PyMAF: 3D Human Pose and Shape Regression with Pyramidal Mesh Alignment Feedback Loop **Supplementary Material**

Hongwen Zhang[§‡*], Yating Tian[†*], Xinchi Zhou[♮], Wanli Ouyang[♮], Yebin Liu[‡], Limin Wang[†], Zhenan Sun[§]
[§]CRIPAC, NLPR, Institute of Automation, Chinese Academy of Sciences, China
[†]State Key Laboratory for Novel Software Technology, Nanjing University, China
[♮]The University of Sydney, Australia [‡]Department of Automation, Tsinghua University, China

This Supplementary Material provides additional details of our approach and more experimental results that were not included in the main manuscript due to space constraints. In Section S1, we provide more details of our experiments and the implementation of our approach. In Sections S2 and S3, we give the descriptions of the datasets and evaluation metrics used in our experiments. Finally, we include more quantitative and qualitative results in Section S4. We also make available the code and video results at the project page https://hongwenzhang.github.io/pymaf.

S1. More Experimental Details

Our network is trained with the Adam [8] optimizer and batch size of 64. The learning rate is set to $5e{-}5$ without learning rate decay during training. Similar to SPIN [9], our network is first trained on Human3.6M for 60 epochs and then on the mixture of both 2D and 3D datasets for another 60 epochs.

The parameter regressors of PyMAF have the same design with that of HMR [6] except for their slightly different input and output dimensions. Specifically, a regressor consists of two fully-connected layers each with 1024 hidden neurons and dropout added in between, followed by a final layer at the end with 157-dimension output, corresponding to the residual of shape and pose parameters. The regressors in our network adopt the continuous representation [20] for 3D rotations in the pose parameters $\boldsymbol{\theta}$. During the extraction of mesh-aligned features, the dimension of point-wise features is reduced from C_s (*i.e.*, 256) to 5, where a three-layer MLP consisting of two hidden layers with neuron numbers of (128, 64) is used. The feature pyramid of PyMAF is generated by three deconvolution layers. The deconvolutions are not compulsory but help to produce better features maps. In our experiments, using the feature maps in the earlier layers is also feasible but inferior to our final solution.

*: Equal contribution.

Runtime. The PyTorch implementation of PyMAF takes about 30 ms to process one sample on the machine with a single 2080 Ti GPU. The proposed mesh alignment feedback loop takes about 6 ms for each iteration, including the time of generating new feature maps via deconvolution, projecting the mesh on image planes, the extraction of mesh-aligned features via bilinear sampling and MLPs, and the prediction of parameter updates by the regressor. For each iteration, compared to the feedback loop in HMR [6] or SPIN [9], PyMAF introduces additional runtime in the generation of feature maps, the current SMPL meshes, and the mesh-aligned features, which accounts for 0.3 ms, 4 ms, and 1.2 ms respectively. We can see that the generation of the feature pyramid and mesh-aligned features is quite efficient, and the main runtime overhead comes from the SMPL mesh generation given the current parameters. In practice, we can speed up this process by using a down-sampled version of SMPL to generate the mesh with 431 vertices directly. Note that the prediction of dense correspondences and the auxiliary supervision in the pipeline are needed for training only, which accounts for additional 15% runtime.

S2. Datasets

Following the protocols of previous work [6, 9], we train our network on several datasets with 3D or 2D annotations, including Human3.6M [3], MPI-INF-3DHP [13], LSP [4], LSP-Extended [5], MPII [1], COCO [11]. For the last five datasets, we also incorporate the SMPL parameters fitted in [2, 9] as pseudo groud-truth annotations for training. Here, we provide more descriptions of the datasets to supplement the main manuscript.

3DPW [17] is captured in challenging outdoor scenes with IMU-equipped actors under various activities. This dataset provides accurate shape and pose ground truth annotations. Following the protocol of previous work [7, 9], we do not use its data for training but only perform evaluations on its test set.

Human3.6M [3] is commonly used as the benchmark dataset for 3D human pose estimation, consisting of 3.6 million video frames captured in the controlled environment. The ground truth SMPL parameters in Human3.6M are generated by applying MoSh [12] to the sparse 3D MoCap marker data, as done in Kanazawa *et al*. [6]. It is common protocols [14, 15, 6] to use five subjects (S1, S5, S6, S7, S8) for training and two subjects (S9, S11) for evaluation. The original videos are also down-sampled from 50 fps to 10 fps to remove redundant frames, resulting in 312,188 frames for training and 26,859 frames for evaluation.

