


# PyMAF: 3D Human Pose and Shape Regression with Pyramidal Mesh Alignment Feedback Loop

Hongwen Zhang<sup>§†\*</sup>, Yating Tian<sup>†\*</sup>, Xinchi Zhou<sup>‡</sup>, Wanli Ouyang<sup>‡</sup>, Yebin Liu<sup>‡</sup>, Limin Wang<sup>†</sup> , Zhenan Sun<sup>§</sup> 

<sup>§</sup>CRIPAC, NLPR, Institute of Automation, Chinese Academy of Sciences, China

<sup>†</sup>State Key Laboratory for Novel Software Technology, Nanjing University, China

<sup>‡</sup>The University of Sydney, Australia <sup>‡</sup>Department of Automation, Tsinghua University, China

{hongwen.zhang@cripac, znsun@nlpr}.ia.ac.cn {yatingtian@smail., lmwang@}nju.edu.cn


{xinchi.zhou1, wanli.ouyang}@sydney.edu.au liuyebin@mail.tsinghua.edu.cn

## Abstract

Regression-based methods have recently shown promising results in reconstructing human meshes from monocular images. By directly mapping raw pixels to model parameters, these methods can produce parametric models in a feed-forward manner via neural networks. However, minor deviation in parameters may lead to noticeable misalignment between the estimated meshes and image evidences. To address this issue, we propose a Pyramidal Mesh Alignment Feedback (PyMAF) loop to leverage a feature pyramid and rectify the predicted parameters explicitly based on the mesh-image alignment status in our deep regressor. In PyMAF, given the currently predicted parameters, mesh-aligned evidences will be extracted from finer-resolution features accordingly and fed back for parameter rectification. To reduce noise and enhance the reliability of these evidences, an auxiliary pixel-wise supervision is imposed on the feature encoder, which provides mesh-image correspondence guidance for our network to preserve the most related information in spatial features. The efficacy of our approach is validated on several benchmarks, including Human3.6M, 3DPW, LSP, and COCO, where experimental results show that our approach consistently improves the mesh-image alignment of the reconstruction. The project page with code and video results can be found at <https://hongwenzhang.github.io/pymaf>.

## 1. Introduction

Aiming at the same goal of producing natural and well-aligned results, two different paradigms for human mesh recovery have been investigated in the research community. Optimization-based methods [5, 29, 65] explicitly fit

\*: Equal contribution. : Corresponding authors.

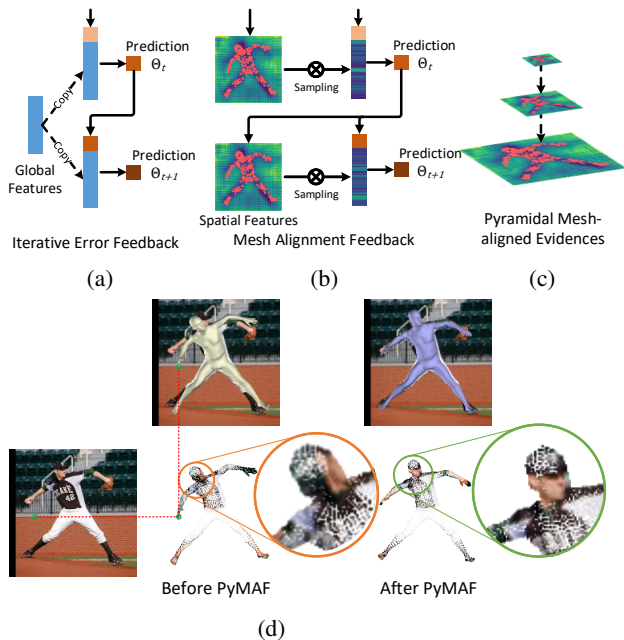


Figure 1: Illustration of our main idea. (a) The commonly-used iterative error feedback. (b) Our mesh alignment feedback. (c) Mesh-aligned evidences extracted from a feature pyramid. (d) Our approach PyMAF improves the mesh-image alignment of the estimated mesh.

the models to 2D evidences, which can typically produce results with accurate mesh-image alignments but tend to be slow and sensitive to the initialization. Alternatively, regression-based ones [22, 44, 27, 26] suggest to directly predict model parameters from images, which have shown very promising results, and yet still suffer from the coarse alignment between predicted meshes and image evidences.

For parametric models like SMPL [35], the joint poses are represented as the relative rotations with respect to their parent joints, which means that minor rotation errors accumulated along the kinematic chain can result in notice-

able drifts of joint positions. In order to generate well-aligned results, optimization-based methods [5, 29] design data terms in the objective function so that the alignment between the projection of meshes and 2D evidences can be explicitly optimized. Similar strategies are also exploited in regression-based methods [22, 44, 27, 26] to impose 2D supervisions upon the projection of estimated meshes in the training procedure. However, during testing, these deep regressors either are open-loop or simply include an Iterative Error Feedback (IEF) loop [22] in their architectures. As shown in Fig. 1(a), IEF reuses the same global features in its feedback loop, making its regressors hardly perceive the mesh-image misalignment in the inference phase.

As suggested in previous works [46, 39, 32, 52], neural networks tend to retain high-level information and discard detailed local features when reducing the spatial size of feature maps. To leverage spatial information in the regression networks, it is essential to extract pixel-wise contexts for fine-grained perception. Several attempts have been made to leverage pixel-wise representations such as part segmentation [41] or dense correspondences [61, 68] in their regression networks. Though these regressors take pixel-level evidences into consideration, it is still challenging for them to learn structural priors and get hold of spatial details at the same time based merely on high-resolution contexts.

Motivated by above observations, we design a Pyramidal Mesh Alignment Feedback (PyMAF) loop in our regression network to exploit multi-scale contexts for better mesh-image alignment of the reconstruction. The central idea of our approach is to correct parametric deviations explicitly and progressively based on the alignment status. In PyMAF, mesh-aligned evidences will be extracted from spatial features according to the 2D projection of the estimated mesh and then fed back to regressors for parameter updates. As illustrated in Fig. 1, the mesh alignment feedback loop can take advantage of more informative features for parameter correction in comparison with the commonly used iterative error feedback loop [22, 7]. To leverage multi-scale contexts, mesh-aligned evidences are extracted from a feature pyramid so that coarse-aligned meshes can be corrected with large step sizes based on the lower-resolution features. Moreover, to enhance the reliability of the spatial cues, an auxiliary task is imposed on the highest-resolution feature to infer pixel-wise dense correspondences, which provides guidance for the image encoder to preserve the mesh-image alignment information. The contributions of this work can be summarized as follows:

- A mesh alignment feedback loop is introduced for regression-based human mesh recovery, where mesh-aligned evidences are exploited to correct parametric errors explicitly so that the estimated meshes can be better aligned with input images.
- A feature pyramid is further incorporated with the mesh

alignment feedback loop so that the regression network can leverage multi-scale alignment contexts.

- An auxiliary pixel-wise supervision is imposed on the image encoder so that its spatial features can be more informative and the mesh-aligned evidences can be more relevant and reliable.

## 2. Related Work

### 2.1. Human Pose and Shape Recovery

**Optimization-based Approaches.** Pioneering work in this field mainly focus on the optimization process of fitting parametric models (*e.g.*, SCAPE [3] and SMPL [35]) to 2D observations such as keypoints and silhouettes [50, 13, 5]. In their objective functions, prior terms are designed to penalize the unnatural shape and pose, while data terms measure the fitting errors between the re-projection of meshes and 2D evidences. Based on this paradigm, different updates have been investigated to incorporate information such as 2D/3D body joints [5, 70], silhouettes [29, 16], part segmentation [65] in the fitting procedure. Despite the well-aligned results obtained by these optimization-based methods, their fitting process tends to be slow and sensitive to initialization. Recently, Song *et al.* [51] exploit the learned gradient descent in the fitting process. Though this solution leverages rich 2D pose datasets and alleviates many issues in traditional optimization-based methods, it still relies on the accuracy of 2D poses and breaks the end-to-end learning. Alternatively, our solution supports the end-to-end learning but can also leverage rich 2D datasets thanks to the progress (*e.g.*, SPIN [26] and EFT [21]) in the generation of more precise pseudo 3D ground-truth for 2D datasets.

