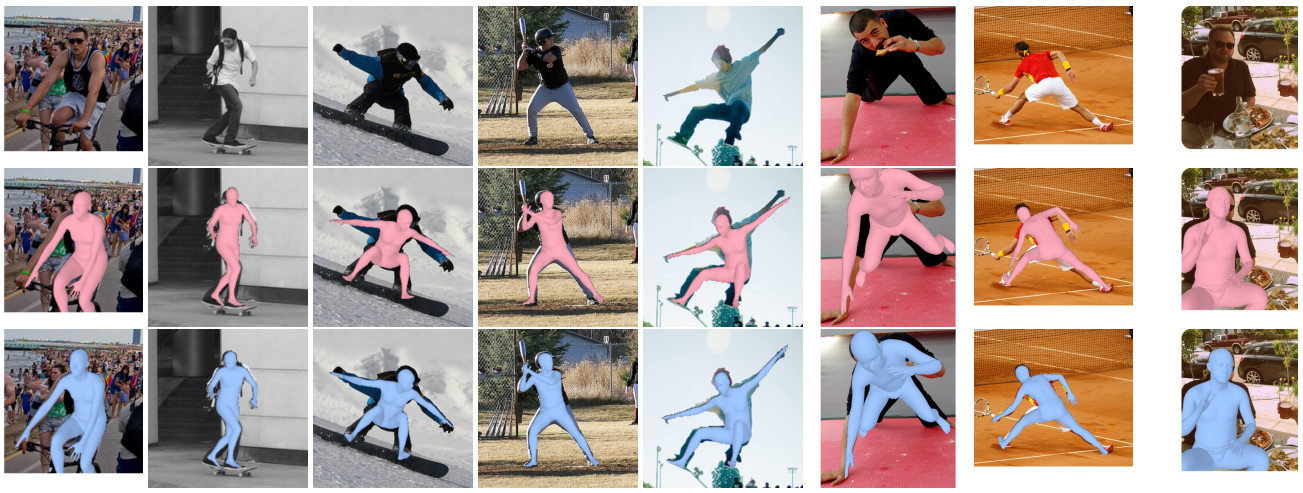

Figure 4. **Qualitative Results** from COCO. The top row shows the input images, the middle row shows the initial estimate without iterative updates (T=0), and the last row shows results from ReFit (T=5).

Figure 5. **Failure examples** from 3DPW. They typically contain severe occlusions, interactions with another human, and blurry or ambiguous scenarios.

ground truth 3D pose and shape. We use this dataset to test the generalization. When training with BEDLAM, we also include a 3D vertex loss following BEDLAM-CLIFF, the baseline for this dataset.

B.3. Results and Visualization.

More examples. We show more examples from 3DPW from different error percentiles in Figure 2. We observe good alignment to the images even in the 99$^{\text{th}}$ percentile, which has a higher error than 99% of the 3DPW test samples.

We include more in-the-wild examples from COCO in Figure 4, which shows initial estimations without iterative updates (T=0) and results from ReFit (T=5).

Feature map visualization. We visualize the feature maps from the semantic keypoint model in Figure 3. The main model that produces state-of-the-art results has no ReLU operator on the feature maps, which allows both positive and negative values. We train an alternative model with ReLU on the feature maps. This alternative model produces slightly lower benchmark results, but the features are "cleaner" as they are strictly positive.

Nevertheless, the feature maps capture meaningful features around the corresponding keypoints, often appearing as peaks or valleys. This paper does not explore other auxiliary supervision or regularization, but other studies have indicated the benefit of auxiliary intermediate supervision [22]. Combining the feature maps with different outputs, such as explicit keypoint detection [20], can also be explored in future work.

Failure examples. We show examples at the 99.9$^{\text{th}}$ error percentile from 3DPW in Figure 5. These examples are deemed failures. They typically contain severe occlusions, close interaction with another human, or unclear images due to far distance or low lighting. In some examples, we also observe left-right flips of the reconstruction.

Our state-of-the-art model is not trained with extreme crop augmentation, but we believe such augmentation can improve cropped cases in real-world applications [7, 11]. Left-right flipping can be addressed with video input and a temporal prior [5]. Close interaction with another human is a fruitful direction for future research [4].

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 2

[2] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8726–8737, 2023. 3

[3] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015. 1

[4] Mihai Fieraru, Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, Vlad Olaru, and Cristian Sminchisescu. Three-dimensional reconstruction of human interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7214–7223, 2020. 4

[5] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape and pose estimation over time. In *2017 international conference on 3D vision (3DV)*, pages 421–430. IEEE, 2017. 4

[6] 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, 2013. 2

[7] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2021. 2, 3, 4

[8] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 1, 3

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[10] Imry Kissos, Lior Fritz, Matan Goldman, Omer Meir, Eduard Oks, and Mark Kliger. Beyond weak perspective for monocular 3d human pose estimation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 541–554. Springer, 2020. 1

[11] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11127–11137, 2021. 1, 2, 4

[12] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019. 1, 2, 3

[13] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 590–606. Springer, 2022. 1, 2

[14] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12939–12948, 2021. 2

[15] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[16] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 3

[17] 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 *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017. 2

[18] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 1

[19] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018. 2

[20] Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Neural descent for visual 3d human pose and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14484–14493, 2021. 4

[21] Hongwen Zhang, Yating Tian, Yuxiang Zhang, Mengcheng Li, Liang An, Zhenan Sun, and Yebin Liu. Pymaf-x: Towards well-aligned full-body model regression from monocular images. *arXiv preprint arXiv:2207.06400*, 2022. 2

[22] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11446–11456, 2021. 2, 4

[23] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 1