MPI-INF-3DHP [13] is a recently introduced 3D human pose dataset covering more actor subjects and poses than Human3.6M. The images of this dataset were collected under both indoor and outdoor scenes, and the 3D annotations were captured by a multi-camera marker-less MoCap system. Hence, there are some noise in the 3D ground truth annotations.

LSP-Extended [5] is a 2D human pose benchmark dataset, containing person images with challenging poses. There are 14 visible 2D keypoint locations annotated for each image and 9,428 samples used for training.

LSP [4] is a standard benchmark dataset for 2D human pose estimation. In our experiments, we will employ its test set for silhouette/parts segmentation evaluation, where the annotations come from Lassner *et al*. [10]. There are 1,000 samples used for evaluation.

MPII [1] is a standard benchmark for 2D human pose estimation. There are 25,000 images collected from YouTube videos covering a wide range of activities. We discard those images without complete keypoint annotations, producing 14,810 samples for training.

COCO [11] contains a large scale of person images labeled with 17 keypoints. In our experiments, we only use those persons annotated with at least 12 keypoints, resulting in 28,344 samples for training. Since this dataset do not contain ground-truth meshes, we conduct quantitative evaluation on the 2D keypoint localization task using its validation set, which consists of 50,197 samples. Following [18], we crop input images using the ground-truth bounding boxes.

S3. Evaluation Metrics

In the main manuscript, we report results of our approach in a variety of evaluation metrics for quantitative comparisons with the state of the art, where all metrics are computed in the same way as previous work [6, 15, 9] in the literature.

To quantitatively evaluate the 3D human reconstruction and pose estimation performance on 3DPW and Human3.6M, PVE, MPJPE, and PA-MPJPE are adopted as the evaluation metrics in Table 1 of the main manuscript. They

(a) PA-MPJPE: 26.9, MPJPE: 74.3 (b) PA-MPJPE: 27.7, MPJPE: 43.4

Figure S1: Examples of two reconstruction results. (a) A reconstruction result with a lower PA-MPJPE value but worse mesh-image alignment. (b) A reconstruction result with a higher PA-MPJPE value but better mesh-image alignment.

are all reported in millimeters (mm) by default. Among these three metrics, PVE denotes the mean Per-vertex Error defined as the average point-to-point Euclidean distance between the predicted mesh vertices and the ground truth mesh vertices. MPJPE denotes the Mean Per Joint Position Error, and PA-MPJPE denotes MPJPE after rigid alignment of the prediction with ground truth using Procrustes Analysis. Note that the metric PA-MPJPE can not fully reveal the mesh-image alignment performance since it is calculated as MPJPE after rigid alignment. As depicted in Fig. S1, a reconstruction result with a lower PA-MPJPE value can have a higher MPJPE value and worse alignment between the reprojected mesh and image.

In Table 2 of the main manuscript, segmentation accuracy metrics quantitatively measure the mesh-image alignment of different approaches on the LSP dataset. As originally done in Lassner *et al*. [10], silhouette (*i.e*., Foreground/Background, FB) and Part segmentation are considered in calculating the accuracy and f1 scores.

For 2D human pose estimation task on COCO[1], the commonly-used Average Precision (AP) is adopted as the evaluation metric. AP is calculated based on the Object Keypoint Similarity (OKS), which plays a similar role as IoU in object detection. In Table 3 of the main manuscript, the results are reported using mean AP, and variants of AP including AP_{50} (AP at OKS = 0.50), AP_{75} (AP at OKS = 0.75), AP_M for persons with medium sizes, and AP_L for persons with large sizes.

S4. More Experimental Results

More Quantitative Results. To evaluate the performances of PyMAF on human images with occlusions and different body shape styles, we conduct evaluation experiments on 3DOH50K [19] and SSP-3D [16] datasets. The test set of 3DOH50K includes 1,290 person images in occlusion scenarios, while SSP-3D consists of 311 images of sport persons with a variety of body shapes and poses.