**Regression-based Approaches.** Alternatively, taking advantage of the powerful nonlinear mapping capability of neural networks, recent regression-based approaches [22, 44, 41, 26, 9, 18, 8] have made significant advances in predicting human models directly from monocular images. These deep regressors take 2D evidences as input and learn model priors implicitly in a data-driven manner under different types of supervision signals [55, 22, 42, 45, 11, 63, 28] during the learning procedure. To mitigate the learning difficulty of the regressor, different network architectures have also been designed to leverage proxy representations such as silhouette [44, 56], 2D/3D joints [55, 44, 38], segmentation [41, 47] and dense correspondences [61, 68]. Such strategies can benefit from synthetic data [61, 49] and the progress in the estimation of proxy representations [6, 1, 52, 59]. Despite the effectiveness of these modules, the quality of proxy representations becomes the bottleneck for the reconstruction task, which may also block the end-to-end learning of the deep regressor. Moreover, though supervision signals are imposed on the projection of the estimated models to penalize the fitting misalignment

during the training of deep regressors, their architectures can hardly perceive the misalignment during the inference phase. In comparison, the proposed PyMAF is close-loop for both training and inference, which enables a feedback loop in our deep regressor to leverage spatial evidences for better mesh-image alignment of the estimated human models. Our work focuses on the design of regressor architectures and can also provides a better regressor for those approaches using post-processing [14] or the work on pseudo ground-truth generation [26, 21].

Directly regressing model parameters from images is very challenging even for neural networks. Existing methods have also offered non-parametric solutions to reconstruct human body models in non-parametric representations. Among them, volumetric representation [56, 72], implicit function [48, 71], mesh vertices [27, 31], and position maps [62, 69, 66] have been adopted as regression targets. Using non-parametric representations as the regression targets is more readily to leverage high-resolution features but needs further processing to retrieve parametric models from the outputs. Besides, solely using high-resolution features makes the algorithms more sensitive to occlusions without additional structure priors. In our solution, the deep regressor makes use of spatial features at multiple scales for high-level and fine-grained perception and produces parametric models directly with no further processing required.

## 2.2. Iterative Fitting in Regression Tasks

Strategies of incorporating fitting processes along with the regression have also been investigated in the literature. For human model reconstruction, Kolotouros *et al.* [26] combine an iterative fitting procedure to the training procedure in order to generate preciser ground truth for better supervision. To refine the estimated meshes during both training and inference phases, several attempts have been made to deform human meshes so that they can be aligned with the intermediate estimations such as depth maps [74], part segmentation [64], and dense correspondences [14]. These approaches adopt intermediate estimations as the fitting goals and hence rely on their quality. In contrast, our approach uses the currently estimated meshes to extract deep features for refinement, which not only behaves symmetrically for both training and inference but also enables fully end-to-end learning of the deep regressor.

To put our approach in a broader view, there have been remarkable efforts made to involve iterative fitting strategies in other computer vision tasks, including facial landmark localization [60, 54], human/hand pose estimation [40, 7], *etc.* For generic objects, Pixel2Mesh [58] progressively deforms an initial ellipsoid by leveraging perceptual features. Following the spirit of these works, we exploit new strategies to extract fine-grained evidences and contribute novel solutions in the context of human mesh recovery.

## 3. Methodology

In this section, we will present the technical details of our approach. As illustrated in Fig. 2, our network produces a feature pyramid for mesh recovery in a coarse-to-fine fashion. Coarse-aligned predictions will be improved by utilizing the mesh-aligned evidences extracted from spatial feature maps. Moreover, an auxiliary prediction task is imposed on the image encoder, so that those spatial cues can be more reliable and relevant.

### 3.1. Feature Pyramid for Human Mesh Regression

The goal of our image encoder is to generate a pyramid of spatial features from coarse to fine granularities, which provide descriptions of the posed person in the image at different scale levels. The feature pyramid will be used in subsequent predictions of the SMPL model with the pose, shape, and camera parameters  $\Theta = \{\theta, \beta, \pi\}$ .

Formally, the encoder takes an image  $I$  as input and outputs a set of spatial features  $\{\phi_s^t \in \mathbb{R}^{C_s \times H_s^t \times W_s^t}\}_{t=0}^{T-1}$  at the end, where  $H_s^t$  and  $W_s^t$  are monotonically increasing. At level  $t$ , based on the feature map  $\phi_s^t$ , a set of sampling points  $X_t$  will be used to extract point-wise features. Specifically, for each 2D point  $x$  in  $X_t$ , point-wise features  $\phi_s^t(x) \in \mathbb{R}^{C_s \times 1}$  will be extracted from  $\phi_s^t$  accordingly using the bilinear sampling. These point-wise features will go through a MLP (multi-layer perceptron) for dimension reduction and be further concatenated together as a feature vector  $\phi_p^t$ , *i.e.*,

$$\phi_p^t = \mathcal{F}(\phi_s^t, X_t) = \oplus (\{f(\phi_s^t(x)), \text{for } x \text{ in } X_t\}), \quad (1)$$

where  $\mathcal{F}(\cdot)$  denotes the feature sampling and processing operations,  $\oplus$  denotes the concatenation, and  $f(\cdot)$  is the MLP. After that, a parameter regressor  $\mathcal{R}_t$  takes features  $\phi_p^t$  and the current estimation of parameters  $\Theta_t$  as inputs and outputs the parameter residual. Parameters are then updated as  $\Theta_{t+1}$  by adding the residual to  $\Theta_t$ . For the level  $t = 0$ ,  $\Theta_0$  adopts the mean parameters calculated from training data.

Given the parameter predictions  $\Theta$  (the subscript  $t$  is omitted for simplicity) at each level, a mesh with vertices of  $M = \mathcal{M}(\theta, \beta) \in \mathbb{R}^{N \times 3}$  can be generated accordingly, where  $N = 6890$  denotes the number of vertices in the SMPL model. These mesh vertices can be mapped to sparse 3D joints  $J \in \mathbb{R}^{N_j \times 3}$  by a pretrained linear regressor, and further projected on the image coordinate system as 2D key-points  $K = \Pi(J) \in \mathbb{R}^{N_j \times 2}$ , where  $\Pi(\cdot)$  denotes the projection function based on the camera parameters  $\pi$ . Note that the pose parameters in  $\Theta$  are represented as relative rotations along kinematic chains and minor parameter errors can lead to large misalignment between the 2D projection and image evidences. To penalize such misalignment during the training of the regression network, we follow common practices [22, 26] to add 2D supervisions on the

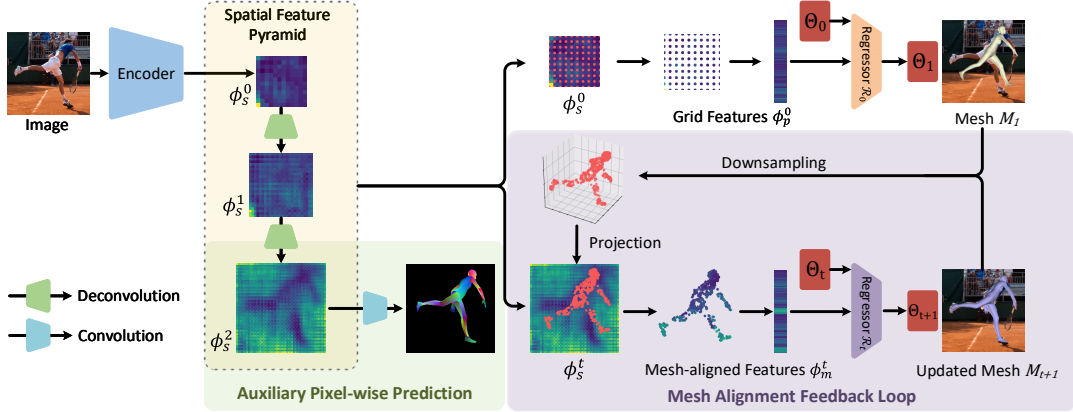


Figure 2: Overview of the proposed Pyramidal Mesh Alignment Feedback (PyMAF). PyMAF leverages a feature pyramid and enables an alignment feedback loop in our network. Given a coarse-aligned model prediction, mesh-aligned evidences are extracted from finer-resolution features accordingly and fed back a regressor for parameter rectification. To enhance the reliability of spatial evidences, an auxiliary pixel-wise prediction task is imposed on the final output of the image encoder.

2D keypoints projected from the estimated mesh. Meanwhile, additional 3D supervisions on 3D joints and model parameters are added when ground truth 3D labels are available. Overall, the loss function for the parameter regressor is written as

$$\mathcal{L}_{reg} = \lambda_{2d} \|K - \hat{K}\| + \lambda_{3d} \|J - \hat{J}\| + \lambda_{para} \|\Theta - \hat{\Theta}\|, \quad (2)$$

where  $\|\cdot\|$  is the squared L2 norm,  $\hat{K}$ ,  $\hat{J}$ , and  $\hat{\Theta}$  denote the ground truth 2D keypoints, 3D joints, and model parameters, respectively.

One of the improvements over the commonly used parameter regressors is that our regressors can better leverage spatial information. Unlike the commonly used regressors taking the global features  $\phi_g \in \mathbb{R}^{C_g \times 1}$  as input, our regressor uses the point-wise information obtained from spatial features  $\phi_s^t$ . A straight-forward strategy to extract point-wise features would be using the points  $X_t$  with a grid pattern and uniformly sampling features from  $\phi_s^t$ . In the proposed approach, the sampling points  $X_t$  adopt the grid pattern at the level  $t = 0$  and will be updated according to the currently estimated mesh when  $t > 0$ . We will show that, such a mesh conditioned sampling strategy helps the regressor to produce more plausible results.

### 3.2. Mesh Alignment Feedback Loop

As mentioned in HMR [22], directly regressing mesh parameters in one go is challenging. To tackle this issue, HMR uses an Iterative Error Feedback (IEF) loop to iteratively update  $\Theta$  by taking the global features  $\phi_g$  and the current estimation of  $\Theta$  as input. Though the IEF strategy reduces parameter errors progressively, it uses the same global features each time for parameter update, which lacks fine-grained information and is not adaptive to new predictions. By contrast, we propose a Mesh Alignment Feedback (MAF) loop

so that mesh-aligned evidences can be leveraged in our regressor to rectify current parameters and improve the mesh-image alignment of the currently estimated model.

To this end, when  $t > 0$ , we extract mesh-aligned features from  $\phi_s^t$  based on the currently estimated mesh  $M_t$  to obtain more fine-grained and position-sensitive evidences. Compared with the global features or the uniformly sampled grid features, mesh-aligned features can reflect the mesh-image alignment status of the current estimation, which is more informative for parameter rectification. Specifically, the sampling points  $X_t$  are obtained by first down-sampling the mesh  $M_t$  to  $\tilde{M}_t$  and then projecting it on the 2D image plane, *i.e.*,  $X_t = \Pi(\tilde{M}_t)$ . Based on  $X_t$ , the mesh-aligned features  $\phi_m^t$  will be extracted from  $\phi_s^t$  using Eq. 1, *i.e.*,  $\phi_m^t = \phi_p^t = \mathcal{F}(\phi_s^t, \Pi(\tilde{M}_t))$ . These mesh-aligned features will be fed into the regressor  $\mathcal{R}_t$  for parameter update. Overall, the proposed mesh alignment feedback loop can be formulated as

$$\Theta_{t+1} = \Theta_t + \mathcal{R}_t \left( \Theta_t, \mathcal{F}(\phi_s^t, \Pi(\tilde{M}_t)) \right), \text{ for } t > 0. \quad (3)$$

### 3.3. Auxiliary Pixel-wise Supervision

As depicted in the second row of Fig. 3, spatial features tend to be affected by the noisy inputs, since input images may contain a large amount of unrelated information such as occlusions, appearance and illumination variations. To improve the reliability of the mesh-aligned cues extracted from spatial features, we impose an auxiliary pixel-wise prediction task on the spatial features at the last level. Specifically, during training, the spatial feature maps  $\phi_s^{T-1}$  will go through a convolutional layer for the prediction of dense correspondence maps with pixel-wise supervision applied. Dense correspondences encode the mapping relationship between foreground pixels on 2D image plane and

mesh vertices in 3D space. In this way, the auxiliary supervision provides mesh-image correspondence guidance for the image encoder to preserve the most related information in the spatial feature maps.

In our implementation, we adopt the IUUV maps defined in DensePose [1] as the dense correspondence representation, which consists of part index and UV values of the mesh vertices. Note that we do not use DensePose annotations in the dataset but render IUUV maps based on the ground-truth SMPL models [68]. During training, classification and regression losses are applied on the part index  $P$  and  $UV$  channels of dense correspondence maps, respectively. Specifically, for the part index  $P$  channels, a cross-entropy loss is applied to classify a pixel belonging to either background or one among body parts. For the  $UV$  channels, a smooth L1 loss is applied to regress the corresponding  $UV$  values of the foreground pixels. Only the foreground regions are taken into account in the  $UV$  regression loss, *i.e.*, the estimated  $UV$  channels are firstly masked by the ground-truth part index channels before applying the regression loss. Overall, the loss function for the auxiliary pixel-wise supervision is written as

$$\begin{aligned} \mathcal{L}_{aux} = & \lambda_{pi} CrossEntropy(P, \hat{P}) \\ & + \lambda_{uv} SmoothL1(\hat{P} \odot U, \hat{P} \odot \hat{U}) \\ & + \lambda_{uv} SmoothL1(\hat{P} \odot V, \hat{P} \odot \hat{V}), \end{aligned} \quad (4)$$

where  $\odot$  denotes the mask operation. Note that the auxiliary prediction is required in the training phase only.

Fig. 3 visualizes the spatial features of the encoder trained with and without auxiliary supervision, where the feature maps are simply added along the channel dimension as grayscale images and visualized with colormap. We can see that the spatial features are more neat and robust to the input variations when the auxiliary supervision is applied.

## 4. Experiments

### 4.1. Implementation Details

The proposed PyMAF is validated on the ResNet-50 [15] backbone pre-trained on ImageNet [10]. The ResNet-50 backbone takes a  $224 \times 224$  image as input and produces image features with the size of  $2048 \times 7 \times 7$ . For the classic regression network HMR [22], a  $2048 \times 1$  global feature vector will be obtained after average pooling. In our approach, the image features will go through deconvolution layers, resulting spatial feature maps with resolutions of  $\{14 \times 14, 28 \times 28, 56 \times 56\}$ , where  $C_s = 256$  for all resolutions. Here, the maximum number  $T$  is set to 3, which is equal to the iteration number used in HMR. When generating mesh-aligned features, the SMPL mesh is down-sampled using a pre-computed downsampling matrix provided in [27], after which the vertex number drops from

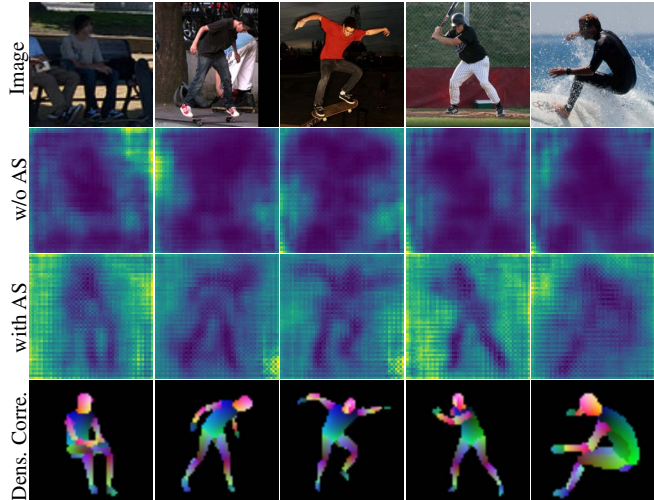


Figure 3: Visualization of the spatial feature maps and predicted dense correspondences. Top: Input images. Second / Third Row: Spatial feature maps learned without / with the Auxiliary Supervision (AS). Bottom: Predicted dense correspondence maps under the auxiliary supervision.