[1] https://cocodataset.org/#keypoints-eval

	3DOH50K		SSP-3D
	PVE↓	MPJPE↓	mIOU↑
SPIN [9]	113.4	102.3	70.2
Baseline	113.1	102.0	70.8
PyMAF	**107.3**	**96.2**	**72.1**

Table S1: Reconstruction performances on 3DOH50K and SSP-3D datasets.

Figure S2: Reconstruction results of PyMAF on the SSP-3D [16] dataset. PyMAF fails to handle extreme shapes due to the lack of training data.

Note that we only perform testing on these two datasets and do not use their data for training. Experimental results on 3DOH50K and SSP-3D are reported in Tab. S1, and qualitative results are shown in Figures S3 and S2. PyMAF can improve the reconstruction under occlusions on 3DOH50K and help with more accurate shape estimation on SSP-3D. Despite the numerical performance gains, PyMAF fails to handle extreme shapes on the SSP-3D dataset, as shown in Figure S2.

More Qualitative Results. We provide more qualitative results and compare our PyMAF with the state-of-the-art approach SPIN [9]. Figure S4 shows the qualitative differences between each iterative loop in SPIN [9] and PyMAF, which uses the global features and spatial features for the parameter update respectively. We can see that PyMAF convergences much faster and corrects the mesh parameters more effectively. In Figure S5, we show more reconstruction results for qualitative comparisons with SPIN on both indoor and in-the-wild datasets, where PyMAF can produce natural results which are better-aligned with the images under challenging cases. Note that our approach is complementary to SPIN, since SPIN aims at providing better supervision for the regression network, while our work focuses on the architecture design of the regression network.

To demonstrate the efficacy of the parameter rectification in PyMAF, we further provide more examples on COCO and 3DPW in Figures S6 and S7, respectively. We can observe that PyMAF improves the mesh-image alignment progressively by correcting the predictions based on the current observations. We also visualize some erroneous results of our approach in Figure S8, where PyMAF may fail when the initial reconstructed results have severe deviations due to the heavy occlusions or ambiguous limb connections in complex scenes.

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. 1, 2

[2] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Proceedings of the European Conference on Computer Vision*, pages 561–578. Springer, 2016. 1

[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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. 1, 2

[4] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *British Machine Vision Conference*, volume 2, page 5, 2010. 1, 2

[5] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1465–1472. IEEE, 2011. 1, 2

[6] 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, 2

[7] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5614–5623, 2019. 1

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014. 1

[9] 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 International Conference on Computer Vision*, pages 2252–2261, 2019. 1, 2, 3, 4, 5