6890 to 431. The mesh-aligned features of each point will be processed by a three-layer MLP so that their dimension will be reduced from  $C_s$  to 5. Hence, the mesh-aligned feature vector has a length of  $2155 = 431 \times 5$ , which is similar to the length of global features. For the grid features used at  $t = 0$ , they are uniformly sampled from  $\phi_s^0$  with a  $21 \times 21$  grid pattern, *i.e.*, the point number is  $441 = 21 \times 21$  which is approximate to the vertices number 431 after mesh down-sampling. The regressors  $\mathcal{R}_t$  have the same architecture with the regressor in HMR except that they have slightly different input dimensions. Following the setting of SPIN [26], we train our network using the Adam [24] optimizer with the learning rate set to  $5e-5$  and the batch size of 64 on a single 2080 Ti GPU. No learning rate decay is applied during training. More details of the implementation can be found in our code and the Supplementary Material.

### 4.2. Datasets

Following the settings of previous work [22, 26], our approach is trained on a mixture of data from several datasets with 3D and 2D annotations, including Human3.6M [17], MPI-INF-3DHP [37], LSP [19], LSP-Extended [20], MPII [2], and COCO [33]. For the last five datasets, we also utilize their pseudo ground-truth SMPL parameters [5, 26] for training. We do not use training data from 3DPW [57] but only perform evaluations on its test set. Moreover, we do not use the DensePose annotations in COCO for auxiliary supervision, but render IUUV maps based on the ground-truth SMPL meshes using the method described in [68]. We evaluate our approach using a variety of metrics for quantitative comparisons with previous meth-

Method	3DPW			Human3.6M		
	PVE	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE	
Temporal	Kanazawa <i>et al.</i> [23]	139.3	116.5	72.6	-	56.9
	Doersch <i>et al.</i> [11]	-	-	74.7	-	-
	Arnab <i>et al.</i> [4]	-	-	72.2	77.8	54.3
	DSD [53]	-	-	69.5	59.1	42.4
	VIBE [25]	113.4	93.5	56.5	65.9	41.5
	Frame-based	Pavlakos <i>et al.</i> [44]	-	-	-	-
HMR [22]		-	130.0	76.7	88.0	56.8
NBF [41]		-	-	-	-	59.9
GraphCMR [27]		-	-	70.2	-	50.1
HoloPose [14]		-	-	-	60.3	46.5
DenseRaC [61]		-	-	-	76.8	48.0
SPIN [26]		116.4	96.9	59.2	62.5	41.1
DecoMR [66]		-	-	61.7 <sup>†</sup>	-	<b>39.3<sup>†</sup></b>
DaNet [67]		-	-	56.9	61.5	48.6
Song <i>et al.</i> [51]		-	-	<b>55.9</b>	-	56.4
I2L-MeshNet [38]		-	100.0	60.0	<b>55.7<sup>†</sup></b>	41.1 <sup>†</sup>
HKMR [12]		-	-	-	59.6	43.2
Baseline		117.9	98.5	60.9	64.8	43.7
PyMAF w/o AS		113.6	95.6	58.8	60.3	42.3
PyMAF	<b>110.1</b>	<b>92.8</b>	58.9	57.7	40.5	

Table 1: Reconstruction errors on 3DPW and Human3.6M. <sup>†</sup> denotes the numbers evaluated on non-parametric results.

ods, *i.e.*, PVE, MPJPE, and PA-MPJPE for the evaluation of 3D pose estimation, the segmentation accuracy, f1 scores, and AP for the measure of mesh-image alignment. Detailed descriptions about the datasets and evaluation metrics can be found in the Supplementary Material.

### 4.3. Comparison with the State of the Art

**3D Human Pose and Shape Estimation.** We first evaluate our approach on the 3D human pose and shape estimation task, and make comparisons with previous state-of-the-art regression-based methods. We present evaluation results for quantitative comparison on 3DPW and Human3.6M datasets in Table 1. Our PyMAF achieves competitive or superior results among previous approaches, including both frame-based and temporal approaches. Note that the approaches reported in Table 1 are not strictly comparable since they may use different training data, learning rate schedules, or training epochs, *etc.* For a fair comparison, we report results of our baseline in Table 1, which is trained under the same setting with PyMAF. The baseline approach has the same network architecture with HMR [22] and also adopts the 6D rotation representation [73] for pose parameters. Compared with the baseline, PyMAF reduces the MPJPE by 5.7 mm and 7.1 mm on 3DPW and Human3.6M datasets, respectively. The auxiliary supervision (AS) also helps PyMAF to have better reconstruction results as shown in the last two rows of Table 1.

From Table 1, we can see that PyMAF has more notable improvements on the metrics MPJPE and PVE. We would argue that the metric PA-MPJPE can not fully reveal the

Method	FB Seg.		Part Seg.	
	acc.	f1	acc.	f1
SMPLify oracle [5]	92.17	0.88	88.82	0.67
SMPLify [5]	91.89	0.88	87.71	0.64
SMPLify on [44]	92.17	0.88	88.24	0.64
HMR [22]	91.67	0.87	87.12	0.60
BodyNet [56]	92.75	0.84	-	-
CMR [27]	91.46	0.87	88.69	0.66
TexturePose [42]	91.82	0.87	89.00	0.67
SPIN [26]	91.83	0.87	89.41	0.68
DecoMR [66]	92.10	0.88	89.45	0.69
HKMR [12]	92.23	0.88	89.59	0.69
Baseline	91.67	0.87	89.23	0.68
PyMAF w/o AS	92.43	0.88	89.98	0.70
PyMAF	<b>92.79</b>	<b>0.89</b>	<b>90.47</b>	<b>0.72</b>

Table 2: Foreground-background and six-part segmentation accuracy and f1 scores on the LSP test set. SMPLify Oracle denotes the SMPLify using ground-truth keypoints as inputs.

mesh-image alignment performance since it is calculated as the MPJPE after rigid alignment. As depicted in the Supplementary Material, a reconstruction result with smaller PA-MPJPE value can have larger MPJPE value and worse alignment between the reprojected mesh and image.

**2D Segmentation and Pose Estimation.** To quantitatively measure the mesh-image alignment of the predictions, we also conduct evaluation on the 2D segmentation and pose estimation task, where the predicted meshes are projected on the image plane to obtain 2D part segmentation and keypoints. Table 2 reports the assessment of foreground-background and six-part segmentation performance on the LSP test set. As shown in Table 2, optimization-based approaches remain very competitive in terms of 2D alignment metrics and tend to outperform most of the regression-based ones. The reason behind it is that optimization-based approaches is optimized for the mesh-image alignment explicitly. Though PyMAF is regression-based, it surpasses all other methods including the optimization-based ones.

Finally, we evaluate 2D human pose estimation performance on the COCO validation set to verify the effectiveness of our approach in real-world scenarios. During the evaluation, we project keypoints from the estimated mesh on the image plane, and compute the Average Precision (AP) based on the keypoint similarity with the ground truth 2D keypoints. The results of keypoint localization APs are reported in Table 3. OpenPose [6], a widely-used 2D human pose estimation algorithm, are also included for reference. We can see that, the COCO dataset is very challenging for approaches to human mesh reconstruction as they typically have much worse performances in terms of 2D keypoint localization accuracy. In Table 3, we also include the results of the optimization-based SMPLify [5] by fitting the SMPL



Figure 4: Qualitative comparison of reconstruction results on the COCO validation set.

Method	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>M</sub>	AP <sub>L</sub>
OpenPose [6]	<b>65.3</b>	<b>85.2</b>	<b>71.3</b>	<b>62.2</b>	<b>70.7</b>
SMPLify [5]	22.0	37.7	<b>23.1</b>	<b>27.7</b>	17.6
HMR [22]	18.9	47.5	11.7	21.5	17.0
GraphCMR [27]	9.3	26.9	4.2	11.3	8.1
SPIN [26]	17.3	39.1	13.5	19.0	16.6
Baseline	16.8	38.2	12.8	18.5	16.0
PyMAF w/o AS	20.7	43.9	17.4	22.3	19.9
PyMAF	<b>24.6</b>	<b>48.9</b>	22.7	26.0	<b>24.2</b>

Table 3: Keypoint localization APs on the COCO validation set. There is a total of 50,197 samples used for evaluation. Results of SMPLify [5] are evaluated based on the implementation in SPIN [26] with 300 optimization iterations. Results of HMR [22], GraphCMR [27], and SPIN [26] are evaluated based on their publicly released code and models.

model to the ground-truth 2D keypoints. As pointed out in previous work [26], SMPLify may produce well-aligned but unnatural results. Moreover, SMPLify is much more time-consuming than regression-based solutions. Among approaches recovering 3D human mesh, PyMAF outperforms previous regression-based methods by remarkable margins. Compared with our baseline, PyMAF significantly improves the AP and AP<sub>50</sub> by 7.8% and 10.7%, respectively. The auxiliary supervision (AS) also considerably contributes to more robust reconstruction in this challenging dataset and brings 3.9% performance gain on AP. Reconstruction results on COCO are depicted in Fig. 4 for qualitative comparisons, where PyMAF convincingly performs better than SPIN [26] and our baseline by producing better-aligned and natural results.

#### 4.4. Ablation Study

In this part, we will perform ablation studies under various settings on Human3.6M to validate the efficacy of the

Feedback Feat.	Pyramid?	# Regressor	MPJPE	PA-MPJPE
Global (Baseline)	No	1	84.1	55.6
Global	No	1	84.3	55.3
Grid			80.5	54.7
Mesh-aligned	Yes	3	<b>79.6</b>	<b>53.4</b>
Grid			79.7	54.3
Mesh-aligned			<b>76.8</b>	<b>50.9</b>

Table 4: Ablation study on using different types of feedback features for refinement. No auxiliary supervision is applied.

key components proposed in our approach. All ablation approaches are trained and tested on Human3.6M, as it includes ground-truth 3D labels and is the most widely used benchmark for 3D human pose and shape estimation.

**Efficacy of Mesh-aligned Features.** In PyMAF, mesh-aligned features provide the current mesh-image alignment information in the feedback loop, which is essential for better mesh recovery. To verify this, we alternatively replace mesh-aligned features with the global features or the grid features uniformly sampled from spatial features as the input for parameter regressors. Table 4 reports the performances of the approaches equipped with different types of features in the feedback loop. The results under the non-pyramidal setting are also included in Table 4, where the grid and mesh-aligned features are extracted from the feature maps with the highest resolution (*i.e.*,  $56 \times 56$ ) and the mesh-aligned features are extracted on the reprojected points of the mesh under the mean pose at  $t = 0$ . For fair comparisons with Baseline, the approaches with global, grid, and mesh-aligned feedback features under the non-pyramidal setting also use a single regressor but have individual supervision on the prediction of each iteration. Besides, all approaches in Table 4 do not use the auxiliary supervision.

Unsurprisingly, using mesh-aligned features yields the best performance under both non-pyramidal and pyramidal designs. The approach using the grid features sampled from spatial feature maps has better results than that using the global features but is worse than the mesh-aligned counterpart. When using pyramidal feature maps, the mesh-aligned solution achieves even more performance gain since multi-scale mesh-alignment evidences can be leveraged in the feedback loop. Though the grid features largely contain spatial cues on uniformly distributed pixel positions, they can not reflect the alignment status of the current estimation. This implies that mesh-aligned features are the most informative one for the regressor to rectify the current mesh parameters.

**Benefit from Auxiliary Supervision.** The auxiliary pixel-wise supervision helps to enhance the reliability of the mesh-aligned evidences extracted from spatial features. Using alternative pixel-wise supervision such as part segmen-

Aux. Supv.	Input Type	MPJPE	PA-MPJPE
None	Feature	76.8	50.9
Part. Seg.	Part. Seg.	108.1	75.9
	Feature	75.5	49.2
	Feature*Part. Seg.	77.6	51.1
Dense Corr.	Dense Corr.	77.8	54.7
	Feature	<b>75.1</b>	<b>48.9</b>

Table 5: Ablation study on using different auxiliary supervision settings and input types for regressors.

tation rather than dense correspondences is also possible in our framework. In our approach, these auxiliary predictions are solely needed for supervisions during training since the point-wise features are extracted from feature maps. For more in-depth analyses, we have also tried extracting point-wise features from the auxiliary predictions, *i.e.*, the input type of regressors are intermediate representations such as part segmentation or dense correspondences. Table 5 shows the comparison of experiments with different auxiliary supervision settings and input types for regressors during training. Using part segmentation is slightly worse than our dense correspondence solution. Compared with the part segmentation, the dense correspondences preserve not only clean but also rich information in foreground regions. Moreover, using feature maps for point-wise feature extraction consistently performs better than using auxiliary predictions. This can be explained by the fact that using intermediate representations as input for regressors hampers the end-to-end learning of the whole network. Under the auxiliary supervision strategy, the spatial feature maps are learned with the signal backpropagated from both auxiliary prediction and parameter correction tasks. In this way, the background features can also contain information for mesh parameter correction since the deep features have larger receptive fields and are trained in an end-to-end manner. As shown in Table 5, when the mesh-aligned features are masked with the foreground region of part segmentation predictions, the performance degrades from 75.5 mm to 77.6 mm on MPJPE.

**Effect of the Initialization in the Feedback Loop.** In our approach, the point-wise features are initially extracted on grid points for coarse mesh predictions before the extraction of mesh-aligned features. It is also feasible to extract mesh-aligned features based on a mean-pose mesh at  $t = 0$  as the initial features. The performances of the approaches using different initial features are reported in Table 6, where PyMAF can improve reconstruction results under both initialization settings. For the approach using the initial features extracted on the projection of the mean-pose mesh, we visualize its estimated meshes after each iteration in Fig 5. Though the mean-pose mesh is far away from the ground-truth, PyMAF can correct the drift of body parts progressively and result in better-aligned human models.

Initial Feat.	Metric	$M_0$	$M_1$	$M_2$	$M_3$
Mean-pose Mesh	MPJPE	274.0	81.4	78.0	77.3
	PA-MPJPE	131.7	54.1	51.1	50.3
Grid Points	MPJPE	274.0	80.3	76.6	75.1
	PA-MPJPE	131.7	52.1	49.9	48.9

Table 6: Ablation study on using different initial features across iterations in the feedback loop.

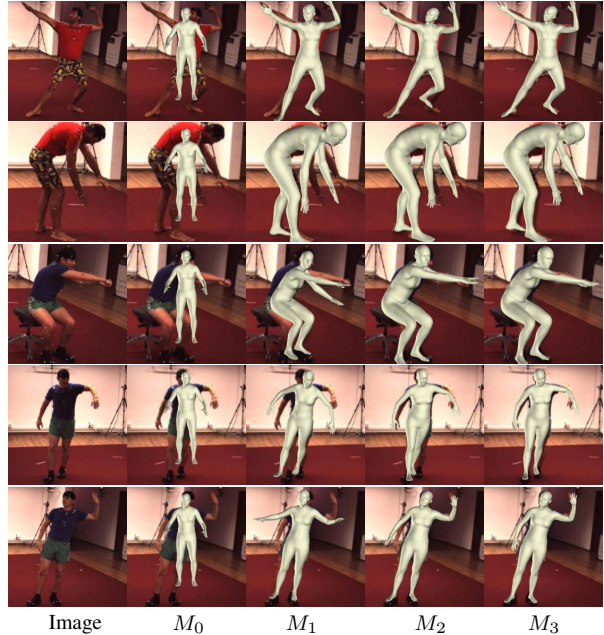


Figure 5: Visualization of reconstruction results across different iterations in the feedback loop.