[10] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6050–6059, 2017. 2

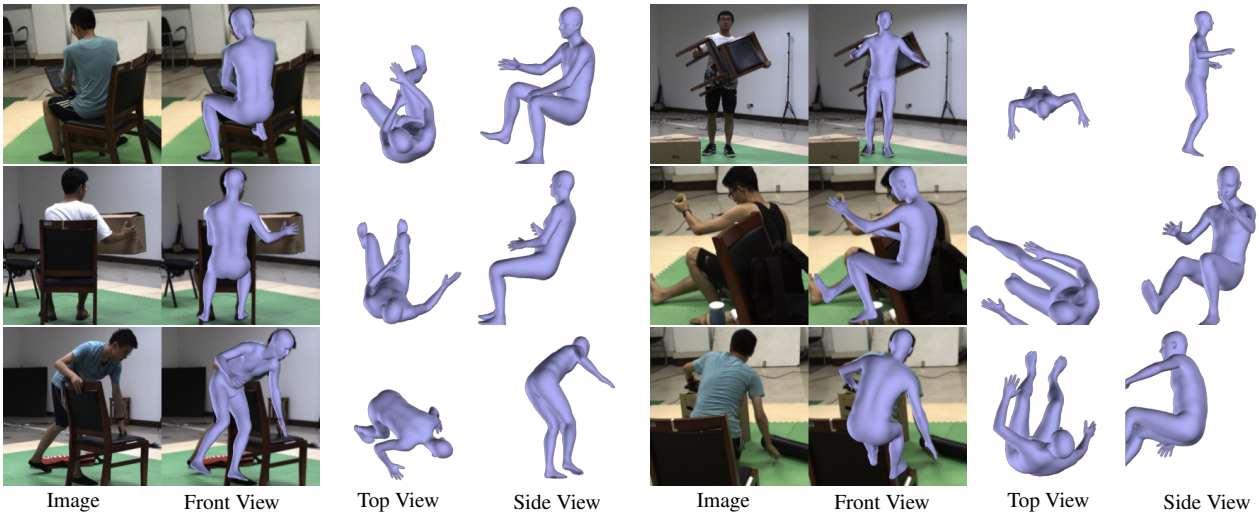

Figure S3: Reconstruction results of PyMAF on the 3DOH50K [19] dataset. PyMAF helps to handle occlusions.

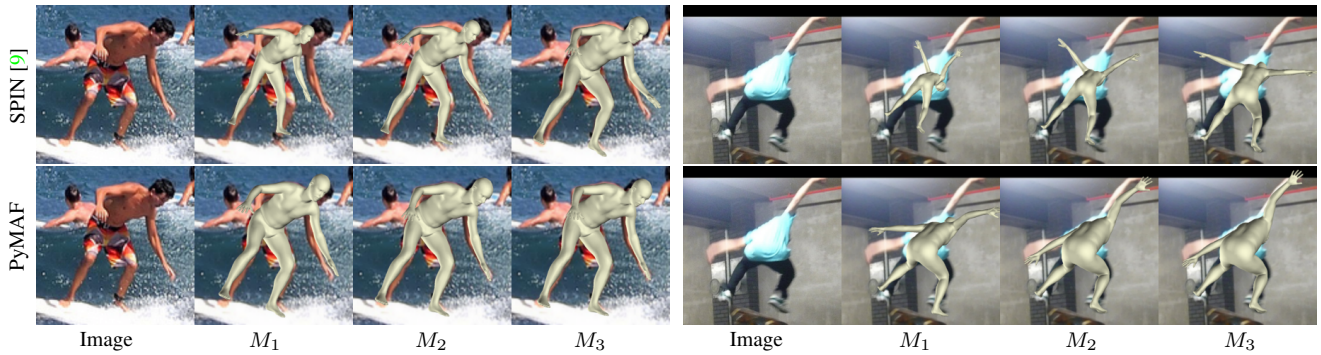

Figure S4: Qualitative differences between each iterative loop of the SPIN [9] using global features *vs*. the PyMAF using spatial features.

[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 *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1, 2

[12] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics*, 33(6):220, 2014. 2

[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 3D Vision*, pages 506–516. IEEE, 2017. 1, 2

[14] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017. 2

[15] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018. 2

[16] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Synthetic training for accurate 3d human pose and shape estimation in the wild. In *British Machine Vision Conference*, 2020. 2, 3

[17] 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*, pages 601–617, 2018. 1

[18] Hongwen Zhang, Jie Cao, Guo Lu, Wanli Ouyang, and Zhenan Sun. Learning 3d human shape and pose from dense body parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[19] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2020. 2, 4

[20] 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 Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 1

Figure S5: Qualitative comparison of the reconstruction results between SPIN [9] and our PyMAF approach. For each example, the upper / lower results correspond to the reconstructed meshes of SPIN (pink) / PyMAF (purple). Examples come from various datasets, including COCO (Rows 1-3), 3DPW (Row 4), and Human3.6M (Row 5).

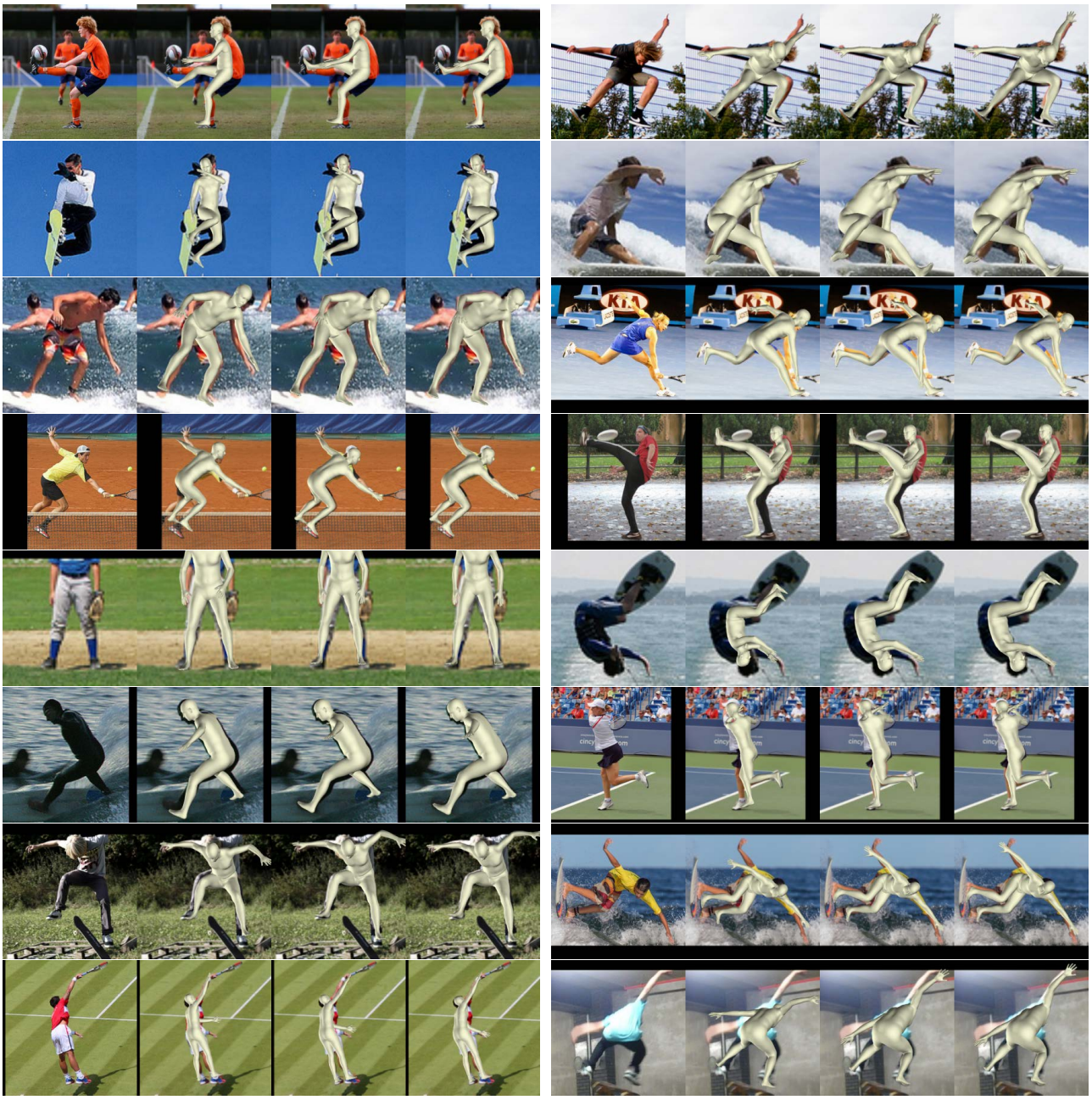

Figure S6: Successful results of PyMAF on the COCO dataset. For each example from left to right: image, the results after each iteration.