## 5. Limitations and Future Work

In this paper, we present Pyramidal Mesh Alignment Feedback (PyMAF) for regression-based human mesh recovery. PyMAF is primarily motivated by the observation of the reprojection misalignment of parametric mesh results. Though PyMAF improves the mesh-image alignment on 2D image planes, it can still hardly address the depth ambiguity problem in 3D space. Moreover, PyMAF fails to handle extreme shapes due to the lack of training data, as shown in the Supplementary Material.

In the future, PyMAF can be extended and incorporated with the recent progress to improve the alignment in 3D space [36, 30]. Moreover, combining PyMAF with HoloPose [14], SPIN [26], or EFT [21] for the generation of more precise pseudo 3D ground-truth labels would also be interesting future work.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (Grant U1836217, 62125107, 62076119, 61921006), Australian Research Council (DP200103223), and Australian Medical Research Future Fund (MRFAI000085). Wanli Ouyang is also supported by SenseTime.



## References

- [1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 2, 5
- [2] 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. 5, 12
- [3] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics*, volume 24, pages 408–416. ACM, 2005. 2
- [4] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2019. 6
- [5] 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, 2, 5, 6, 7, 12
- [6] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En Wei, and Yaser A Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 6, 7
- [7] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4733–4742, 2016. 2, 3
- [8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *Proceedings of the European Conference on Computer Vision*, 2020. 2
- [9] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *Proceedings of the European Conference on Computer Vision*. Springer, 2020. 2
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 5
- [11] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. In *Advances in Neural Information Processing Systems*, pages 12949–12961, 2019. 2, 6
- [12] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Kosecka, and Ziyang Wu. Hierarchical kinematic human mesh recovery. In *Proceedings of the European Conference on Computer Vision*. Springer, 2020. 6
- [13] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1381–1388. IEEE, 2009. 2
- [14] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10884–10894, 2019. 3, 6, 8
- [15] 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. 5
- [16] 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 *International Conference on 3D Vision*, pages 421–430. IEEE, 2017. 2
- [17] 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. 5, 12
- [18] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2020. 2
- [19] 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. 5, 12
- [20] 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. 5, 12
- [21] 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, 3, 8
- [22] 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, 3, 4, 5, 6, 7, 12, 13
- [23] 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. 6, 12
- [24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2014. 5, 12
- [25] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. 6

- [26] 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](#), [5](#), [6](#), [7](#), [8](#), [12](#), [13](#), [14](#), [15](#)
- [27] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#)
- [28] Jogendra Nath Kundu, Mugalodi Rakesh, Varun Jampani, Rahul Mysore Venkatesh, and R Venkatesh Babu. Appearance consensus driven self-supervised human mesh recovery. In *Proceedings of the European Conference on Computer Vision*, pages 794–812. Springer, 2020. [2](#)
- [29] 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. [1](#), [2](#), [12](#), [13](#)
- [30] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3383–3393, 2021. [8](#)
- [31] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. [3](#)
- [32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. [2](#)
- [33] 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. [5](#), [12](#)
- [34] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics*, 33(6):220, 2014. [12](#)
- [35] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248, 2015. [1](#), [2](#)
- [36] Tianyu Luan, Yali Wang, Junhao Zhang, Zhe Wang, Zhipeng Zhou, and Yu Qiao. Pc-hmr: Pose calibration for 3d human mesh recovery from 2d images/videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. [8](#)
- [37] 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. [5](#), [12](#)
- [38] 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 *Proceedings of the European Conference on Computer Vision*, 2020. [2](#), [6](#)
- [39] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 483–499. Springer, 2016. [2](#)
- [40] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Generalized feedback loop for joint hand-object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):1898–1912, 2019. [3](#)
- [41] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *International Conference on 3D Vision*, pages 484–494. IEEE, 2018. [2](#), [6](#)
- [42] Georgios Pavlakos, Nikos Kolotouros, and Kostas Daniilidis. Texturepose: Supervising human mesh estimation with texture consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 803–812, 2019. [2](#), [6](#)
- [43] 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. [12](#)
- [44] 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. [1](#), [2](#), [6](#), [12](#), [13](#)
- [45] Yu Rong, Ziwei Liu, Cheng Li, Kaidi Cao, and Chen Change Loy. Delving deep into hybrid annotations for 3d human recovery in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5340–5348, 2019. [2](#)
- [46] 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](#)
- [47] Nadine Rueegg, Christoph Lassner, Michael J Black, and Konrad Schindler. Chained representation cycling: Learning to estimate 3d human pose and shape by cycling between representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. [2](#)
- [48] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019. [3](#)
- [49] 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](#), [13](#), [14](#)
- [50] Leonid Sigal, Alexandru Balan, and Michael J Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Advances in Neural Information Processing Systems*, pages 1337–1344, 2008. [2](#)

- [51] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In *Proceedings of the European Conference on Computer Vision*, 2020. 2, 6
- [52] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2
- [53] Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, YiLi Fu, and Tao Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5349–5358, 2019. 6
- [54] George Trigeorgis, Patrick Snape, Mihalis A Nicolaou, Epameinondas Antonakos, and Stefanos Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4187, 2016. 3
- [55] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Advances in Neural Information Processing Systems*, pages 5236–5246, 2017. 2
- [56] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European Conference on Computer Vision*, pages 20–36, 2018. 2, 3, 6
- [57] 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. 5, 12
- [58] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision*, pages 52–67, 2018. 3
- [59] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3d multi-person pose estimation using multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2
- [60] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 532–539, 2013. 3
- [61] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7760–7770, 2019. 2, 6
- [62] Pengfei Yao, Zheng Fang, Fan Wu, Yao Feng, and Jiwei Li. Densebody: Directly regressing dense 3d human pose and shape from a single color image. *arXiv preprint arXiv:1903.10153*, 2019. 3
- [63] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, Bill Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3d human pose and shape reconstruction with normalizing flows. In *Proceedings of the European Conference on Computer Vision*. Springer, 2020. 2
- [64] 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 Conference on Computer Vision and Pattern Recognition*, pages 14484–14493, 2021. 3
- [65] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2148–2157, 2018. 1, 2
- [66] Wang Zeng, Wanli Ouyang, Ping Luo, Wentao Liu, and Xiaogang Wang. 3d human mesh regression with dense correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7054–7063, 2020. 3, 6
- [67] Hongwen Zhang, Jie Cao, Guo Lu, Wanli Ouyang, and Zhenan Sun. Danet: Decompose-and-aggregate network for 3d human shape and pose estimation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 935–944. ACM, 2019. 6
- [68] 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, 5, 12
- [69] 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. 3, 13, 14
- [70] Yuxiang Zhang, Zhe Li, Tao Yu, Mengcheng Li, Liang An, and Yebin Liu. Light-weight multi-person total capture using sparse multi-view cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2
- [71] Yang Zheng, Ruizhi Shao, Yuxiang Zhang, Tao Yu, Zerong Zheng, Qionghai Dai, and Yebin Liu. Deepmulticap: Performance capture of multiple characters using sparse multiview cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 3
- [72] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7739–7749, 2019. 3
- [73] 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. 6, 12
- [74] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4491–4500, 2019. 3

## A. Appendix

### A.1. More Experimental Details

Our network is trained with the Adam [24] optimizer and batch size of 64. The learning rate is set to  $5e-5$  without learning rate decay during training. Similar to SPIN [26], 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 [22] 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 [73] for 3D rotations in the pose parameters  $\theta$ . 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.

**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 [22] or SPIN [26], 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.

### A.2. Datasets

Following the protocols of previous work [22, 26], we train our network on several datasets with 3D or 2D annotations, including Human3.6M [17], MPI-INF-3DHP [37], LSP [19], LSP-Extended [20], MPII [2], COCO [33]. For the last five datasets, we also incorporate the SMPL parameters fitted in [5, 26] as pseudo ground-truth annotations for training. Here, we provide more descriptions of the datasets to supplement the main manuscript.

**3DPW** [57] 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 [23, 26], we do not use its data for training but only perform evaluations on its test set.

**Human3.6M** [17] 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 [34] to the sparse 3D MoCap marker data, as done in Kanazawa *et al.* [22]. It is common protocols [43, 44, 22] 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** [37] 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** [20] 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** [19] 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.* [29]. There are 1,000 samples used for evaluation.

**MPII** [2] 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** [33] 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 [68], we crop input images using the ground-truth bounding boxes.



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

Figure 6: 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.

### A.3. 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 [22, 44, 26] 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 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. 6, 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.* [29], 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<sup>1</sup>, 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<sub>50</sub> (AP at OKS = 0.50), AP<sub>75</sub> (AP at OKS = 0.75), AP<sub>M</sub> for persons with medium sizes, and AP<sub>L</sub> for persons with large sizes.

### A.4. 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 [69] and SSP-3D [49] 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. 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. 7, and qualitative results are shown in Figures 7 and 8. 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 8.

**More Qualitative Results.** We provide more qualitative results and compare our PyMAF with the state-of-the-art approach SPIN [26]. Figure 9 shows the qualitative differences between each iterative loop in SPIN [26] 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

	3DOH50K		SSP-3D
	PVE↓	MPJPE↓	mIOU↑
SPIN [26]	113.4	102.3	70.2
Baseline	113.1	102.0	70.8
PyMAF	<b>107.3</b>	<b>96.2</b>	<b>72.1</b>

Table 7: Reconstruction performances on 3DOH50K and SSP-3D datasets.

mesh parameters more effectively. In Figure 10, 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 11 and 12, 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 13, where PyMAF may fail when the initial reconstructed results have severe deviations due to the heavy occlusions or ambiguous limb connections in complex scenes.

<sup>1</sup><https://cocodataset.org/#keypoints-eval>

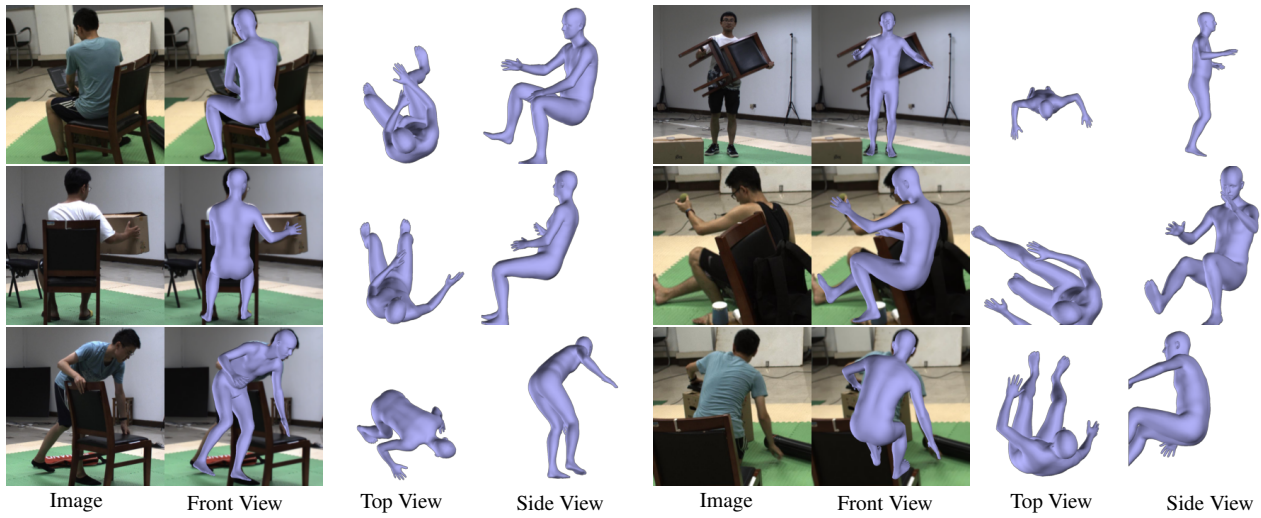


Figure 7: Reconstruction results of PyMAF on the 3DOH50K [69] dataset. PyMAF helps to handle occlusions.



Figure 8: Reconstruction results of PyMAF on the SSP-3D [49] dataset. PyMAF fails to handle extreme shapes due to the lack of training data.

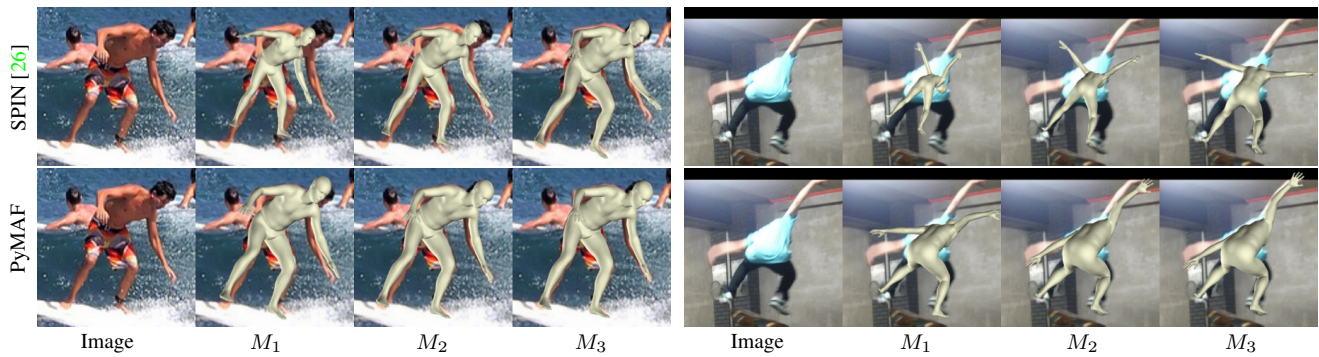


Figure 9: Qualitative differences between each iterative loop of the SPIN [26] using global features vs. the PyMAF using spatial features.



Figure 10: Qualitative comparison of the reconstruction results between SPIN [26] 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).