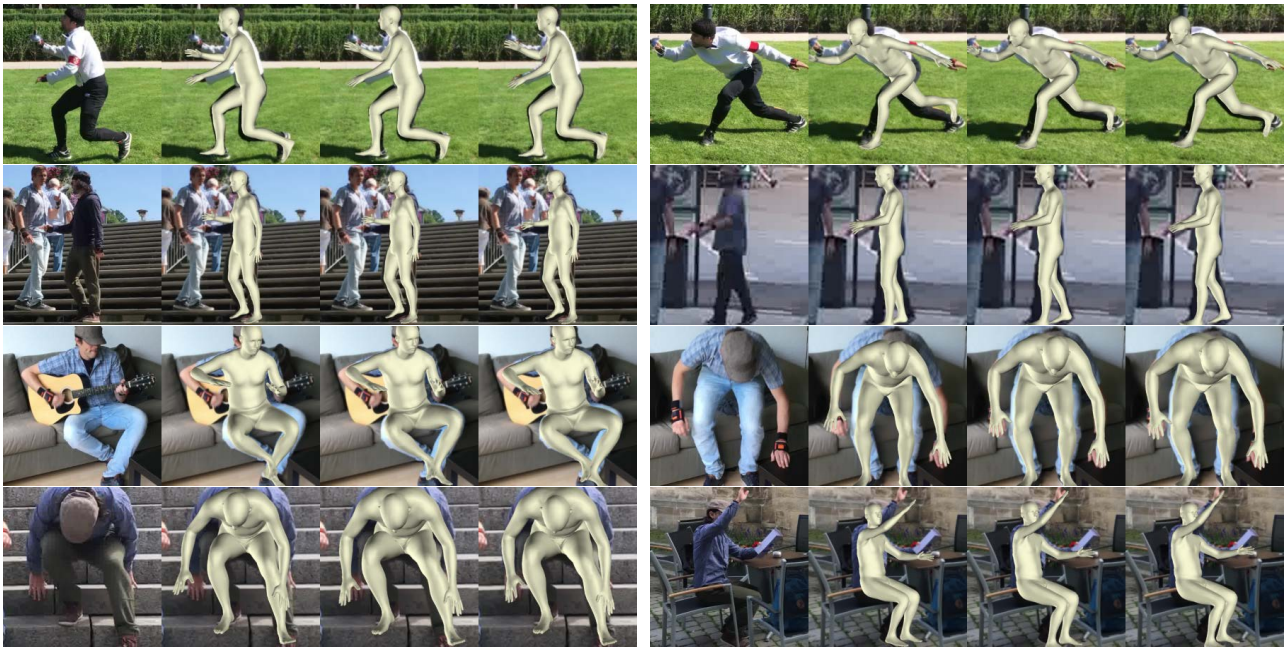

Figure S7: Successful results of PyMAF on the 3DPW dataset. Examples have the same layout with Figure S6.

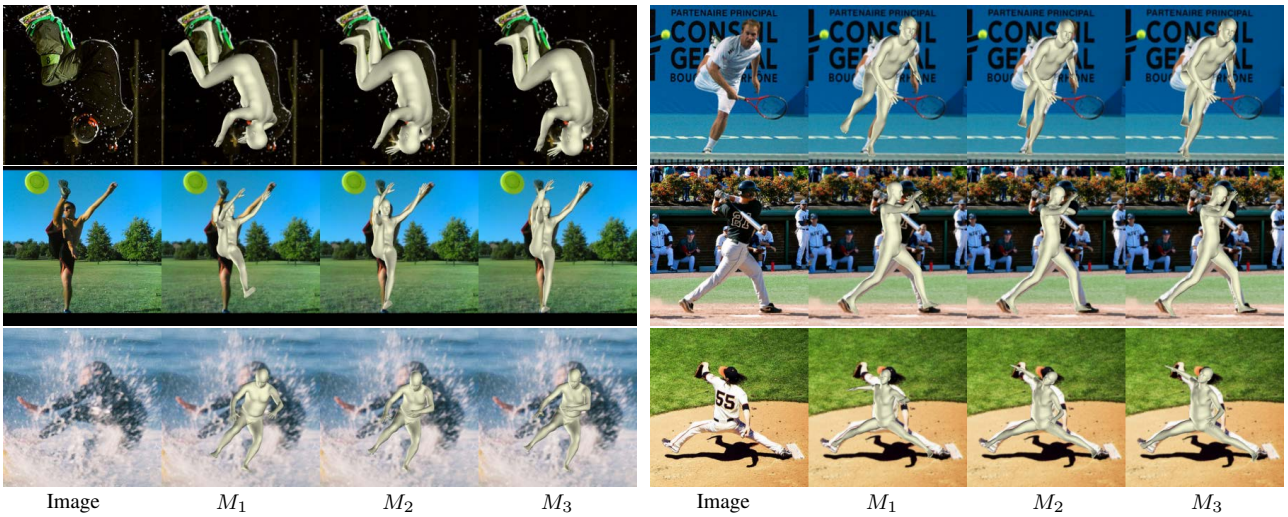

Figure S8: Erroneous reconstructions of our network. Though PyMAF can improve the alignment of some body parts, it remains challenging for PyMAF to correct those body parts with severe deviations, heavy occlusions, or ambiguous limb connections. Examples have the same layout with Figure S6.