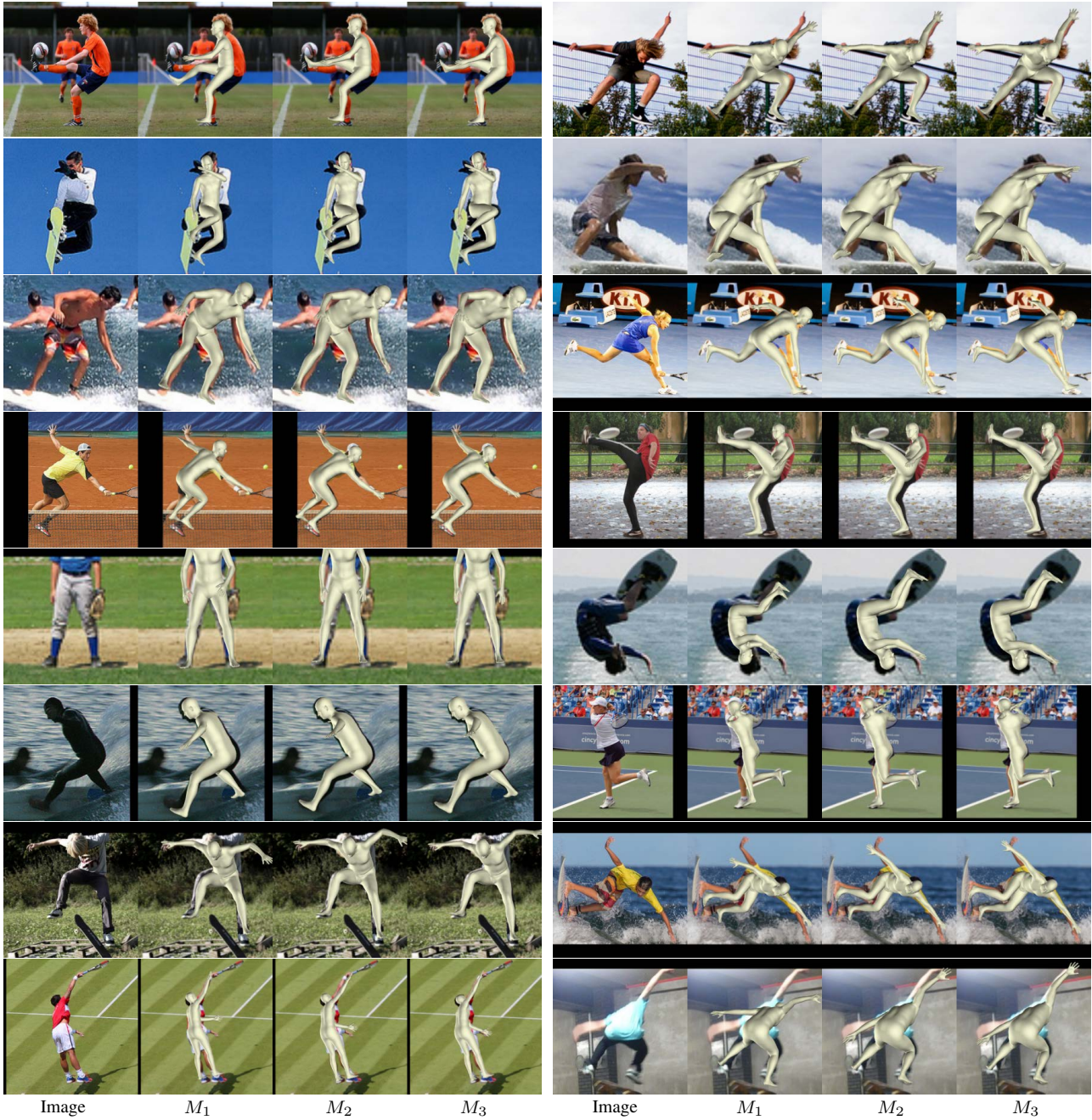


Figure 11: Successful results of PyMAF on the COCO dataset. For each example from left to right: image, the results after each iteration.



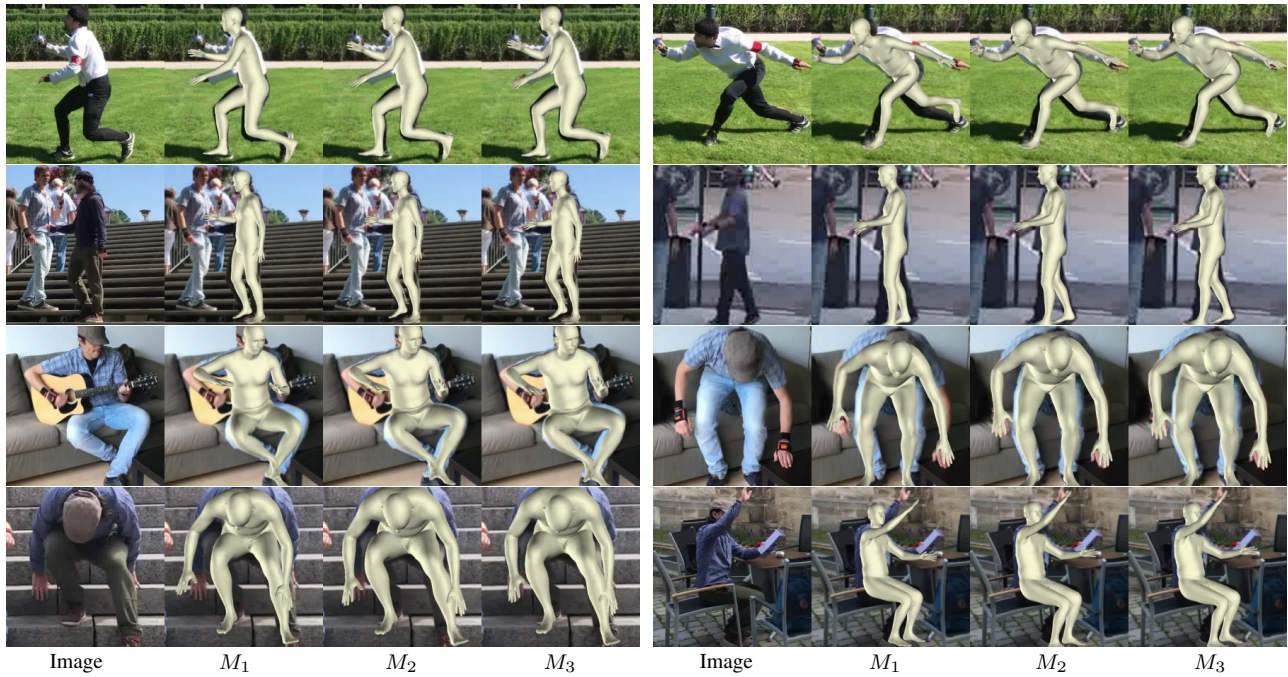


Figure 12: Successful results of PyMAF on the 3DPW dataset. Examples have the same layout with Figure 11.

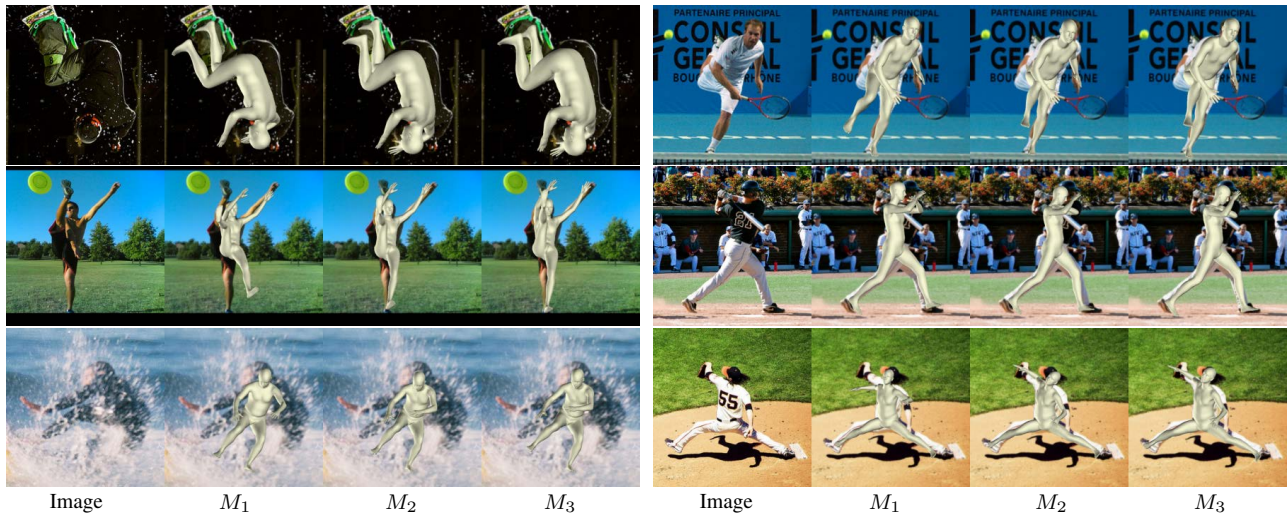


Figure 13: Erroneous reconstructions of our network. Though PyMAF can improve the alignment of some body parts, it remains challenging for PyMAF to correct those body parts with severe deviations, heavy occlusions, or ambiguous limb connections. Examples have the same layout with Figure 11.